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

8

H8/3048B Group

Hardware Manual

Renesas 8-Bit Single-Chip Microcomputer
H8 Family/H8/300H Series

H8/3048B	HD6433048B
	HD6433048BV
H8/3048F-ONE	HD64F3048B
	HD64F3048BV

Renesas Electronics
www.renesas.com

Rev.3.00 2006.09

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

RENESAS

2. Treatment of Unused Input Pins

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

3. Processing before Initialization

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

4. Prohibition of Access to Undefined or Reserved Addresses

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

spaces.

The on-chip emulator (E10T)*2 has functions that allow it to emulate directly a microc
mounted on the user board. This makes possible on-board program debugging.

The on-chip supporting functions include ROM, RAM, a 16-bit integrated timer unit (
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), a refresh controller, and other facilities. Of the two SCI channels, one has b
expanded to support the ISO/IEC7816-3 smart card interface. Functions have also bee
reduce power consumption in battery-powered applications: individual modules can b
standby, and the frequency of the system clock supplied to the chip can be divided do
software control.

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

With these features, the H8/3048B Group can be used to implement compact, high-pe
systems easily.

Versions with either flash memory (F-ZTAT™*1) or mask ROM as the on-chip ROM
available. This enables users to respond quickly and flexibly to changing application s
from the initial production stage through full-scale volume production.

This manual describes the H8/3048B Group hardware. For information on H8/3048 G
products, please refer to the H8/3048 Group Hardware Manual. For details of the instr
refer to the H8/300H Series Programming Manual.

Notes: 1. F-ZTAT (Flexible ZTAT) is a trademark of Renesas Technology Corp.
2. An on-chip emulator (E10T) is not provided in the mask ROM version.

1. Only programs in the on-chip flash memory can be developed and debugged. Conse emulation is not possible for programs in external memory or in the no-ROM mode
2. Refresh controller and DMAC operation are not supported, so settings should not b the registers for these modules.
3. During break mode of on-chip emulation, the watchdog timer stops counting. Acco counter value may be invalid after resuming from the break mode.
4. The FWE (BRK) pin and pins P91, P93, and P95 are reserved for the E10T, and ca used.
5. Area H'F7000 to H'F7FFF in 1-M address mode (area H'FF7000 to H'FF77FF in 16 mode) is used by the E10T, and is not available to the user.
6. The initial program instructions following a reset should be initialize stack pointer (read mode register (MDCR). (After initializing SP using the MOV.L instruction, us MOV.B instruction to read the MDCR register.)
7. Emulation of the hardware standby mode is not supported.

2. H8/3048F-ONE has single power supply with flash memory and E10T inst

Hardware Manual	H8/3048 Group (Rev. 7.0)			H8/3048B G (Rev. 3.	
ROM Type	**ZTAT**	**Mask ROM**	**F-ZTAT**		
Model Type	H8/3048	H8/3048 mask ROM version H8/3047 mask ROM version H8/3045 mask ROM version H8/3044 mask ROM version	H8/3048F	H8/3048F-ONE	H8 RO
Model Spec	PROM model	Mask ROM model	Dual power supply, flash memory is installed	Single power supply, flash memory installed, internal step-down (5 V operation model), high-speed operation model	Ma m
			Refer to 1.4, Differences between H8/3048F and H8/3048F-ONE.	Refer to 1.4.3, Differences between H8/3048F and H8/3048F-ONE.	
Model Type No.	HD6473048	HD6433048 HD6433047 HD6433045 HD6433044	HD64F3048	HD64F3048B (5 V operation model) HD64F3048BV (3 V operation model)	HD (5 m HD (3 m
Pin Assignment	Refer to figure 1.2, Pin Arrangement of H8/3048ZTAT, H8/3048 Mask ROM Version, H8/3047 Mask ROM Version, H8/3045 Mask ROM Version, H8/3044 Mask ROM Version, and H8/3048F (FP-100B or TFP-100B, Top View), in section 1.			5-V operation models and an external capa connected. Refer to figure 1.3, H Pin Arrangement (FP 100B, Top View), in s	

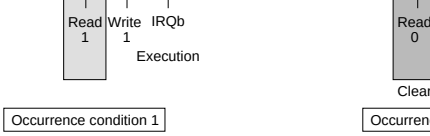
		H8/3044: 32 kbytes			
Flash Memory	—	—	Refer to section 19, Flash Memory (H8/3048F Dual Power Supply).	Refer to section 18, ROM (H8/3048F-ONE: Single Power Supply, H8/3048B Mask ROM Version)	
Clock Pulse Generator	Refer to section 20, Clock Pulse Generator.			Refer to section 19, Cl Generator.	
Power-Down State	Refer to section 21, Power-Down State.			Refer to section 20, Po State.	
	Clock oscillator settling time: Waiting time of up to 131072 states			Clock oscillator settling time of up to 262144 s	
Electrical Characteristics (Clock Rate)	Refer to table 22.1, Electrical Characteristics of H8/3048 Group Products, in section 22.			Refer to table 21.1, Ele Characteristics of H8/3 and H8/3048B Group F section 21.	
	1 to 18 MHz		1 to 16 MHz	5 V operation models: 2 to 25 MHz, 3 V operation models: 2 to 25 MHz.	
List of Registers	Refer to table B.1, Comparison of H8/3048 Group Internal I/O Register Specificatio appendix B.				
	Refer to appendix B.1, Addresses.				
Notes on Usage	—	—	—	Refer to section 1.4, Notes on H8/3048F-ONE (Single Power Supply)	
On-chip Emulator (E10T)	—	—	—	On-chip emulator (E10T)	

1.3.1 Pin Arrangement Figure 1.3 H8/3048B Group Pin Arrangement (FP-100B or TFP-100B, Top View)	8	Note amended Note: 1. For the 5 V operation product, this pin is u V_{CL} terminal, and for the 3 V operation models, this as the V_{CC} terminal that requires an external capaci
1.3.3 Pin Functions Table 1.4 Pin Functions	18, 19	Table amended

Type	Symbol	Pin No.	I/O	Name and Function
A/D and D/A converters	AV_{CC}	76	Input	Power supply pin for the converters. Connect to th supply (V_{CC}) when not us D/A converters.
	AV_{SS}	86	Input	Ground pin for the A/D a Connect to system grour
	V_{REF}	77	Input	Reference voltage input D/A converters. Connect power supply (V_{CC}) when and D/A converters.

1.4.2 Product Type Names and Markings Table 1.5 Differences in H8/3048F and H8/3048F-ONE	21	Sample markings amended

	Dual Power Supply Model: H8/3048F	Single Power Supply Model: H8/
Sample markings	H8/3048 3J1 HD 64F3048F16	64F3048F25 H8/3048F-ONE PGM 5.0 B 0021 BK80090



10.2.3 Timer Mode Register (TMDR) — 335

Bit 6—Phase Counting Mode Flag (MDF)

Table amended

Counting Direction	Down-Counting				Up-Counting	
TCLKA pin	↑	High	↓	Low	↑	Low
TCLKB pin	Low	↑	High	↓	High	↑

13.2.8 Bit Rate Register (BRR) — 473, 475

Table 13.3 Examples of Bit Rates and BRR Settings in Asynchronous Mode

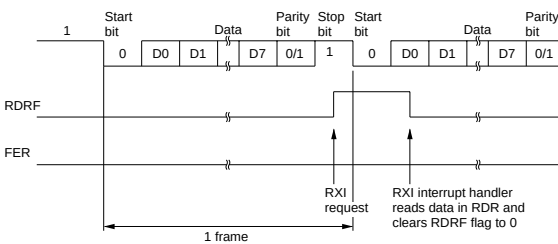
Table amended

	ϕ (MHz)								
	3			3.6864			25		
Bit Rate (bits/s)	n	N	Error (%)	n	N	Error (%)	n	N	Erro (%)
110	1	212	0.03	2	64	0.70	3	110	–0.0
150	1	155	0.16	1	191	0.00	3	80	0.47
300	1	77	0.16	1	95	0.00	2	162	–0.1
600	0	155	0.16	0	191	0.00	2	80	0.47
1200	0	77	0.16	0	95	0.00	1	162	–0.1
2400	0	38	0.16	0	47	0.00	1	80	0.47
4800	0	19	–2.34	0	23	0.00	0	162	–0.1
9600	0	9	–2.34	0	11	0.00	0	80	0.47
19200	0	4	–2.34	0	5	0.00	0	40	–0.7
31250	0	2	0.00	0	3	–7.84	0	24	0.00
38400	0	1	22.07	0	2	0.00	0	19	1.73

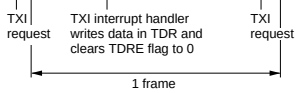
13.3.2 Operation in Asynchronous Mode — 492

Figure 13.8 Example of SCI Receive Operation (8-Bit Data with Parity and One Stop Bit)

Figure amended



RENESAS



13.3.4 Synchronous Operation Clock	500	Description amended An internal clock generated by the on-chip baud rat generator or an external clock input from the SCK p selected by setting the C/$\overline{A}$ bit in SMR and the CKE CKE0 bits in SCR. See table 13.9.
14.2.3 Serial Mode Register (SMR) Bit 7—GSM Mode (GM)	521	Table amended

Bit 7: GM	Description
0	Using the regular smart card interface mode • The TEND flag is set 12.5 etu after the beginning of the start • Clock output on/off control only
1	Using the GSM mode smart card interface mode • The TEND flag is set 11.0 etu after the beginning of the start • Clock output on/off and fixed-high/fixed-low control (set by SCR)

18.5.1 Flash Memory Control Register 1 (FLMCR1) Bit 1—Erase Bit (E)	587	Note amended Note: * Do not access flash memory while the E bit
Section 21 Electrical Characteristics Table 21.1 Electrical Characteristics of H8/3048 Group and H8/3048B Group Products	653, 654	Table amended

Item		H8/3048B Group		
		H8/3048 ZTAT	H8/3048 F-ONE (Single Power Supply)	H8/3048B Mask ROM
Absolute maximum ratings	V_{PP} pin rating	Yes	—	—
Flash memory charac-teristics*4		—	See table 21.11	—

RENESAS

specifications)

B.1 Addresses (For H8/3048F-ONE, H8/3048B Mask ROM Version) — 742

Note amended

Note: 4. Byte data must be used to access FLMCR1 FLMCR2, EBR, and RAMCR.
Registers FLMCR1, FLMCR2, EBR, and RAMCR ar implemented in the flash memory version only. The ROM version does not have these registers.

B.3 Function — 829

Table amended

ADCR

H8/3048F-ONE	Not include this register
H8/3048F H8/3048B mask ROM version H8/3048ZTAT H8/3048 mask ROM version H8/3047 mask ROM version H8/3045 mask ROM version H8/3044 mask ROM version	Include this register

ADCR — 829

Table amended

H8/3048F-ONE	Include this register
H8/3048F H8/3048B mask ROM version H8/3048ZTAT H8/3048 mask ROM version H8/3047 mask ROM version H8/3045 mask ROM version H8/3044 mask ROM version	Not include this register

SYSCR — 833

Table amended

Standby timer select 2 to 0

Bit 6	Bit 5	Bit 4	Standby Timer	
STS2	STS1	STS0	H8/3048F-ONE H8/3048B mask ROM version	*
0	0	0	Waiting time = 8,192 states	Waiting time = 8,192 states
		1	Waiting time = 16,384 states	Waiting time = 16,384 states
	1	0	Waiting time = 32,768 states	Waiting time = 32,768 states
		1	Waiting time = 65,536 states	Waiting time = 65,536 states
1	0	0	Waiting time = 131,072 states	Waiting time = 131,072 stat
		1	Waiting time = 262,144 states	Waiting time = 1,024 states
	1	0	Waiting time = 1,024 states	Illegal setting
		1	Illegal setting	Illegal setting

Note: * H8/3048F
H8/3048ZTAT
H8/3048 mask ROM version
H8/3047 mask ROM version
H8/3045 mask ROM version
H8/3044 mask ROM version

RENESAS

The H8/300H CPU has a 32-bit internal architecture with sixteen 16-bit general regist concise, optimized instruction set designed for speed. It can address a 16-Mbyte linea space. Its instruction set is upward-compatible at the object-code level with the H8/30 enabling easy porting of software from the H8/300 Series.

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

The H8/3048B Group has 128 kbytes of on-chip ROM and 4 kbytes of on-chip RAM.

Seven MCU operating modes offer a choice of data bus width and address space size. (modes 1 to 7) include one single-chip mode and six expanded modes.

In addition to mask ROM products, the H8/3048B Group includes F-ZTAT™*1 versi with on-chip user-programmable flash memory. It enables users to respond quickly an changing application specifications as well as to conditions when ramping up from ini volume production. The on-chip emulator (E10T)*2 is capable of direct emulation of t microcontroller when mounted in the user's system, thereby making possible on-boar debugging.

Table 1.1 summarizes the features of the H8/3048B Group.

Notes: 1. F-ZTAT (Flexible ZTAT™) is a trademark of Renesas Technology Corp.
2. An on-chip emulator (E10T) is not provided in the mask ROM version.

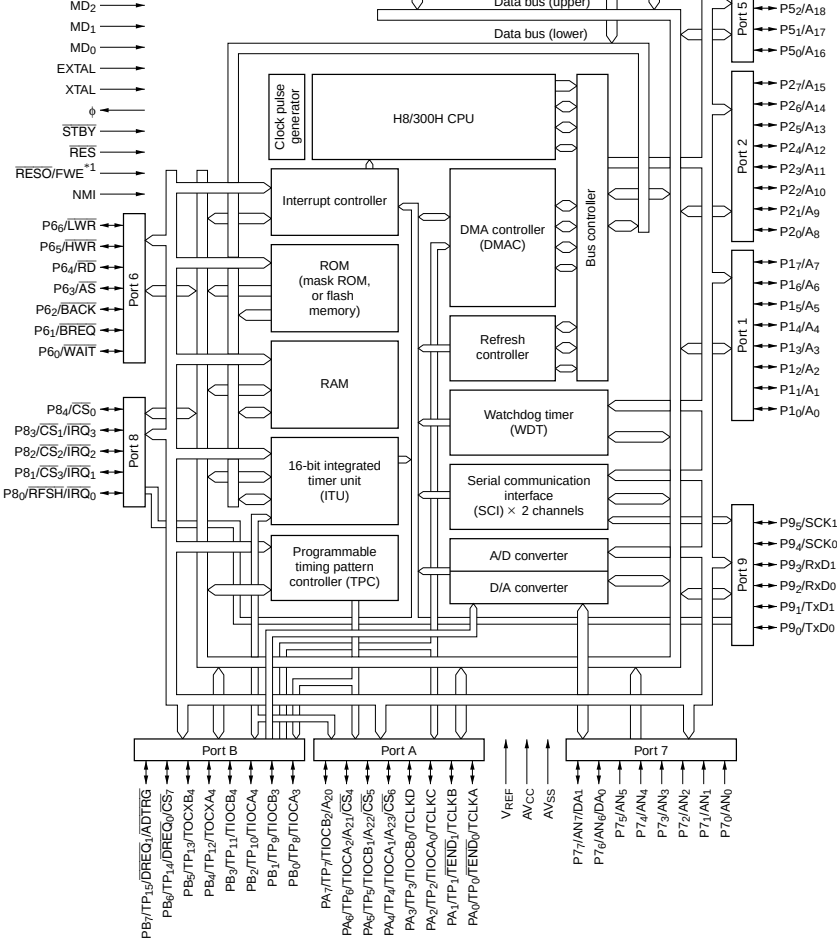
	• High-speed operation (flash memory version) — Maximum clock rate: 25 MHz — Add/subtract: 80 ns — Multiply/divide: 560 ns — 16-Mbyte address space • Instruction features — 8/16/32-bit data transfer, arithmetic, and logic instructions — Signed and unsigned multiply instructions (8 bits × 8 bits, 16 b bits) — Signed and unsigned divide instructions (16 bits ÷ 8 bits, 32 b bits) — Bit accumulator function — Bit manipulation instructions with register-indirect specificatior positions
Memory	• ROM: 128 kbytes • RAM: 4 kbytes
Interrupt controller	• Seven external interrupt pins: NMI, $\overline{IRQ}_0$ to $\overline{IRQ}_5$ • 30 internal interrupts • Three selectable interrupt priority levels
Bus controller	• Address space can be partitioned into eight areas, with independ specifications in each area • Chip select output available for areas 0 to 7 • 8-bit access or 16-bit access selectable for each area • Two-state or three-state access selectable for each area • Selection of four wait modes • Bus arbitration function

	• Usable as an interval timer
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 interrup channels 0 to 3, transmit-data-empty and receive-data-full in SCI channel 0, or external 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 interrup channels 0 to 3, external requests, or auto-request
16-bit integrated timer unit (ITU)	• Five 16-bit timer channels, capable of processing up to 12 pulse 10 pulse inputs • One 16-bit timer counter (channels 0 to 4) • Two multiplexed output compare/input capture pins (channels 0 • Operation can be synchronized (channels 0 to 4) • PWM mode available (channels 0 to 4) • Phase counting mode available (channel 2) • Buffering available (channels 3 and 4) • Reset-synchronized PWM mode available (channels 3 and 4) • Complementary PWM mode available (channels 3 and 4) • DMAC can be activated by compare match/input capture A inter (channels 0 to 3)

<table>
<tr><td>Serial communication interface (SCI), 2 channels</td><td>• Selection of asynchronous or synchronous mode
• Full duplex: can transmit and receive simultaneously
• On-chip baud-rate generator
• Smart card interface functions added (SCI0 only)</td></tr>
<tr><td>A/D converter</td><td>• 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 externally triggered</td></tr>
<tr><td>D/A converter</td><td>• Resolution: 8 bits
• Two channels
• D/A outputs can be sustained in software standby mode</td></tr>
<tr><td>I/O ports</td><td>• 70 input/output pins
• 8 input-only pins</td></tr>
<tr><td>Operating modes</td><td>• Seven MCU operating modes
<table>
<tr><th>Mode</th><th>Address Space</th><th>Address Pins</th><th>Initial Bus Width</th><th>Max.</th></tr>
<tr><td>Mode 1</td><td>1 Mbyte</td><td>A_{19} to A_0</td><td>8 bits</td><td>16 bi</td></tr>
<tr><td>Mode 2</td><td>1 Mbyte</td><td>A_{19} to A_0</td><td>16 bits</td><td>16 bi</td></tr>
<tr><td>Mode 3</td><td>16 Mbytes</td><td>A_{23} to A_0</td><td>8 bits</td><td>16 bi</td></tr>
<tr><td>Mode 4</td><td>16 Mbytes</td><td>A_{23} to A_0</td><td>16 bits</td><td>16 bi</td></tr>
<tr><td>Mode 5</td><td>1 Mbyte</td><td>A_{19} to A_0</td><td>8 bits</td><td>16 bi</td></tr>
<tr><td>Mode 6</td><td>16 Mbytes</td><td>A_{23} to A_0</td><td>8 bits</td><td>16 bi</td></tr>
<tr><td>Mode 7</td><td>1 Mbyte</td><td>—</td><td>—</td><td>—</td></tr>
</table>
• On-chip ROM is disabled in modes 1 to 4</td></tr>
</table>

Product lineup

Model (5 V)	Model (3 V)	Package	ROM
HD64F3048BTE	HD64F3048BVTE	100-pin TQFP (TFP-100B)	Flash memor
HD64F3048BF	HD64F3048BVF	100-pin QFP (FP-100B)	
HD6433048BTE	HD6433048BVTE	100-pin TQFP (TFP-100B)	Mask ROM
HD6433048BF	HD6433048BVF	100-pin QFP (FP-100B)	



Notes: 1. This pin functions as the FWE (input) pin on the H8/3048F-ONE (single power supply on-chip flash memory ver
H8/300H Series versions with on-chip mask ROM it functions as the RESO (output) pin, and on dual power sup
memory versions (V_{PP} = 12 V) and on-chip PROM versions it functions as the RESO (output)/V_{PP} (input) pin.
2. Pin 1 on the H8/3048B Group which operates at 5 V is not used as the V_{CC} terminal, but is used as the V_{CL} terr
external capacitor must be connected. Pin 1 is the V_{CC} pin on versions that operate on 3 V.

Figure 1.1 Block Diagram

have a V_{CL} pin. The 3 V operation models of the H8/3048B Group have pin 1, which i
power supply pin. See section 1.4, Notes on H8/3048F-ONE (Single Power Supply). E
the differences shown in table 1.2, the pin arrangements are the same.

Table 1.2 Comparison of H8/3048B Group and H8/3048 Group Pin Arrangem

Package	Pin Number	H8/3048F-ONE		H8/3048B Mask ROM Version		H8/3048F	H8/3048 ZTAT	H8/3048 Mask ROM Version	H8/3047 Mask ROM Version
		5 V Operation Model	3 V Operation Model	5 V Operation Model	3 V Operation Model				
FP-100B (TFP-100B)	1	V_{CL}	V_{CC}	V_{CL}	V_{CC}	V_{CC}	V_{CC}	V_{CC}	V_{CC}
	10	FWE	FWE	$\overline{RESO}$	$\overline{RESO}$	V_{PP}/ $\overline{RESO}$	V_{PP}/ $\overline{RESO}$	$\overline{RESO}$	$\overline{RESO}$

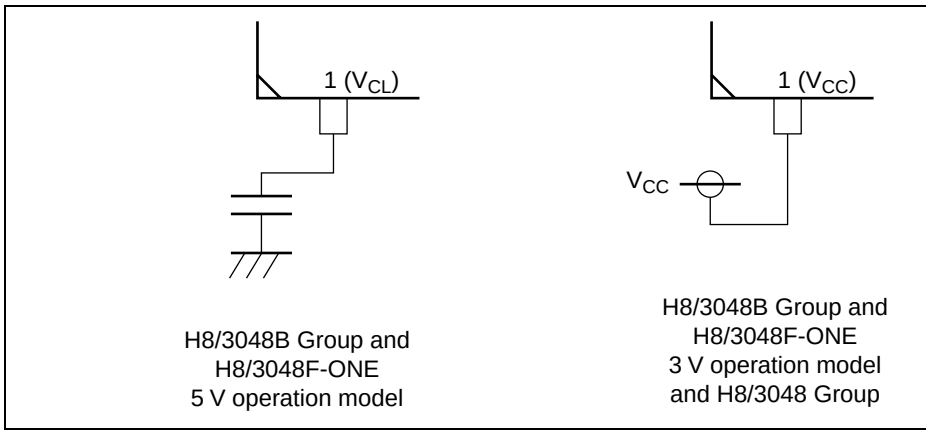
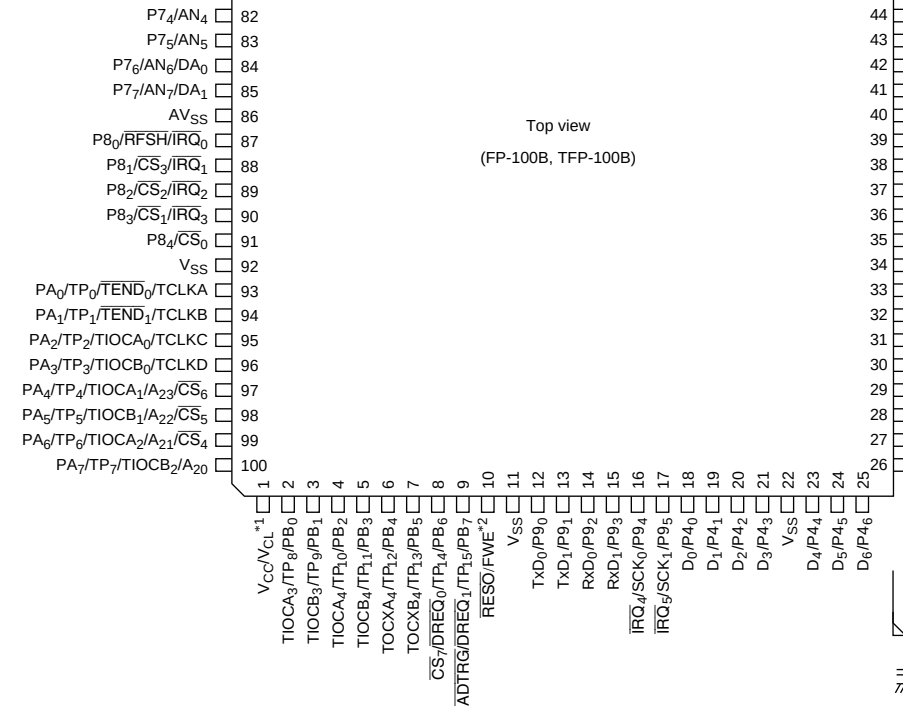


Figure 1.2 Connection of Pin 1



Notes: 1. For the 5 V operation product, this pin is used as the V_{CL} terminal, and for the 3 V opera
this pin is used as the V_{CC} terminal that requires an external capacitor.

2. (1) Pin 10 of the H8/3048F-ONE (single power supply version) functions as the FWE pi
H8/3048F-ONE has no $\overline{\text{RESO}}$ output. Pin 10 functions as the $\overline{\text{RESO}}$ pin in on-chip r
versions and as the $\overline{\text{RESO}}$/V_{PP} pin in on-chip PROM versions and dual power supp
memory versions.

(2) Do NOT apply 12 V to the H8/3048F-ONE (single power supply), or to H8/3048 Gro
H8/3048B Group mask ROM products as the chip will be destroyed.

Figure 1.3 H8/3048B Group Pin Arrangement (FP-100B or TFP-100B, Top

1*3	V_{CL} (V_{CC})	V_{CL} (V_{CC})	V_{CL} (V_{CC})	V_{CL} (V_{CC})	V_{CL} (V_{CC})	V_{CL} (V_{CC})	V_{CL} (V_{CC})	—	V_{CL} (V_{CC}
	V_{CC}	V_{CC}	V_{CC}	V_{CC}	V_{CC}	V_{CC}	V_{CC}	V_{CC}	V_{CC}
2	PB_0/TP_8/ $TIOCA_3$	PB_0/TP_8/ $TIOCA_3$	PB_0/TP_8/ $TIOCA_3$	PB_0/TP_8/ $TIOCA_3$	PB_0/TP_8/ $TIOCA_3$	PB_0/TP_8/ $TIOCA_3$	PB_0/TP_8/ $TIOCA_3$	NC	NC
3	PB_1/TP_9/ $TIOCB_3$	PB_1/TP_9/ $TIOCB_3$	PB_1/TP_9/ $TIOCB_3$	PB_1/TP_9/ $TIOCB_3$	PB_1/TP_9/ $TIOCB_3$	PB_1/TP_9/ $TIOCB_3$	PB_1/TP_9/ $TIOCB_3$	NC	NC
4	PB_2/TP_{10}/ $TIOCA_4$	PB_2/TP_{10}/ $TIOCA_4$	PB_2/TP_{10}/ $TIOCA_4$	PB_2/TP_{10}/ $TIOCA_4$	PB_2/TP_{10}/ $TIOCA_4$	PB_2/TP_{10}/ $TIOCA_4$	PB_2/TP_{10}/ $TIOCA_4$	NC	NC
5	PB_3/TP_{11}/ $TIOCB_4$	PB_3/TP_{11}/ $TIOCB_4$	PB_3/TP_{11}/ $TIOCB_4$	PB_3/TP_{11}/ $TIOCB_4$	PB_3/TP_{11}/ $TIOCB_4$	PB_3/TP_{11}/ $TIOCB_4$	PB_3/TP_{11}/ $TIOCB_4$	NC	NC
6	PB_4/TP_{12}/ $TOCXA_4$	PB_4/TP_{12}/ $TOCXA_4$	PB_4/TP_{12}/ $TOCXA_4$	PB_4/TP_{12}/ $TOCXA_4$	PB_4/TP_{12}/ $TOCXA_4$	PB_4/TP_{12}/ $TOCXA_4$	PB_4/TP_{12}/ $TOCXA_4$	NC	NC
7	PB_5/TP_{13}/ $TOCXB_4$	PB_5/TP_{13}/ $TOCXB_4$	PB_5/TP_{13}/ $TOCXB_4$	PB_5/TP_{13}/ $TOCXB_4$	PB_5/TP_{13}/ $TOCXB_4$	PB_5/TP_{13}/ $TOCXB_4$	PB_5/TP_{13}/ $TOCXB_4$	NC	NC

10*4	FWE	FWE	FWE	FWE	FWE	FWE	FWE	—	FWE
	$\overline{\text{RESO}}$	$\overline{\text{RESO}}$	$\overline{\text{RESO}}$	$\overline{\text{RESO}}$	$\overline{\text{RESO}}$	$\overline{\text{RESO}}$	$\overline{\text{RESO}}$	V_{PP}	V_{PP}
11	V_{SS}	V_{SS}	V_{SS}	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	$P9_0$/TxD_0	NC	NC
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	$P9_1$/TxD_1	NC	NC
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	$P9_2$/RxD_0	NC	NC
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	$P9_3$/RxD_1	NC	NC
16	$P9_4$/SCK_0/ $\overline{\text{IRQ}}_4$	$P9_4$/SCK_0/ $\overline{\text{IRQ}}_4$	$P9_4$/SCK_0/ $\overline{\text{IRQ}}_4$	$P9_4$/SCK_0/ $\overline{\text{IRQ}}_4$	$P9_4$/SCK_0/ $\overline{\text{IRQ}}_4$	$P9_4$/SCK_0/ $\overline{\text{IRQ}}_4$	$P9_4$/SCK_0/ $\overline{\text{IRQ}}_4$	NC	NC
17	$P9_5$/SCK_1/ $\overline{\text{IRQ}}_5$	$P9_5$/SCK_1/ $\overline{\text{IRQ}}_5$	$P9_5$/SCK_1/ $\overline{\text{IRQ}}_5$	$P9_5$/SCK_1/ $\overline{\text{IRQ}}_5$	$P9_5$/SCK_1/ $\overline{\text{IRQ}}_5$	$P9_5$/SCK_1/ $\overline{\text{IRQ}}_5$	$P9_5$/SCK_1/ $\overline{\text{IRQ}}_5$	NC	NC
18	$P4_0$/D_0*1	$P4_0$/D_0*2	$P4_0$/D_0*1	$P4_0$/D_0*2	$P4_0$/D_0*1	$P4_0$/D_0*1	$P4_0$	NC	NC
19	$P4_1$/D_1*1	$P4_1$/D_1*2	$P4_1$/D_1*1	$P4_1$/D_1*2	$P4_1$/D_1*1	$P4_1$/D_1*1	$P4_1$	NC	NC
20	$P4_2$/D_2*1	$P4_2$/D_2*2	$P4_2$/D_2*1	$P4_2$/D_2*2	$P4_2$/D_2*1	$P4_2$/D_2*1	$P4_2$	NC	NC
21	$P4_3$/D_3*1	$P4_3$/D_3*2	$P4_3$/D_3*1	$P4_3$/D_3*2	$P4_3$/D_3*1	$P4_3$/D_3*1	$P4_3$	NC	NC
22	V_{SS}	V_{SS}	V_{SS}	V_{SS}	V_{SS}	V_{SS}	V_{SS}	V_{SS}	V_{SS}

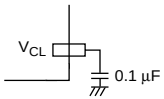
28	D_9	D_9	D_9	D_9	D_9	D_9	$P3_1$	EO_1	I/O_1
29	D_{10}	D_{10}	D_{10}	D_{10}	D_{10}	D_{10}	$P3_2$	EO_2	I/O_2
30	D_{11}	D_{11}	D_{11}	D_{11}	D_{11}	D_{11}	$P3_3$	EO_3	I/O_3
31	D_{12}	D_{12}	D_{12}	D_{12}	D_{12}	D_{12}	$P3_4$	EO_4	I/O_4
32	D_{13}	D_{13}	D_{13}	D_{13}	D_{13}	D_{13}	$P3_5$	EO_5	I/O_5
33	D_{14}	D_{14}	D_{14}	D_{14}	D_{14}	D_{14}	$P3_6$	EO_6	I/O_6
34	D_{15}	D_{15}	D_{15}	D_{15}	D_{15}	D_{15}	$P3_7$	EO_7	I/O_7
35	V_{CC}	V_{CC}	V_{CC}	V_{CC}	V_{CC}	V_{CC}	V_{CC}	V_{CC}	V_{CC}
36	A_0	A_0	A_0	A_0	$P1_0/A_0$	$P1_0/A_0$	$P1_0$	EA_0	A_0
37	A_1	A_1	A_1	A_1	$P1_1/A_1$	$P1_1/A_1$	$P1_1$	EA_1	A_1
38	A_2	A_2	A_2	A_2	$P1_2/A_2$	$P1_2/A_2$	$P1_2$	EA_2	A_2
39	A_3	A_3	A_3	A_3	$P1_3/A_3$	$P1_3/A_3$	$P1_3$	EA_3	A_3
40	A_4	A_4	A_4	A_4	$P1_4/A_4$	$P1_4/A_4$	$P1_4$	EA_4	A_4
41	A_5	A_5	A_5	A_5	$P1_5/A_5$	$P1_5/A_5$	$P1_5$	EA_5	A_5
42	A_6	A_6	A_6	A_6	$P1_6/A_6$	$P1_6/A_6$	$P1_6$	EA_6	A_6
43	A_7	A_7	A_7	A_7	$P1_7/A_7$	$P1_7/A_7$	$P1_7$	EA_7	A_7
44	V_{SS}	V_{SS}	V_{SS}	V_{SS}	V_{SS}	V_{SS}	V_{SS}	V_{SS}	V_{SS}
45	A_8	A_8	A_8	A_8	$P2_0/A_8$	$P2_0/A_8$	$P2_0$	EA_8	A_8
46	A_9	A_9	A_9	A_9	$P2_1/A_9$	$P2_1/A_9$	$P2_1$	$\overline{OE}$	$\overline{OE}$
47	A_{10}	A_{10}	A_{10}	A_{10}	$P2_2/A_{10}$	$P2_2/A_{10}$	$P2_2$	EA_{10}	A_{10}
48	A_{11}	A_{11}	A_{11}	A_{11}	$P2_3/A_{11}$	$P2_3/A_{11}$	$P2_3$	EA_{11}	A_{11}
49	A_{12}	A_{12}	A_{12}	A_{12}	$P2_4/A_{12}$	$P2_4/A_{12}$	$P2_4$	EA_{12}	A_{12}
50	A_{13}	A_{13}	A_{13}	A_{13}	$P2_5/A_{13}$	$P2_5/A_{13}$	$P2_5$	EA_{13}	A_{13}
51	A_{14}	A_{14}	A_{14}	A_{14}	$P2_6/A_{14}$	$P2_6/A_{14}$	$P2_6$	EA_{14}	A_{14}
52	A_{15}	A_{15}	A_{15}	A_{15}	$P2_7/A_{15}$	$P2_7/A_{15}$	$P2_7$	$\overline{CE}$	$\overline{CE}$
53	A_{16}	A_{16}	A_{16}	A_{16}	$P5_0/A_{16}$	$P5_0/A_{16}$	$P5_0$	V_{CC}	V_{CC}
54	A_{17}	A_{17}	A_{17}	A_{17}	$P5_1/A_{17}$	$P5_1/A_{17}$	$P5_1$	V_{CC}	V_{CC}

59	$P6_1$/$\overline{BREQ}$	$P6_1$/$\overline{BREQ}$	$P6_1$/$\overline{BREQ}$	$P6_1$/$\overline{BREQ}$	$P6_1$/$\overline{BREQ}$	$P6_1$/$\overline{BREQ}$	$P6_1$	NC	NC
60	$P6_2$/$\overline{BACK}$	$P6_2$/$\overline{BACK}$	$P6_2$/$\overline{BACK}$	$P6_2$/$\overline{BACK}$	$P6_2$/$\overline{BACK}$	$P6_2$/$\overline{BACK}$	$P6_2$	NC	NC
61	ϕ	ϕ	ϕ	ϕ	ϕ	ϕ	ϕ	NC	NC
62	$\overline{STBY}$	$\overline{STBY}$	$\overline{STBY}$	$\overline{STBY}$	$\overline{STBY}$	$\overline{STBY}$	$\overline{STBY}$	V_{SS}	V_{CC}
63	$\overline{RES}$	$\overline{RES}$	$\overline{RES}$	$\overline{RES}$	$\overline{RES}$	$\overline{RES}$	$\overline{RES}$	NC	$\overline{RES}$
64	NMI	NMI	NMI	NMI	NMI	NMI	NMI	EA_9	A_9
65	V_{SS}	V_{SS}	V_{SS}	V_{SS}	V_{SS}	V_{SS}	V_{SS}	V_{SS}	V_{SS}
66	EXTAL	EXTAL	EXTAL	EXTAL	EXTAL	EXTAL	EXTAL	NC	EXTA
67	XTAL	XTAL	XTAL	XTAL	XTAL	XTAL	XTAL	NC	XTAL
68	V_{CC}	V_{CC}	V_{CC}	V_{CC}	V_{CC}	V_{CC}	V_{CC}	V_{CC}	V_{CC}
69	$\overline{AS}$	$\overline{AS}$	$\overline{AS}$	$\overline{AS}$	$\overline{AS}$	$\overline{AS}$	$P6_3$	NC	A_{16}
70	$\overline{RD}$	$\overline{RD}$	$\overline{RD}$	$\overline{RD}$	$\overline{RD}$	$\overline{RD}$	$P6_4$	NC	NC
71	$\overline{HWR}$	$\overline{HWR}$	$\overline{HWR}$	$\overline{HWR}$	$\overline{HWR}$	$\overline{HWR}$	$P6_5$	NC	V_{CC}
72	$\overline{LWR}$	$\overline{LWR}$	$\overline{LWR}$	$\overline{LWR}$	$\overline{LWR}$	$\overline{LWR}$	$P6_6$	NC	NC
73	MD_0	MD_0	MD_0	MD_0	MD_0	MD_0	MD_0	V_{SS}	V_{SS}
74	MD_1	MD_1	MD_1	MD_1	MD_1	MD_1	MD_1	V_{SS}	V_{SS}
75	MD_2	MD_2	MD_2	MD_2	MD_2	MD_2	MD_2	V_{SS}	V_{SS}
76	AV_{CC}	AV_{CC}	AV_{CC}	AV_{CC}	AV_{CC}	AV_{CC}	AV_{CC}	V_{CC}	V_{CC}
77	V_{REF}	V_{REF}	V_{REF}	V_{REF}	V_{REF}	V_{REF}	V_{REF}	V_{CC}	V_{CC}
78	$P7_0/AN_0$	$P7_0/AN_0$	$P7_0/AN_0$	$P7_0/AN_0$	$P7_0/AN_0$	$P7_0/AN_0$	$P7_0/AN_0$	NC	NC
79	$P7_1/AN_1$	$P7_1/AN_1$	$P7_1/AN_1$	$P7_1/AN_1$	$P7_1/AN_1$	$P7_1/AN_1$	$P7_1/AN_1$	NC	NC
80	$P7_2/AN_2$	$P7_2/AN_2$	$P7_2/AN_2$	$P7_2/AN_2$	$P7_2/AN_2$	$P7_2/AN_2$	$P7_2/AN_2$	NC	NC
81	$P7_3/AN_3$	$P7_3/AN_3$	$P7_3/AN_3$	$P7_3/AN_3$	$P7_3/AN_3$	$P7_3/AN_3$	$P7_3/AN_3$	NC	NC
82	$P7_4/AN_4$	$P7_4/AN_4$	$P7_4/AN_4$	$P7_4/AN_4$	$P7_4/AN_4$	$P7_4/AN_4$	$P7_4/AN_4$	NC	NC
83	$P7_5/AN_5$	$P7_5/AN_5$	$P7_5/AN_5$	$P7_5/AN_5$	$P7_5/AN_5$	$P7_5/AN_5$	$P7_5/AN_5$	NC	NC

87	$P8_0$/ $\overline{RFSH}$/ $\overline{IRQ}_0$	$P8_0$/ $\overline{RFSH}$/ $\overline{IRQ}_0$	$P8_0$/ $\overline{RFSH}$/ $\overline{IRQ}_0$	$P8_0$/ $\overline{RFSH}$/ $\overline{IRQ}_0$	$P8_0$/ $\overline{RFSH}$/ $\overline{IRQ}_0$	$P8_0$/ $\overline{RFSH}$/ $\overline{IRQ}_0$	$P8_0/\overline{IRQ}_0$	EA_{16}	NC
88	$P8_1/\overline{CS}_3$/ $\overline{IRQ}_1$	$P8_1/\overline{CS}_3$/ $\overline{IRQ}_1$	$P8_1/\overline{CS}_3$/ $\overline{IRQ}_1$	$P8_1/\overline{CS}_3$/ $\overline{IRQ}_1$	$P8_1/\overline{CS}_3$/ $\overline{IRQ}_1$	$P8_1/\overline{CS}_3$/ $\overline{IRQ}_1$	$P8_1/\overline{IRQ}_1$	PGM	NC
89	$P8_2/\overline{CS}_2$/ $\overline{IRQ}_2$	$P8_2/\overline{CS}_2$/ $\overline{IRQ}_2$	$P8_2/\overline{CS}_2$/ $\overline{IRQ}_2$	$P8_2/\overline{CS}_2$/ $\overline{IRQ}_2$	$P8_2/\overline{CS}_2$/ $\overline{IRQ}_2$	$P8_2/\overline{CS}_2$/ $\overline{IRQ}_2$	$P8_2/\overline{IRQ}_2$	NC	V_{CC}
90	$P8_3/\overline{CS}_1$/ $\overline{IRQ}_3$	$P8_3/\overline{CS}_1$/ $\overline{IRQ}_3$	$P8_3/\overline{CS}_1$/ $\overline{IRQ}_3$	$P8_3/\overline{CS}_1$/ $\overline{IRQ}_3$	$P8_3/\overline{CS}_1$/ $\overline{IRQ}_3$	$P8_3/\overline{CS}_1$/ $\overline{IRQ}_3$	$P8_3/\overline{IRQ}_3$	NC	$\overline{WE}$
91	$P8_4/\overline{CS}_0$	$P8_4/\overline{CS}_0$	$P8_4/\overline{CS}_0$	$P8_4/\overline{CS}_0$	$P8_4/\overline{CS}_0$	$P8_4/\overline{CS}_0$	$P8_4$	NC	NC
92	V_{SS}	V_{SS}	V_{SS}	V_{SS}	V_{SS}	V_{SS}	V_{SS}	V_{SS}	V_{SS}
93	PA_0/TP_0/ $\overline{TEND}_0$/ TCLKA	PA_0/TP_0/ $\overline{TEND}_0$/ TCLKA	PA_0/TP_0/ $\overline{TEND}_0$/ TCLKA	PA_0/TP_0/ $\overline{TEND}_0$/ TCLKA	PA_0/TP_0/ $\overline{TEND}_0$/ TCLKA	PA_0/TP_0/ $\overline{TEND}_0$/ TCLKA	PA_0/TP_0/ $\overline{TEND}_0$/ TCLKA	NC	NC
94	PA_1/TP_1/ $\overline{TEND}_1$/ TCLKB	PA_1/TP_1/ $\overline{TEND}_1$/ TCLKB	PA_1/TP_1/ $\overline{TEND}_1$/ TCLKB	PA_1/TP_1/ $\overline{TEND}_1$/ TCLKB	PA_1/TP_1/ $\overline{TEND}_1$/ TCLKB	PA_1/TP_1/ $\overline{TEND}_1$/ TCLKB	PA_1/TP_1/ $\overline{TEND}_1$/ TCLKB	NC	NC
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	PA_2/TP_2/ $TIOCA_0$/ TCLKC	NC	NC
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	PA_3/TP_3/ $TIOCB_0$/ TCLKD	NC	NC
97	PA_4/TP_4/ $TIOCA_1$/ $\overline{CS}_6$	PA_4/TP_4/ $TIOCA_1$/ $\overline{CS}_6$	PA_4/TP_4/ $TIOCA_1$/ $\overline{CS}_6$	PA_4/TP_4/ $TIOCA_1$/ $\overline{CS}_6$	PA_4/TP_4/ $TIOCA_1$/ $\overline{CS}_6$	PA_4/TP_4/ $TIOCA_1$/ $A_{23}/\overline{CS}_6$	PA_4/TP_4/ $TIOCA_1$	NC	NC
98	PA_5/TP_5/ $TIOCB_1$/ $\overline{CS}_5$	PA_5/TP_5/ $TIOCB_1$/ $\overline{CS}_5$	PA_5/TP_5/ $TIOCB_1$/ $\overline{CS}_5$	PA_5/TP_5/ $TIOCB_1$/ $\overline{CS}_5$	PA_5/TP_5/ $TIOCB_1$/ $\overline{CS}_5$	PA_5/TP_5/ $TIOCB_1$/ $A_{22}/\overline{CS}_5$	PA_5/TP_5/ $TIOCB_1$	NC	NC
99	PA_6/TP_6/ $TIOCA_2$/ $\overline{CS}_4$	PA_6/TP_6/ $TIOCA_2$/ $\overline{CS}_4$	PA_6/TP_6/ $TIOCA_2$/ $\overline{CS}_4$	PA_6/TP_6/ $TIOCA_2$/ $\overline{CS}_4$	PA_6/TP_6/ $TIOCA_2$/ $\overline{CS}_4$	PA_6/TP_6/ $TIOCA_2$/ $A_{21}/\overline{CS}_4$	PA_6/TP_6/ $TIOCA_2$	NC	NC
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/ $TIOCB_2$	NC	NC

H8/3048 mask ROM version, H8/3047 mask ROM version, H8/3045 mask R
version, and H8/3044 mask ROM version, this pin is also used as the V_{CC} ter

4. This pin functions as the FWE pin on the single power supply on-chip flash m
version. Under no circumstances should 12 V be applied to the single power
chip flash memory version (H8/3048F-ONE), or to H8/3048 Group or H8/304
mask ROM products. Doing so will destroy the chip. This pin functions as an
control signal in modes 5, 6, and 7. The pin functions as the $\overline{\text{RESO}}$ pin in on
ROM versions, on-chip PROM versions, and dual power supply flash memor

				Connect all V_{CC} pins to the syste supply.
	V_{SS}	11, 22, 44, 57, 65, 92	Input	**Ground:** For connection to gro Connect all V_{SS} pins to the 0-V s power supply.
Internal step-down pin	V_{CL}	1*2	Output	The external capacitor must be between the V_{CL} and GND (0 V) connect to V_{CC}. 
Clock	XTAL	67	Input	For connection to a crystal reso For examples of crystal resonat external clock input, see sectio Pulse Generator.
	EXTAL	66	Input	For connection to a crystal reso input of an external clock signa examples of crystal resonator a clock input, see section 19, Clo Generator.
	ϕ	61	Output	**System clock:** Supplies the sy to external devices.

			0	1	1	Mode
			1	0	0	Mode
			1	0	1	Mode
			1	1	0	Mode
			1	1	1	Mode
System control	$\overline{RES}$	63	Input	**Reset input:** When driven low, t resets the chip
	$\overline{RESO}$ ($\overline{RESO}/V_{PP}$)	10	Output	**Reset output:** For the mask RO outputs a reset signal to externa Also used as a power supply for programming of the flash memor with dual power supply.
	FWE*3	10	Input	**Write enable signal:** Write-cont for writing to flash memory for th memory version with single pow
	$\overline{STBY}$	62	Input	**Standby:** When driven low, this transition to hardware standby m
	$\overline{BREQ}$	59	Input	**Bus request:** Used by an extern master to request the bus right
	$\overline{BACK}$	60	Output	**Bus request acknowledge:** Ind the bus has been granted to an master
Interrupts	NMI	64	Input	**Nonmaskable interrupt:** Reque nonmaskable interrupt
	$\overline{IRQ_5}$ to $\overline{IRQ_0}$	17, 16, 90 to 87	Input	**Interrupt request 5 to 0:** Maska interrupt request pins
Address bus	A_{23} to A_0	97 to 100, 56 to 45, 43 to 36	Output	**Address bus:** Outputs address
Data bus	D_{15} to D_0	34 to 23, 21 to 18	Input/ output	**Data bus:** Bidirectional data bus

				the external address space; inc data on the upper data bus (D_{15}
	$\overline{LWR}$	72	Output	**Low write:** Goes low to indicat the external address space; inc data on the lower data bus (D_7
	$\overline{WAIT}$	58	Input	**Wait:** Requests insertion of wa bus cycles during access to the address space
Refresh controller	$\overline{RFSH}$	87	Output	**Refresh:** Indicates a refresh cy
	$\overline{CS_3}$	88	Output	**Row address strobe $\overline{RAS}$:** Ro strobe signal for DRAM connec
	$\overline{RD}$	70	Output	**Column address strobe $\overline{CAS}$:** address strobe signal for DRAM to area 3; used with 2$\overline{WE}$ DRA
				Write enable $\overline{WE}$: Write enable DRAM connected to area 3; us 2$\overline{CAS}$ DRAM.
	$\overline{HWR}$	71	Output	**Upper write $\overline{UW}$:** Write enable DRAM connected to area 3; us DRAM.
				Upper column address strob Column address strobe signal f connected to area 3; used with DRAM.
	$\overline{LWR}$	72	Output	**Lower write $\overline{LW}$:** Write enable DRAM connected to area 3; us DRAM.
				Lower column address strob Column address strobe signal f connected to area 3; used with DRAM.

	$TIOCA_4$ to $TIOCA_0$	4, 2, 99, 97, 95	Input/ output	**Input capture/output compare** GRA4 to GRA0 output compare capture, or PWM output
	$TIOCB_4$ to $TIOCB_0$	5, 3, 100, 98, 96	Input/ output	**Input capture/output compare** GRB4 to GRB0 output compare capture
	$TOCXA_4$	6	Output	**Output compare XA4:** PWM ou
	$TOCXB_4$	7	Output	**Output compare XB4:** PWM ou
Programmable timing pattern controller (TPC)	TP_{15} to TP_0	9 to 2, 100 to 93	Output	**TPC output 15 to 0:** Pulse outp
Serial communication interface (SCI)	TxD_1, TxD_0	13, 12	Output	**Transmit data (channels 0 and** SCI data output
	RxD_1, RxD_0	15, 14	Input	**Receive data (channels 0 and** SCI data input
	SCK_1, SCK_0	17, 16	Input/ output	**Serial clock (channels 0 and 1** SCI clock input/output
A/D converter	AN_7 to AN_0	85 to 78	Input	**Analog 7 to 0:** Analog input pins
	$\overline{ADTRG}$	9	Input	**A/D trigger:** External trigger inp starting A/D conversion
D/A converter	DA_1, DA_0	85, 84	Output	**Analog output:** Analog output f converter
A/D and D/A converters	AV_{CC}	76	Input	Power supply pin for the A/D and converters. Connect to the syste supply (V_{CC}) when not using the D/A converters.
	AV_{SS}	86	Input	Ground pin for the A/D and D/A Connect to system ground (V_{SS}).

		output	The direction of each pin can b the port 2 data direction registe
$P3_7$ to $P3_0$	34 to 27	Input/ output	**Port 3:** Eight input/output pins. The direction of each pin can b the port 3 data direction registe
$P4_7$ to $P4_0$	26 to 23, 21 to 18	Input/ output	**Port 4:** Eight input/output pins. The direction of each pin can b the port 4 data direction registe
$P5_3$ to $P5_0$	56 to 53	Input/ output	**Port 5:** Four input/output pins. of each pin can be selected in t data direction register (P5DDR)
$P6_6$ to $P6_0$	72 to 69, 60 to 58	Input/ output	**Port 6:** Seven input/output pins direction of each pin can be sel port 6 data direction register (P
$P7_7$ 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. of each pin can be selected in t data direction register (P8DDR)
$P9_5$ to $P9_0$	17 to 12	Input/ output	**Port 9:** Six input/output pins. Th of each pin can be selected in t data direction register (P9DDR)
PA_7 to PA_0	100 to 93	Input/ output	**Port A:** Eight input/output pins. direction of each pin can be sel port A data direction register (P
PB_7 to PB_0	9 to 2	Input/ output	**Port B:** Eight input/output pins. direction of each pin can be sel port B data direction register (P

Notes: 1. For H8/3048 Group products and H8/3048B Group models operating at 3 V
2. For the H8/3048B Group which operates at 5 V.
3. Do NOT apply 12 V to the H8/3048B Group as the chip will be destroyed.

12 V must not be applied to the H8/3048F-ONE (single power supply), as this will perm damage the device.

The flash memory programming power source for the H8/3048F-ONE (single power su V_{CC}.

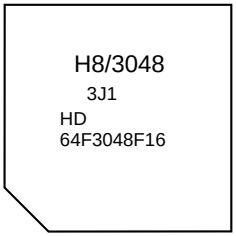
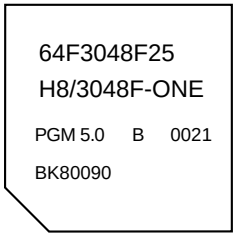
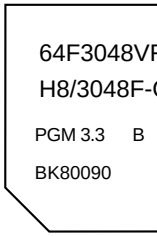
The programming power source for the dual power supply model was the V_{PP} pin (12 V is no V_{PP} pin in the single power supply models. In the H8/3048F-ONE the FWE pin is the same pin position as the V_{PP} pin in the dual power supply model, but FWE is not a p source pin-it is used to control flash memory write enabling.

Also, in boot mode, 12 V must be applied to the MD_2 pin in the dual power supply mod is not necessary in the H8/3048F-ONE (single power supply).

The maximum rating of the FWE and MD_2 pins in the H8/3048F-ONE (single power su V_{CC} +0.3 V. Applying a voltage in excess of the maximum rating will permanently dam device.

Do not select the HN28F101 programmer setting for the H8/3048F-ONE (single power this setting is made by mistake, 12.0 V may be applied to the FWE pin, causing perman damage to the device.

When using a PROM programmer to program the on-chip flash memory in the H8/3048 model (single power supply), use a PROM programmer that supports Renesas Technol microcomputer device types with 128-kbyte on-chip flash memory.

	Model: H8/3048F	Single Power Supply Model: H8/3048F-ONE	
Product type name	HD64F3048F16	HD64F3048BF25	HD64F3048BV
Sample markings			
Flash memory programming power source	V_{PP} power source (12.0 ±0.6 V)	V_{CC} power source (5.0 ±10%)	V_{CC} power sou (3.0 to 3.6 V)

1.4.3 Differences between H8/3048F and H8/3048F-ONE

Table 1.6 shows the differences between the H8/3048F (dual power supply model) an ONE (single power supply model).

<table>
<tr><th></th><th>Pin 10: V_{PP}/$\overline{\text{RESO}}$</th><th>Pin 10: FWE</th></tr>
<tr><td>ROM/RAM</td><td>128-kbyte flash memory with dual power supply, RAM: 4 kbytes</td><td>128-kbyte flash memory with
power supply, RAM: 4 kbytes</td></tr>
<tr><td>Units of on-board writing</td><td>Writing in 1-byte units</td><td>Writing in 128-byte units</td></tr>
<tr><td>Write/erase voltage</td><td>12 V is externally applied from V_{PP} pin</td><td>Application of 12 V is not req
V_{CC} single power supply</td></tr>
<tr><td>V_{PP} pin functions</td><td>Multiplexes with $\overline{\text{RESO}}$</td><td>FWE function only (no RESO</td></tr>
<tr><td>Boot mode settings</td><td>$\overline{\text{RESO}}$ = 12 V
<table><tr><th></th><th>MD2</th><th>MD1</th><th>MD0</th></tr><tr><td>Mode 5</td><td>12 V</td><td>0</td><td>1</td></tr><tr><td>Mode 6</td><td>12 V</td><td>1</td><td>0</td></tr><tr><td>Mode 7</td><td>12 V</td><td>1</td><td>1</td></tr></table>Cancelled by reset</td><td>FWE = 1
<table><tr><th></th><th>MD2</th><th>MD1</th><th>MD0</th></tr><tr><td>Mode 5</td><td>0</td><td>0</td><td>1</td></tr><tr><td>Mode 6</td><td>0</td><td>1</td><td>0</td></tr><tr><td>Mode 7</td><td>0</td><td>1</td><td>1</td></tr></table>Set to mode 1 in mode 5
Set to mode 2 in mode 6
Set to mode 3 in mode 7
Cancelled by reset</td></tr>
<tr><td>Settings for user program mode</td><td>$\overline{\text{RESO}}$ = 12 V
<table><tr><th></th><th>MD2</th><th>MD1</th><th>MD0</th></tr><tr><td>Mode 5</td><td>1</td><td>0</td><td>1</td></tr><tr><td>Mode 6</td><td>1</td><td>1</td><td>0</td></tr><tr><td>Mode 7</td><td>1</td><td>1</td><td>1</td></tr></table>Cancelled by reset</td><td>FWE = 1
<table><tr><th></th><th>MD2</th><th>MD1</th><th>MD0</th></tr><tr><td>Mode 5</td><td>1</td><td>0</td><td>1</td></tr><tr><td>Mode 6</td><td>1</td><td>1</td><td>0</td></tr><tr><td>Mode 7</td><td>1</td><td>1</td><td>1</td></tr></table>Cancelled by reset</td></tr>
<tr><td>Prewrite processing</td><td>Necessary before erasing</td><td>Not necessary</td></tr>
<tr><td>Erasing blocks</td><td>More than one block can be erased at the same time (verifies in block units and erases only the unerased blocks)</td><td>Erases in one block units. M
one block cannot be erased
time (the erasing flow is diffe</td></tr>
</table>

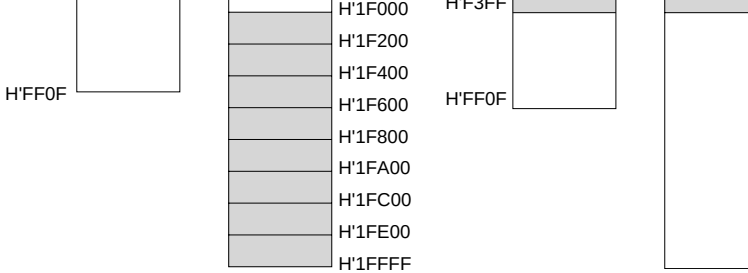
EBR	EBR1 (H'FF42) LB7 | LB6 | LB5 | LB4 | LB3 | LB2 | LB1 | LB0 EBR2 (H'FF43) SB7 | SB6 | SB5 | SB4 | SB3 | SB2 | SB1 | SB0 More than one block can be selected (setting for writing/erasing)	EBR (H'FF42) EB7 | EB6 | EB5 | EB4 | EB3 | EB Only one block can be sele for erasing)
RAMCR	RAMCR (H'FF48) FLER | — | — | — | RAMS | RAM2 | RAM1 | RAM0	RAMCR (H'FF47) — | — | — | — | RAMS | RA
Division of flash memory block	Division in 16 blocks 16 kbytes × 7: LB0 to LB6 12 kbytes × 1: LB7 512 kbytes × 8: SB0 to SB7	Division in 8 blocks 1 kbyte × 4: EB0 to EB3 28 kbytes × 1: EB4 32 kbytes × 3: EB5 to EB7

Flash memory

	H'00000
LB0 (16 kbytes)	
LB1 (16 kbytes)	
LB2 (16 kbytes)	
LB3 (16 kbytes)	
LB4 (16 kbytes)	
LB5 (16 kbytes)	
LB6 (16 kbytes)	
LB7 (12 kbytes)	
SB0 (512 bytes)	
SB1 (512 bytes)	
SB2 (512 bytes)	
SB3 (512 bytes)	
SB4 (512 bytes)	
SB5 (512 bytes)	
SB6 (512 bytes)	
SB7 (512 bytes)	
	H'1FFFF

Flash memory

	H'00000
EB0 (1 kbyte)	
EB1 (1 kbyte)	
EB2 (1 kbyte)	
EB3 (1 kbyte)	
EB4 (28 kbytes)	
EB5 (32 kbytes)	
EB6 (32 kbytes)	
EB7 (32 kbytes)	
	H'1FFF



Reset during operation	The RES signal must be kept low during at least 6 system clock (6ϕ) cycles. (RES pulse width t_{RESW} = min. 6.0 t_{cyc})	The RES signal must be kep at least 20 system clock (20ϕ (RES pulse width t_{RESW} = min.
A/D ADCR	ADCR (H'FFE9) Initial value: H'7F Only bit 7 can be read or written. Other bits are reserved and always read as 1; writing to these bits is invalid.	ADCR (H'FFE9) Initial value: H'7E Only bit 7 can be read or wri Bit 0 is reserved and must no 1. Other bits are reserved and a as 1; writing to these bits is i
WDT RSTCSR	RSTCSR (H'FFAB) Initial value: H'3F Only bits 7 and 6 can be read or written. Other bits are reserved and always read as 1; writing to these bits is invalid.	RSTCSR (H'FFAB) Initial value: H'3F Only bit 7 can be read or wri Bit 6 is reserved and must no 1. Other bits are reserved and a as 1; writing to these bits is i

		1	65,536 states
1	0	0	131,072 states
		1	1,024 states
	1	—	Illegal setting

1	0	0	131,07
		1	262,14
	1	0	1,024
		1	Illegal

Details on flash memory	Refer to section 19, Flash Memory (H8/3048F, Dual Power Supply).	Refer to section 18, ROM (H ONE: Single Power Supply, Mask ROM Version)
Electrical characteristics (clock rate)	Clock rate: 1 to 16 MHz Refer to section 22, Table 22.1 Electrical Characteristics of H8/3048 Group Products.*2	Clock rate: 2 to 25 MHz Refer to section 21, Table 2 Electrical Characteristics of Group and H8/3048B Grou
List of registers	Refer to appendix B, Table B.1 Comparison of H8/3048 Group Internal I/O Register Specifications*2	Refer to appendix B, Table Comparison of H8/3048 Gr I/O Register Specifications*
	Refer to appendix B.2, Addresses (For H8/3048F, H8/3048ZTAT, H8/3048 Mask-ROM, H8/3047 Mask-ROM, H8/3045 Mask-ROM, and H8/3044 Mask-ROM Versions)*2	Refer to appendix B.1, Add (For H8/3048F-ONE, H8/30 ROM Version)*2
On-chip emulator	—	On-chip emulator (E10T)

Notes: 1. Refer to the “H8/3048 Group, H8/3048F-ZTAT™ Hardware Manual” for info about H8/3048F.

2. H8/3048F and H8/3048F-ONE can be referred to also on this manual.

H8/3048F, H8/3048ZTAT and on-chip mask ROM models (H8/3048, H8/3047, H8/304
H8/3044).

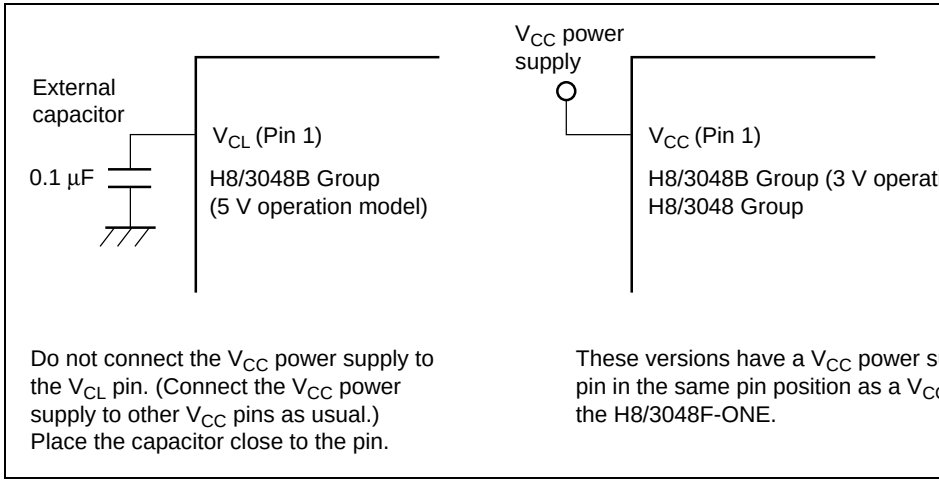


Figure 1.4 Method of Connecting H8/3048B Group V_{CL} Capacitor

The 3 V operation models of the H8/3048B Group do not have a V_{CL} pin. The 3 V oper
models have a V_{CC} power supply pin at the location of the V_{CL} pin in the 5 V operation m
Therefore, 3 V operation models do not require connection of an external capacitor, and
should be connected to the power supply in the same way as other V_{CC} pins.

Figure 1.5 Difference between 5 V and 3 V Operation Models

1.4.5 Note on Changeover to H8/3048 Group Mask ROM Version

Care is required when changing from the H8/3048F-ONE with on-chip flash memory
with on-chip H8/3048 Group mask ROM (H8/3048, H8/3047, H8/3045, or H8/3044).

An external capacitor must be connected to the V_{CL} pin of the H8/3048F-ONE (5 V m
V_{CL} pin occupies the same location as a V_{CC} pin in the on-chip mask ROM versions. Cl
a mask ROM version must therefore be taken into account when undertaking pattern d
in the board design stage.

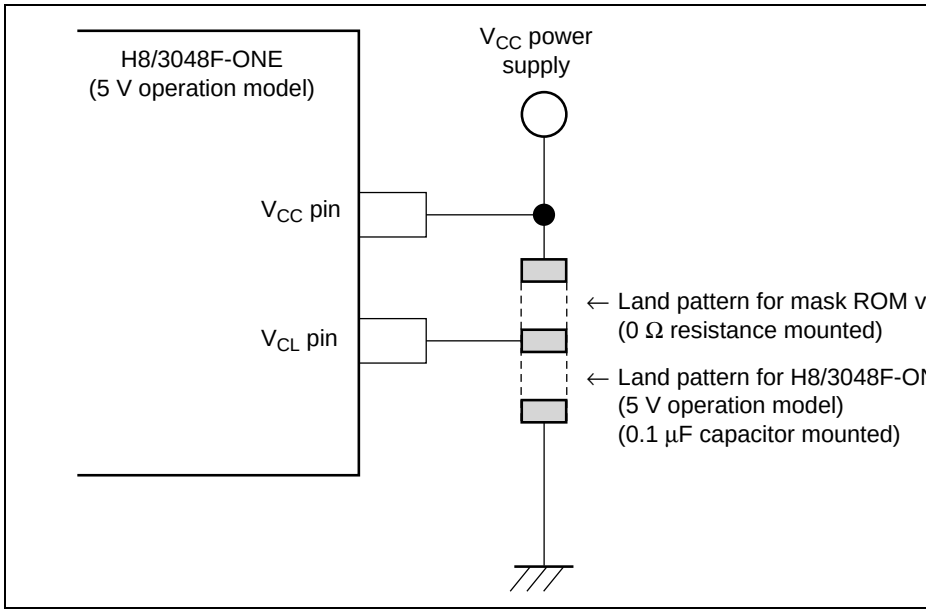


Figure 1.6 Example of Board Pattern Providing for External Capacit

operating on an external clock.

For setting details, see section 20.4.3, Selection of Waiting Time for Exit from Softwar
Mode.

1.6 Notes on Crystal Resonator Connection

The H8/3048B Group support an operating frequency of up to 25 MHz. If a crystal reso
a frequency higher than 20 MHz is connected, attention must be paid to circuit constant
external load capacitance values. For details see section 19.2.1, Connecting a Crystal R

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 data transfer and 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
- High-speed operation
  - All frequently-used instructions execute in two to four states
  - Maximum clock frequency: 25 MHz (H8/3048B Group)
  - 8/16/32-bit register-register add/subtract: 80 ns @ 25 MHz/125 ns @ 16 MHz
  - 8 × 8-bit register-register multiply: 560 ns @ 25 MHz/875 ns @ 16 MH
  - 16 ÷ 8-bit register-register divide: 560 ns @ 25 MHz/875 ns @ 16 MH

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.
    (Normal mode is not available in the H8/3048B Group.)
- 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.

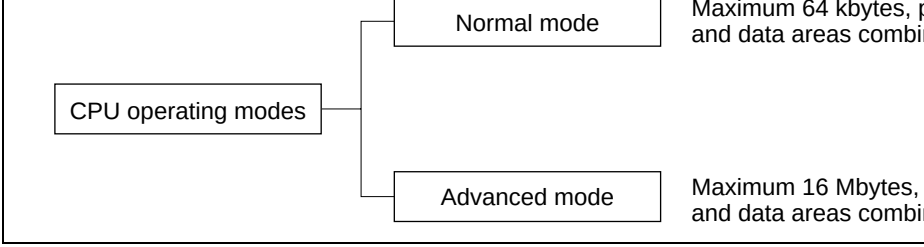


Figure 2.1 CPU Operating Modes

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

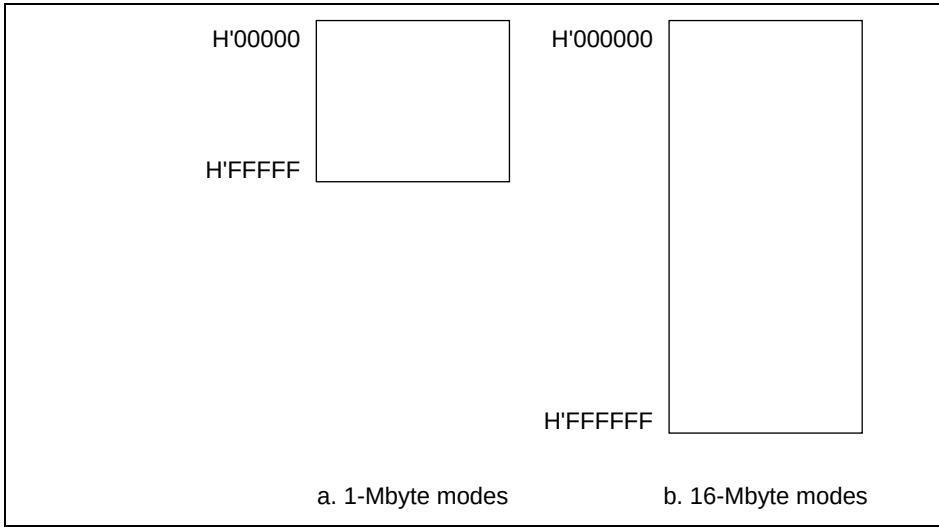
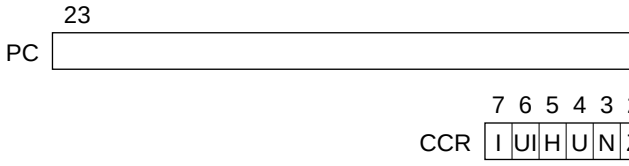


Figure 2.2 Memory Map

General Registers (ERn)

	15 0	7 0	7
ER0	E0	R0H	R0L
ER1	E1	R1H	R1L
ER2	E2	R2H	R2L
ER3	E3	R3H	R3L
ER4	E4	R4H	R4L
ER5	E5	R5H	R5L
ER6	E6	R6H	R6L
ER7	E7 (SP)	R7H	R7L

Control Registers (CR)



Legend:
SP: Stack pointer
PC: Program counter
CCR: Condition code register
I: Interrupt mask bit
UI: User bit or interrupt mask bit
H: Half-carry flag
U: User bit
N: Negative flag
Z: Zero flag
V: Overflow flag
C: Carry flag

Figure 2.3 CPU Internal Registers

(R0 to R7). These registers are functionally equivalent, providing a maximum sixteen 1
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 si
registers.

Figure 2.4 illustrates the usage of the general registers. The usage of each register can b
independently.

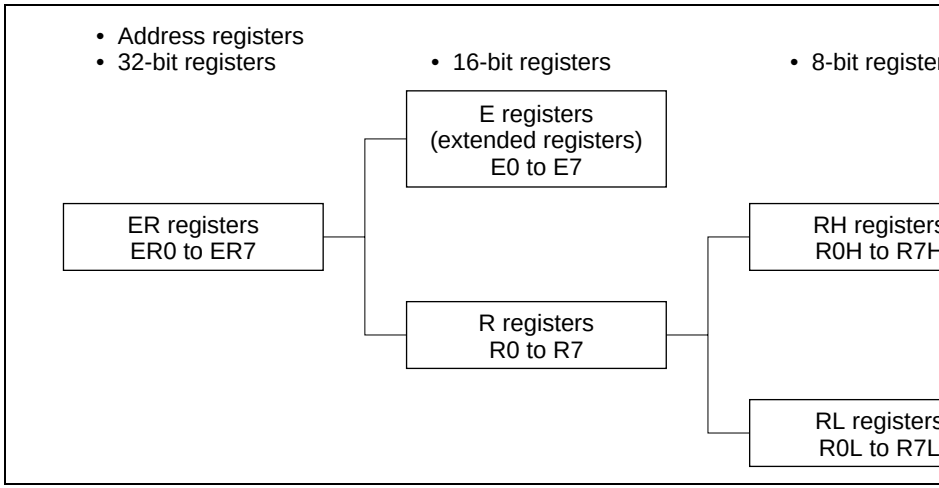


Figure 2.4 Usage of General Registers

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
(CCR).

Program Counter (PC)

This 24-bit counter indicates the address of the next instruction the CPU will execute.
of all CPU instructions is 2 bytes (one word) or a multiple of 2 bytes, so the least sign
bit is ignored. When an instruction is fetched, the least significant PC bit is regarded a

Condition Code Register (CCR)

This 8-bit register contains internal CPU status information, including the interrupt ma
and half-carry (H), negative (N), zero (Z), overflow (V), and carry (C) flags.

- **Bit 7—Interrupt Mask Bit (I):** Masks interrupts other than NMI when set to 1. N
accepted regardless of the I bit setting. The I bit is set to 1 at the start of an excepti
sequence.
- **Bit 6—User Bit or Interrupt Mask Bit (UI):** Can be written and read by softwar
LDC, STC, ANDC, ORC, and XORC instructions. This bit can also be used as an
mask bit. For details see section 5, Interrupt Controller.
- **Bit 5—Half-Carry Flag (H):** When the ADD.B, ADDX.B, SUB.B, SUBX.B, CM
NEG.B instruction is executed, this flag is set to 1 if there is a carry or borrow at b
cleared to 0 otherwise. When the ADD.W, SUB.W, CMP.W, or NEG.W instructio

other times.

- **Bit 0—Carry Flag (C):** Set to 1 when a carry occurs, and cleared to 0 otherwise. U
  - — Add instructions, to indicate a carry
  - — Subtract instructions, to indicate a borrow
  - — Shift and rotate instructions, to store the value shifted out of the end bit

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

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 cond branch (Bcc) instructions.

For the action of each instruction on the flag bits, see appendix A.1, Instruction List. Fo 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, an in CCR is set to 1. The other CCR bits and the general registers are not initialized. In pa the stack pointer (ER7) is not initialized. The stack pointer must therefore be initialized MOV.L instruction executed immediately after a reset.

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

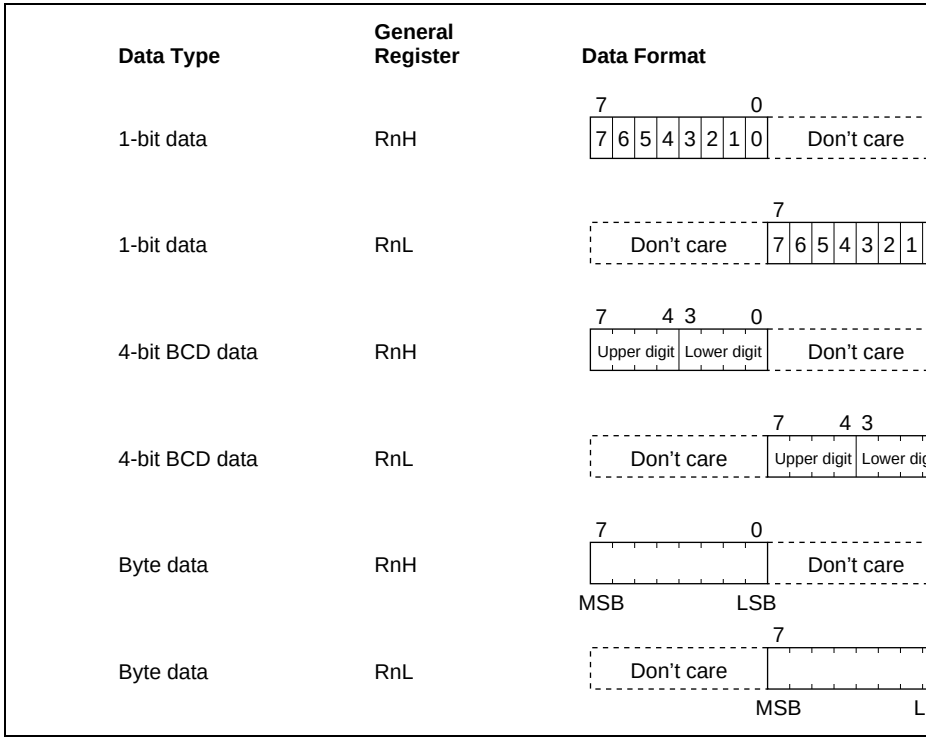
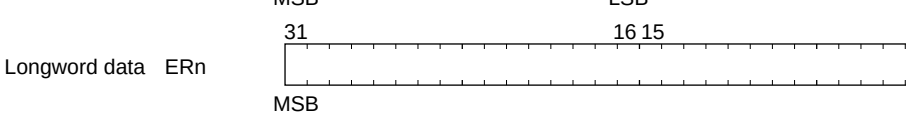


Figure 2.6 General Register Data Formats (1)



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

Figure 2.7 General Register Data Formats (2)

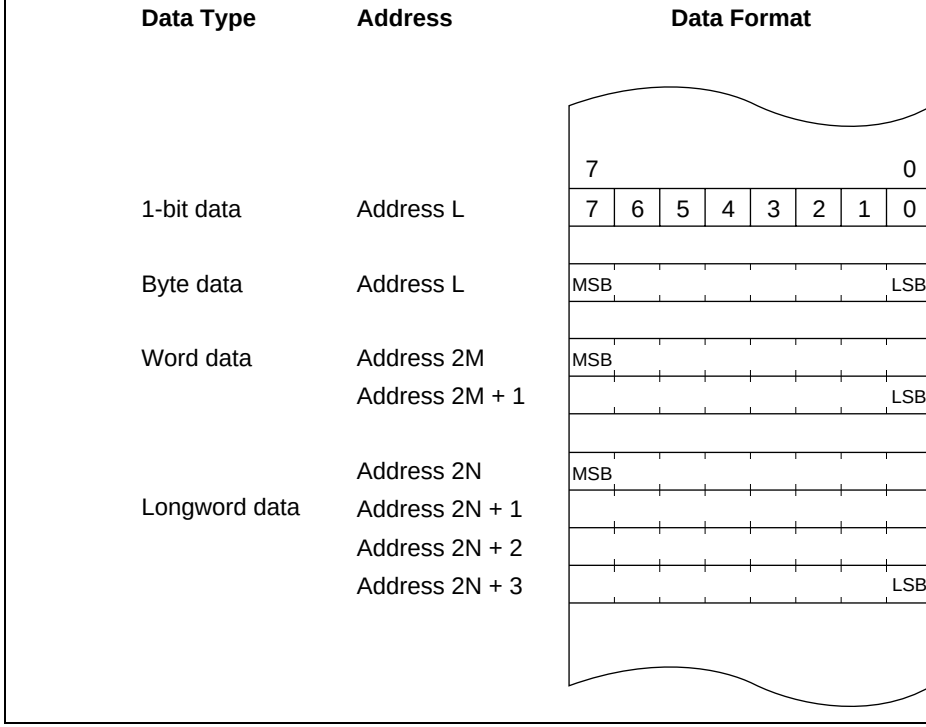


Figure 2.8 Memory Data Formats

When ER7 (SP) is used as an address register to access the stack, the operand size sho
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, BXOR, 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

Tot

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 H8/3048B Group.
3. Bcc is a generic branching instruction.

		#xx	Rn	@ERn	@(d:16,E	@(d:24,E	@ERn+/-	@aa:8	@aa:16	@aa:24	@(d:8,PC	@(d:16,P
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	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		—	—	—	—	—	—	—	—	—	—	—

Legend:
B: Byte
W: Word
L: Longword

Note: * Not availabe in the H8/3048B Group.

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 re to R7, E0 to E7), and 32-bit data or address registers (ER0 to ER7).

MOVTPE	B	Rs → (EAs) Cannot be used in the H8/3048B Group.
POP	W/L	@SP+ → Rn Pops a general register from the stack. POP.W Rn is ide MOV.W @SP+, Rn. Similarly, POP.L ERn is identical to @SP+, ERn.
PUSH	W/L	Rn → @–SP Pushes a general register onto the stack. PUSH.W Rn is MOV.W Rn, @–SP. Similarly, PUSH.L ERn is identical t ERn, @–SP.

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

		Performs addition or subtraction with carry or borrow on d general registers, or on immediate data and data in a gen register.
INC, DEC	B/W/L	Rd ± 1 → Rd, Rd ± 2 → Rd Increments or decrements a general register by 1 or 2. (B operands can 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 register.
DAA, DAS	B	Rd decimal adjust → Rd Decimal-adjusts an addition or subtraction result in a gene by referring to CCR to produce 4-bit BCD data.
MULXU	B/W	Rd × Rs → Rd Performs unsigned multiplication on data in two general re either 8 bits × 8 bits → 16 bits or 16 bits × 16 bits → 32 bi
MULXS	B/W	Rd × Rs → Rd Performs signed multiplication on data in two general regi either 8 bits × 8 bits → 16 bits or 16 bits × 16 bits → 32 bi
DIVXU	B/W	Rd ÷ Rs → Rd Performs unsigned division on data in two general registe 16 bits ÷ 8 bits → 8-bit quotient and 8-bit remainder or 32 bits → 16-bit quotient and 16-bit remainder.
DIVXS	B/W	Rd ÷ Rs → Rd Performs signed division on data in two general registers: bits ÷ 8 bits → 8-bit quotient and 8-bit remainder, or 32 bi → 16-bit quotient and 16-bit remainder.

EXTS	W/L	Rd (sign extension) → Rd Extends byte data in the lower 8 bits of a 16-bit register t or extends word data in the lower 16 bits of a 32-bit regis longword data, by extending the sign bit.
EXTU	W/L	Rd (zero extension) → Rd Extends byte data in the lower 8 bits of a 16-bit register t or extends word data in the lower 16 bits of a 32-bit regis longword data, by padding with zeros.

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

		general register or immediate data.
XOR	B/W/L	Rd ⊕ Rs → Rd, Rd ⊕ #IMM → Rd Performs a logical exclusive OR operation on a general re another general register or immediate data.
NOT	B/W/L	¬ Rd → Rd Takes the one's complement of general register contents.

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

Table 2.6 Shift Instructions

Instruction	Size*	Function
SHAL, SHAR	B/W/L	Rd (shift) → Rd Performs an arithmetic shift on general register contents.
SHLL, SHLR	B/W/L	Rd (shift) → Rd Performs a logical shift on general register contents.
ROTL, ROTR	B/W/L	Rd (rotate) → Rd Rotates general register contents.
ROTXL, ROTXR	B/W/L	Rd (rotate) → Rd Rotates general register contents through the carry bit.

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

		Clears a specified bit in a general register or memory op The bit number is specified by 3-bit immediate data or th bits of a general register.
BNOT	B	¬ (<bit-No.> of <EAd>) → (<bit-No.> of <EAd>) Inverts a specified bit in a general register or memory op bit number is specified by 3-bit immediate data or the lo a general register.
BTST	B	¬ (<bit-No.> of <EAd>) → Z Tests a specified bit in a general register or memory ope sets or clears the Z flag accordingly. The bit number is s 3-bit immediate data or the lower 3 bits of a general regi
BAND	B	C ∧ (<bit-No.> of <EAd>) → C ANDs the carry flag with a specified bit in a general regis memory 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 register or memory operand and stores the result in the 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 regist memory 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 register or memory operand and stores the result in the The bit number is specified by 3-bit immediate data.

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

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

BHI	High	C ∨ Z =
BLS	Low or same	C ∨ Z =
Bcc (BHS)	Carry clear (high or same)	C = 0
BCS (BLO)	Carry set (low)	C = 1
BNE	Not equal	Z = 0
BEQ	Equal	Z = 1
BVC	Overflow clear	V = 0
BVS	Overflow set	V = 1
BPL	Plus	N = 0
BMI	Minus	N = 1
BGE	Greater or equal	N ⊕ V =
BLT	Less than	N ⊕ V =
BGT	Greater than	Z ∨ (N ⊕
BLE	Less or equal	Z ∨ (N ⊕

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

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

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

repeat @ER5+ → @ER6+, R4 – 1 → R4
until R4 = 0

else next;

Transfers a data block according to parameters set in ge
registers R4L or R4, ER5, and ER6.

R4L or R4: Size of block (bytes)
ER5: Starting source address
ER6: Starting destination address

Execution of the next instruction begins as soon as the t
completed.

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.

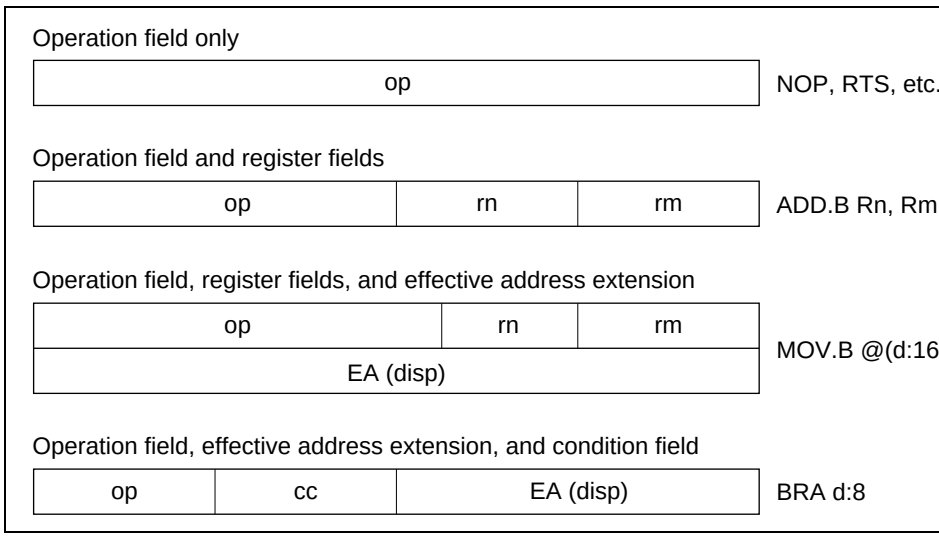


Figure 2.9 Instruction Formats

Step		Description
1	Read	Read data (byte unit) at the specified address
2	Bit manipulation	Modify the specified bit in the read data
3	Write	Write the modified data (byte unit) to the specified ad

In the following example, a BCLR instruction is executed on the data direction registe port 4.

$P4_7$ and $P4_6$ are set as input pins, and are inputting low-level and high-level signals, re

$P4_5$ to $P4_0$ are set as output pins, and are in the low-level output state.

In this example, the BCLR instruction is used to make $P4_0$ an input port.

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
DR	1	0	0	0	0	0	0

Execution of BCLR Instruction

```
BCLR  #0, @P4DDR      ; Execute BCLR instruction on DDR
```

Explanation of BCLR Instruction

To execute the BCLR instruction, the CPU begins by reading P4DDR. Since P4DDR is 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 DDR to complete the BCLR instructi

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 internal I/O registers to 0. In an handling routine, for example, if it is known that the flag is set to 1, it is not necessary t flag ahead of time.

2.7 Addressing Modes and Effective Address Calculation

2.7.1 Addressing Modes

The H8/300H CPU supports the eight addressing modes listed in table 2.11. Each instru a subset of these addressing modes. Arithmetic and logic instructions can use the regist and immediate modes. Data transfer instructions can use all addressing modes except p counter relative and memory indirect. Bit manipulation instructions use register direct, indirect, or absolute (@aa:8) addressing mode to specify an operand, and register direct BCLR, BNOT, and BTST instructions) or immediate (3-bit) addressing mode to specif number in the operand.

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-, or 32-bit register contain operand. R0H to R7H and R0L to R7L can be specified as 8-bit registers. R0 to R7 an can be specified as 16-bit registers. ER0 to ER7 can be specified as 32-bit registers.

2. Register Indirect—@ERn

The register field of the instruction code specifies an address register (ERn), the lowe which contain the address of the operand.

3. Register Indirect with Displacement—@(d:16, ERn) or @(d:24, ERn)

A 16-bit or 24-bit displacement contained in the instruction code is added to the conte address register (ERn) specified by the register field of the instruction, and the lower 2 sum specify the address of a memory operand. A 16-bit displacement is sign-extended added.

- Register indirect with pre-decrement—@–ERn

  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 n
  operand. The result is also stored in the address register. The value subtracted is 1 f
  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 absolute address of a memory operand. The absolute
may be 8 bits long (@aa:8), 16 bits long (@aa:16), or 24 bits long (@aa:24). For an 8-b
address, the upper 16 bits are all assumed to be 1 (H'FFFF). For a 16-bit absolute addre
upper 8 bits are a sign extension. A 24-bit absolute address can access the entire addres
Table 2.12 indicates the accessible 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
24 bits (@aa:24)	H'00000 to H'FFFFF (0 to 1048575)	H'000000 to H'FFFFFF (0 to 16777215)

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

This mode is used in the Bcc and BSR instructions. An 8-bit or 16-bit displacement co
the instruction code is sign-extended to 24 bits and added to the 24-bit PC contents to
24-bit branch address. The PC value to which the displacement is added is the address
byte of the next instruction, so the possible branching range is –126 to +128 bytes (–6
words) or –32766 to +32768 bytes (–16383 to +16384 words) from the branch instruc
resulting value should be an even number.

8. Memory Indirect—@@aa:8

This mode can be used by the JMP and JSR instructions. The instruction code contain
absolute address specifying a memory operand. This memory operand contains a bran
The memory operand is accessed by longword access. The first byte of the memory op
ignored, generating a 24-bit branch address. See figure 2.10. The upper bits of the 8-b
address are assumed to be 0 (H'0000), so the address range is 0 to 255 (H'000000 to H
Note that the first part of this range is also the exception vector area. For further detail
5, Interrupt Controller.

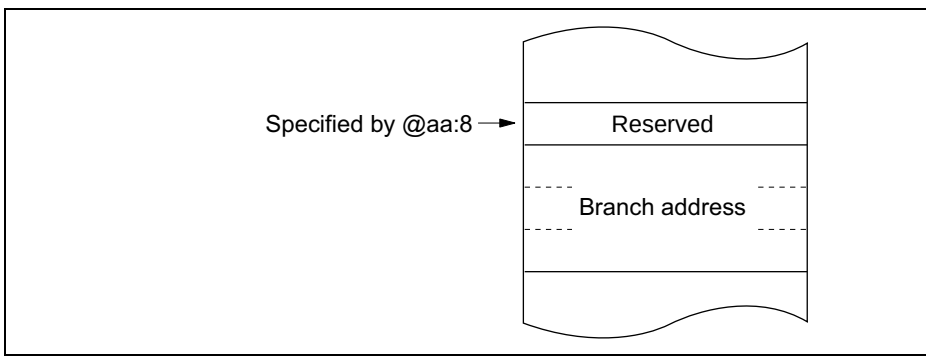
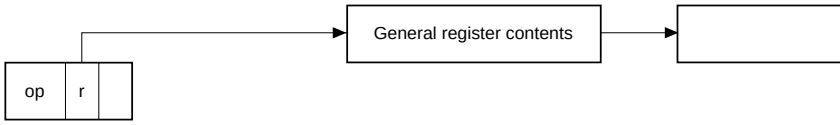
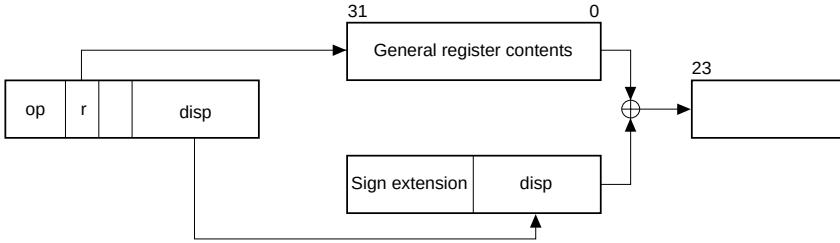


Figure 2.10 Memory-Indirect Branch Address Specification

1-Mbyte operating modes the upper 4 bits of the calculated address are ignored in order
generate a 20-bit effective address.

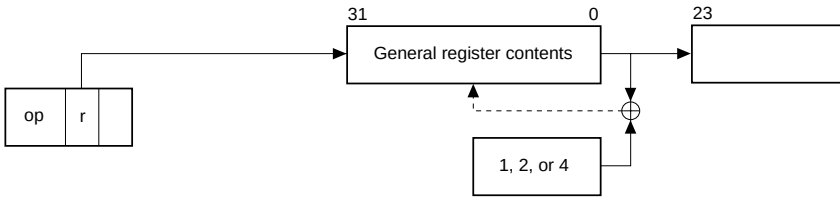


3 Register indirect with displacement
@(d:16, ERn)/@(d:24, ERn)

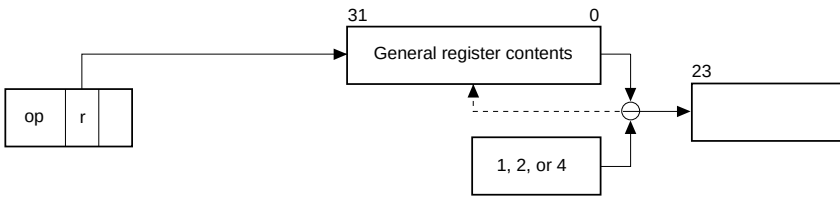


4. Register indirect with post-increment
or pre-decrement

- Register indirect with post-increment @ERn+

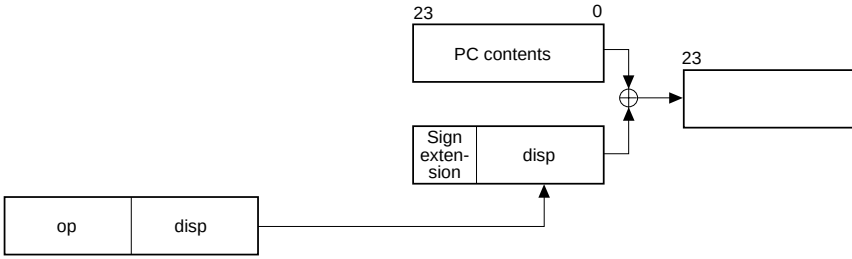


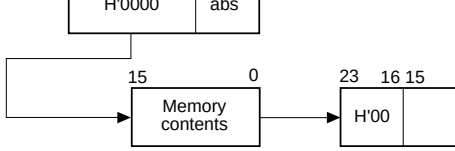
- Register indirect with pre-decrement @–ERn



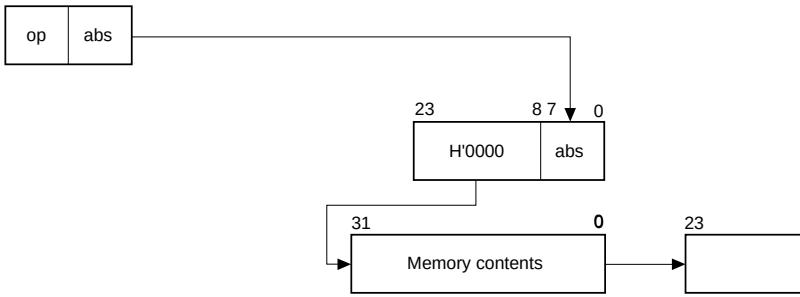
1 for a byte operand, 2 for a word operand, 4 for a longword operand

@aa:24

6	Immediate #xx:8, #xx:16, or #xx:32 	Operand is imm data
7	Program-counter relative @(d:8, PC) or @(d:16, PC) 	



- Advanced mode



Legend:

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

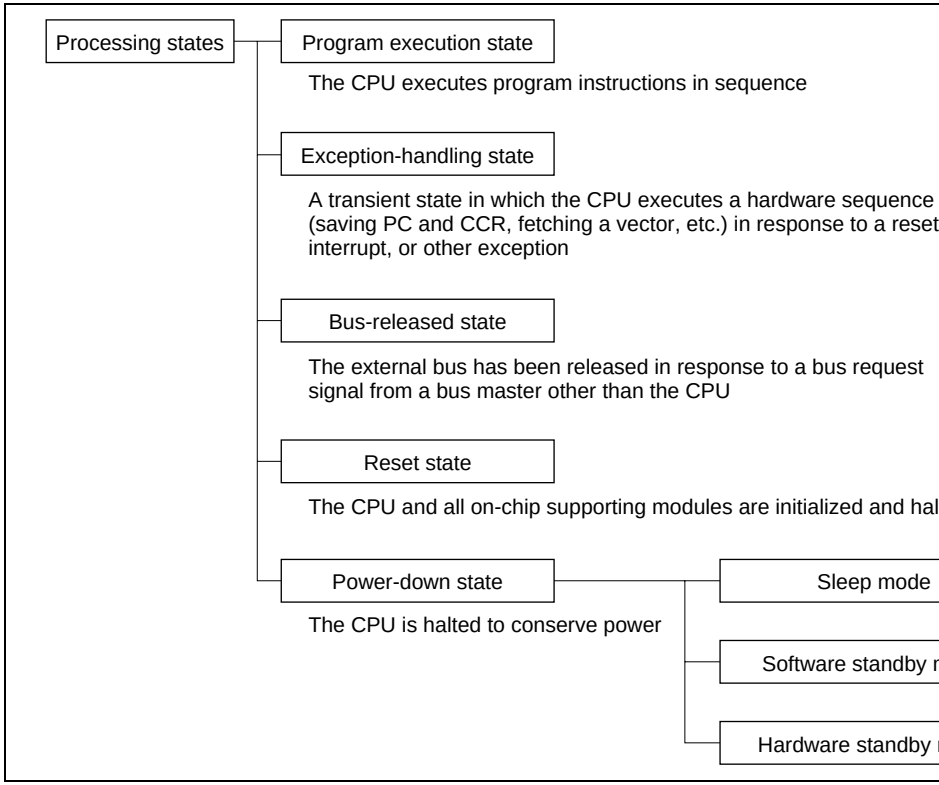


Figure 2.11 Processing States

2.8.2 Program Execution State

In this state the CPU executes program instructions in normal sequence.

Types of Exception Handling and Their Priority: Exception handling is performed
interrupts, and trap instructions. Table 2.14 indicates the types of exception handling a
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 Handli
High	Reset	Synchronized with clock	Exception handling starts i when $\overline{\text{RES}}$ changes from lo
	Interrupt	End of instruction execution or end of exception handling*	When an interrupt is reque exception handling starts a the current instruction or cu exception-handling sequen
Low	Trap instruction	When TRAPA instruction is executed	Exception handling starts w (TRAPA) instruction is exe

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

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

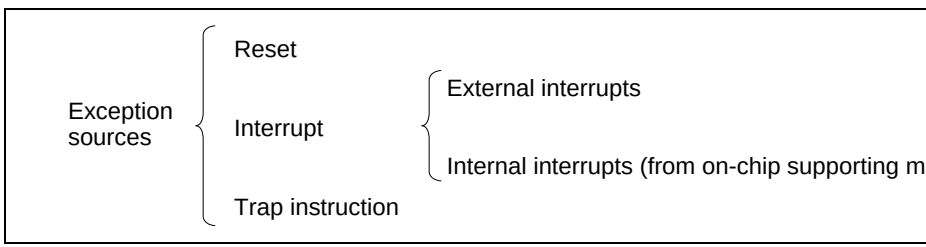
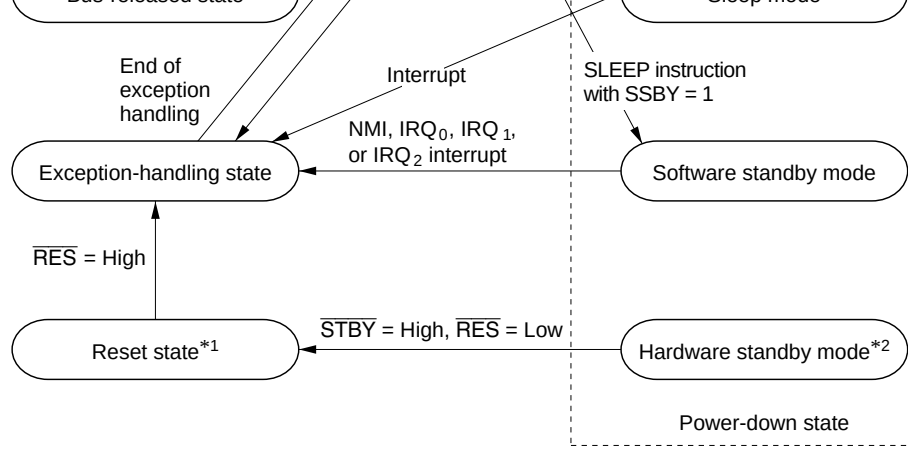


Figure 2.12 Classification of Exception Sources



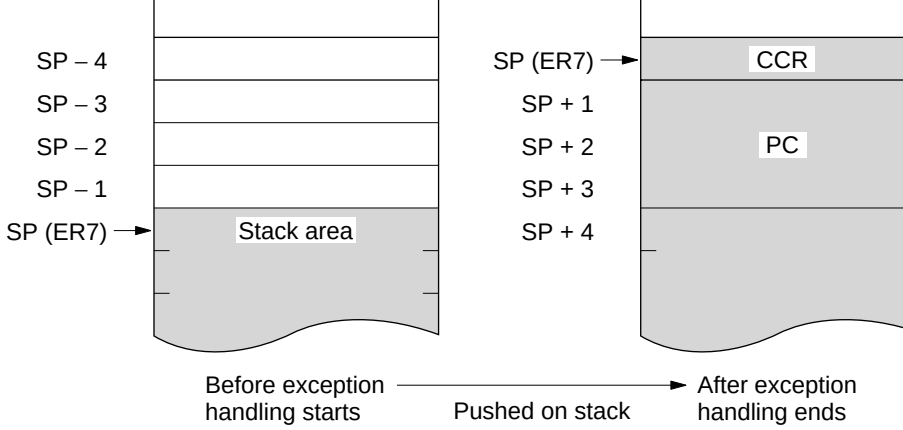
Notes: 1. From any state except hardware standby mode, a transition to the reset state occurs whenever $\overline{RES}$ goes low.
2. From any state, a transition to hardware standby mode occurs when $\overline{STBY}$ goes low.

Figure 2.13 State Transitions

2.8.4 Exception-Handling Sequences

Reset Exception Handling: Reset exception handling has the highest priority. The res entered when the $\overline{RES}$ signal goes low. Reset exception handling starts after that, when changes from low to high. When reset exception handling starts the CPU fetches a start from the exception vector table and starts program execution from that address. All inte including NMI, are disabled during the reset exception-handling sequence and immedia ends.

Interrupt Exception Handling and Trap Instruction Exception Handling: When th exception-handling sequences begin, the CPU references the stack pointer (ER7) and p 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. I



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

2.8.5 Bus-Released State

In this state the bus is released to a bus master other than the CPU, in response to a bu
The bus masters other than the CPU are the DMA controller, the refresh controller, an
bus master. While the bus is released, the CPU halts except for internal operations. Int
requests are not accepted. For details see section 6.3.7, Bus Arbiter Operation.

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 v
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 register
retained.

Software Standby Mode: A transition to software standby mode is made if the SLEEP
instruction is executed while the SSBY bit is set to 1 in SYSCR. The CPU and clock ha
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 re
The I/O ports also remain in their existing states.

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

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

controlled by the bus controller.

2.9.2 On-Chip Memory Access Timing

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

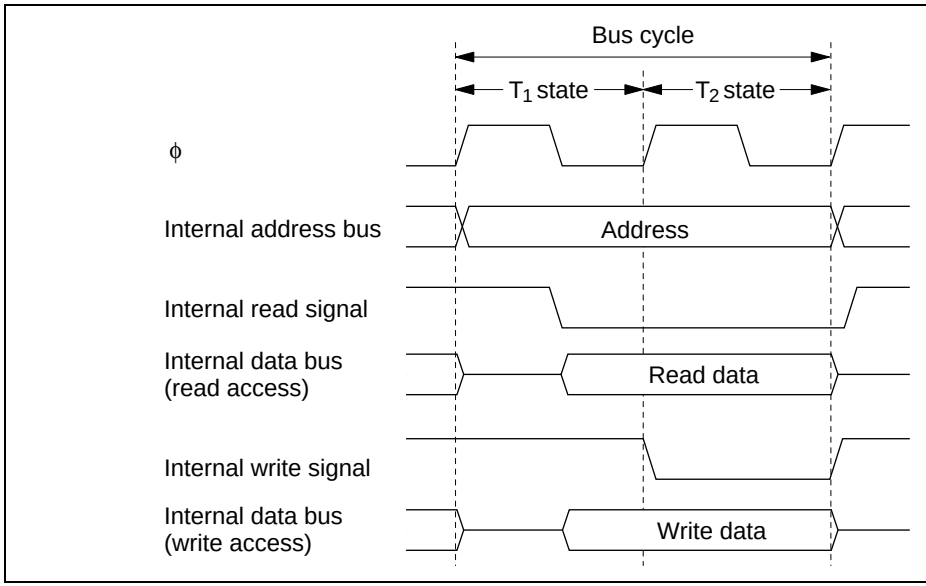


Figure 2.15 On-Chip Memory Access Cycle

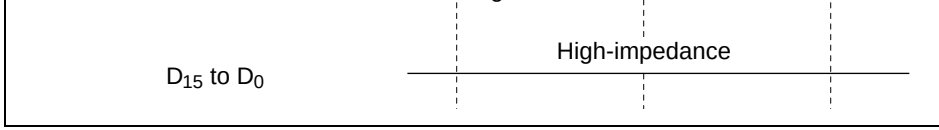


Figure 2.16 Pin States during On-Chip Memory Access

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 bit depending on the register being accessed. Figure 2.17 shows the on-chip supporting mo timing. Figure 2.18 indicates the pin states.

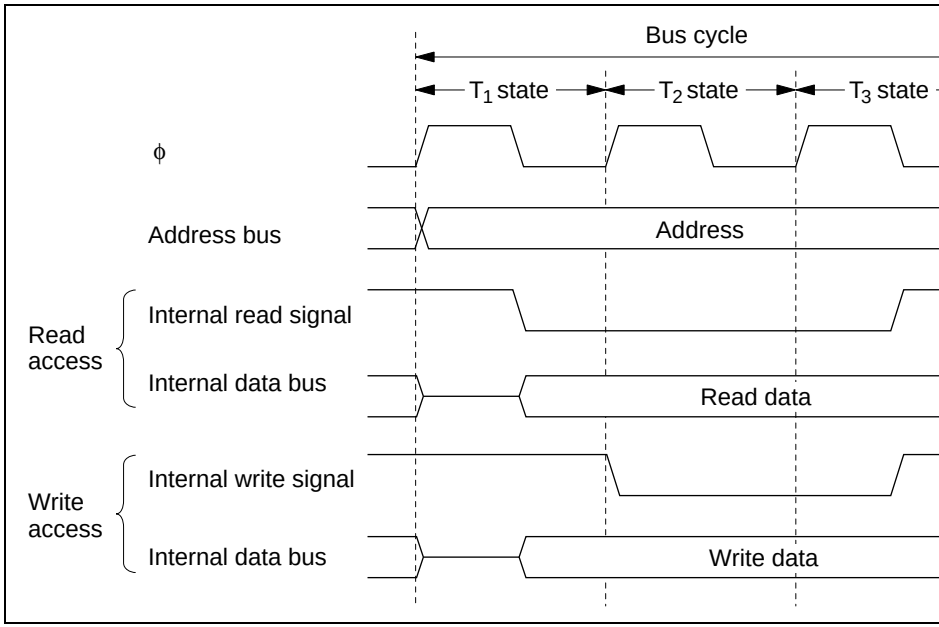


Figure 2.17 Access Cycle for On-Chip Supporting Modules

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

2.9.4 Access to External Address Space

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

pins (MD_2 to MD_0) as indicated in table 3.1. The input at these pins determines the siz
address 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	Expanded mode	8 bits	Enabled
Mode 7	1	1	1	Single-chip advanced mode	—	Enabled

Notes: 1. In modes 1 to 6, 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
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 two choices: 1 Mbyte or 16 Mbytes. The external
either 8 or 16 bits wide depending on ABWCR settings. If 8-bit access is selected for
external data bus is 8 bits wide. For details see section 6, Bus Controller.

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

3.1.2 Register Configuration

The H8/3048B Group has a mode control register (MDCR) that indicates the inputs at t
pins (MD_2 to MD_0), and a system control register (SYSCR). Table 3.2 summarizes thes

Table 3.2 Registers

Address*	Name	Abbreviation	R/W	Initial V
H'FFF1	Mode control register	MDCR	R	Undete
H'FFF2	System control register	SYSCR	R/W	H'0B

Note: * The lower 16 bits of the address are indicated.

3.2 Mode Control Register (MDCR)

MDCR is an 8-bit read-only register that indicates the current operating mode of the H8
Group.

Bit	7	6	5	4	3	2	1
	—	—	—	—	—	MDS2	MDS1
Initial value	1	1	0	0	0	—*	—*
Read/Write	—	—	—	—	—	R	R

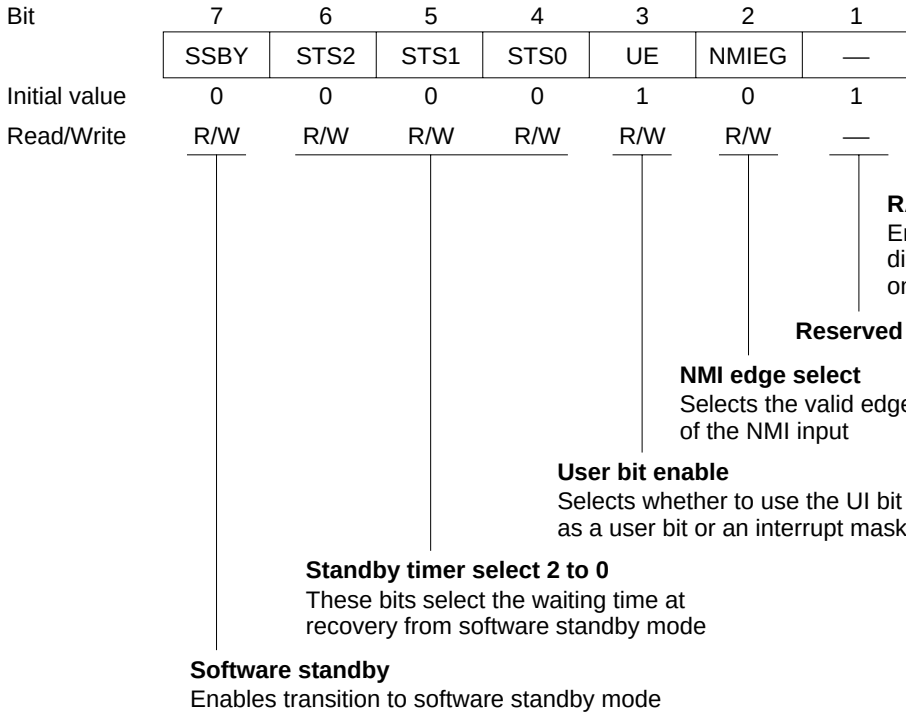
Reserved bits **Reserved bits** **Mode select 2 to**
Bits indicating the
operating mode

Note: * Determined by pins MD_2 to MD_0.

Note: For the flash memory version with single power supply (H8/3048F-ONE), fla
can be written to in the boot mode. In the boot mode, the inverted value of the
is set to bit MDS2.

3.3 System Control Register (SYSCR)

SYSCR is an 8-bit register that controls the operation of the H8/3048B Group.



1	SLEEP instruction causes transition to software standby mode	

Bits 6 to 4—Standby Timer Select (STS2 to STS0): These bits select the length of tim and on-chip supporting modules wait for the internal clock oscillator to settle when soft standby mode is exited by an external interrupt. When using a crystal oscillator, set thes that the waiting time will be at least 7 ms at the system clock rate. For further informati waiting time selection, see section 20.4.3, Selection of Waiting Time for Exit from Soft Standby Mode.

Bit 6: STS2	Bit 5: STS1	Bit 4: STS0	Description	
0	0	0	Waiting time = 8,192 states	(I
		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 3—User Bit Enable (UE): Selects whether to use the UI bit in the condition code r 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	(In

Bit 0—RAM Enable (RAME): Enables or disables the on-chip RAM. The RAME bi
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 (

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 are
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-
all areas. If at least one area is designated for 16-bit access in ABWCR, the bus mode
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.)

3.4.5 Mode 5

Ports 1, 2, and 5 can function as address pins A_{19} to A_0, permitting access to a maximun address space, but following a reset they are input ports. To use ports 1, 2, and 5 as an a the corresponding bits in their data direction registers (P1DDR, P2DDR, and P5DDR) r to 1. The initial bus mode after a reset is 8 bits, with 8-bit access to all areas. If at least designated for 16-bit access in ABWCR, the bus mode switches to 16 bits.

3.4.6 Mode 6

Ports 1, 2, and 5 and part of port A function as address pins A_{23} to A_0, permitting access maximum 16-Mbyte address space, but following a reset they are input ports. To use po and 5 as an address bus, the corresponding bits in their data direction registers (P1DDR and P5DDR) must be set to 1. For A_{23} to A_{21} output, clear bits 7 to 5 of BRCR to 0. (In t A_{20} is always used for address output.)

The initial bus mode after a reset is 8 bits, with 8-bit access to all areas. If at least one a designated for 16-bit access in ABWCR, the bus mode switches to 16 bits.

3.4.7 Mode 7

This mode operates using the on-chip ROM, RAM, and internal I/O registers. All I/O p available. 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 $P2_0$*2	$P2_7$ to $P2_0$*2
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	D_{15} to D_8
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 $P4_0$*1
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 $P5_0$*2
Port A	PA_7 to PA_4	PA_7 to PA_4	PA_7 to PA_5*3, A_{20}	PA_7 to PA_5*3, A_{20}	PA_7 to PA_4	PA_7 to PA_5, A_{20}*3

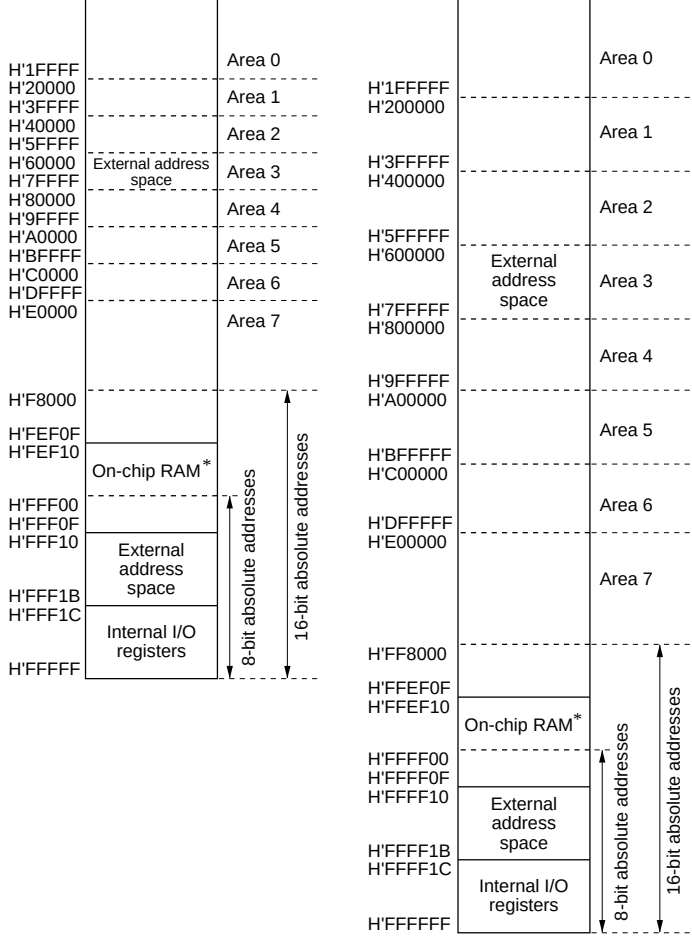
Notes: 1. Initial state. The bus mode can be switched by settings in ABWCR. These p
as $P4_7$ to $P4_0$ in 8-bit bus mode, and as D_7 to D_0 in 16-bit bus mode.
2. Initial state. These pins become address output pins when the correspondi
data direction registers (P1DDR, P2DDR, P5DDR) are set to 1.
3. Initial state. A_{20} is always an address output pin. PA_7 to PA_5 are switched ov
A_{21} output by writing 0 in bits 7 to 5 of BRCR.

3.6 Memory Map in Each Operating Mode

Figure 3.1 shows a memory map of the H8/3048B Group. The address space is divide
areas.

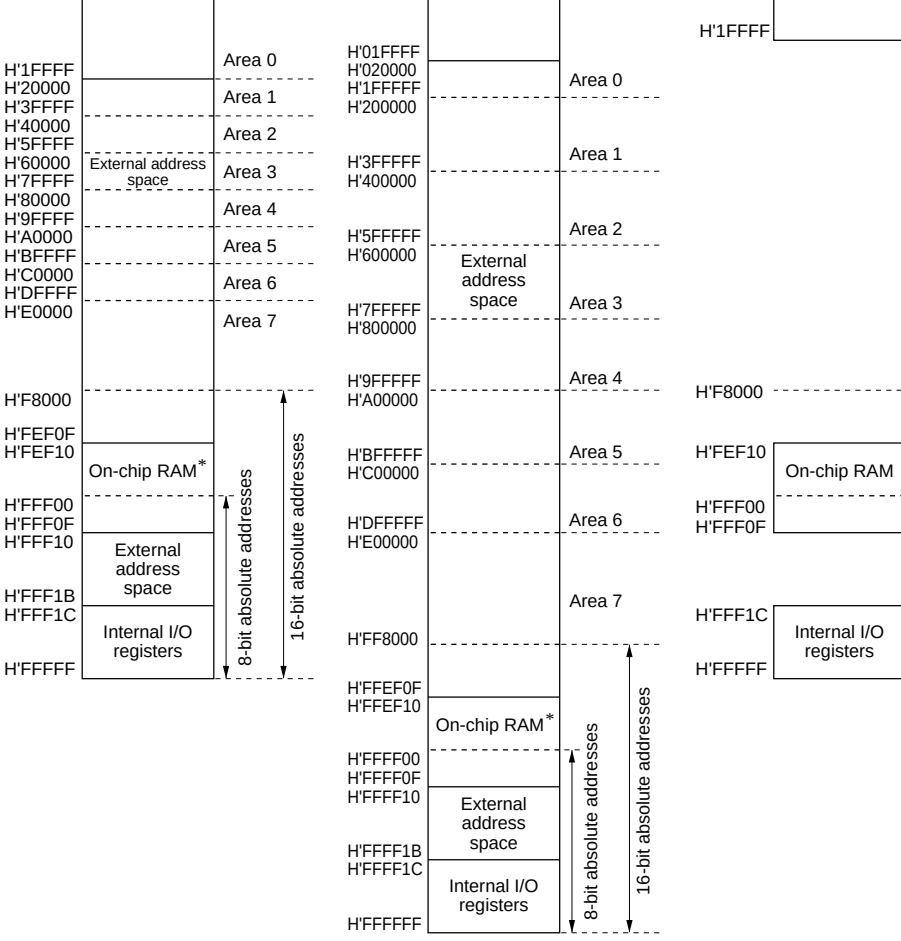
The initial bus mode differs between modes 1 and 2, and also between modes 3 and 4.

The address locations of the on-chip RAM and internal I/O registers differ between th
modes (modes 1, 2, 5, and 7) and 16-Mbyte modes (modes 3, 4, and 6). The address ra
specifiable by the CPU in the 8- and 16-bit absolute addressing modes (@aa:8 and @a
differs.



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

Figure 3.1 H8/3048B Group Memory Map in Each Operating Mode



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

Figure 3.1 H8/3048B Group Memory Map in Each Operating Mode (co

Exception handling is prioritized as shown in table 4.1. If two or more exceptions occu
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 transiti RES pin
	Interrupt	Interrupt requests are handled when executio current instruction or handling of the current e completed
Low	Trap instruction (TRAPA)	Started by execution of a trap instruction (TR

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 the address indicated in that address.

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

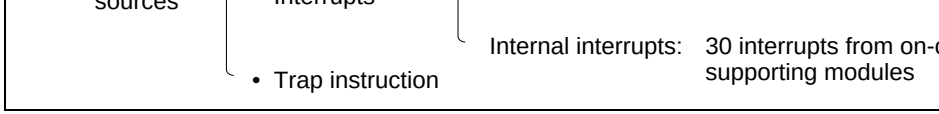


Figure 4.1 Exception Sources

	5	H'0014 to H'0017
	6	H'0018 to H'001B
External interrupt (NMI)	7	H'001C to H'001F
Trap instruction (4 sources)	8	H'0020 to H'0023
	9	H'0024 to H'0027
	10	H'0028 to H'002B
	11	H'002C to H'002F
External interrupt IRQ_0	12	H'0030 to H'0033
External interrupt IRQ_1	13	H'0034 to H'0037
External interrupt IRQ_2	14	H'0038 to H'003B
External interrupt IRQ_3	15	H'003C to H'003F
External interrupt IRQ_4	16	H'0040 to H'0043
External interrupt IRQ_5	17	H'0044 to H'0047
Reserved for system use	18	H'0048 to H'004B
	19	H'004C to H'004F
Internal interrupts*2	20 to 60	H'0050 to H'0053 to H'00F0 to H'00F3

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.

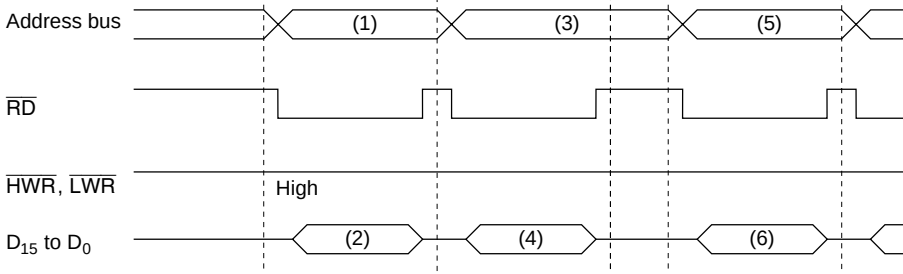
To ensure that the chip is reset properly, hold the $\overline{\text{RES}}$ pin low for at least 20 ms at pow
reset the chip during operation, hold the $\overline{\text{RES}}$ pin low for at least 20 system clock (ϕ) cy
appendix D.2, Pin States at Reset, for the states of the pins in the reset state.

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) are read, and program e
  starts from the address indicated in 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 modes 5, 6, and 7.

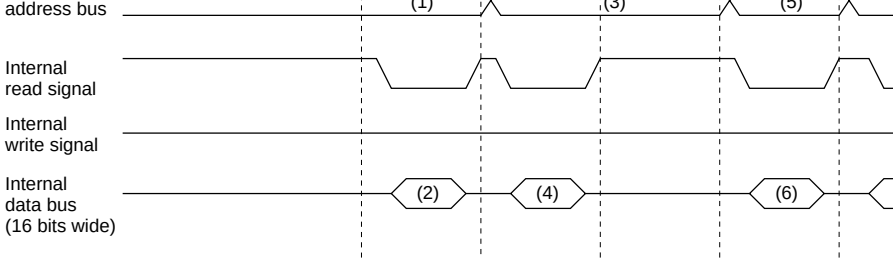
Figure 4.2 Reset Sequence (Modes 1 and 3)



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

Figure 4.4 Reset Sequence (Modes 5, 6, and 7)

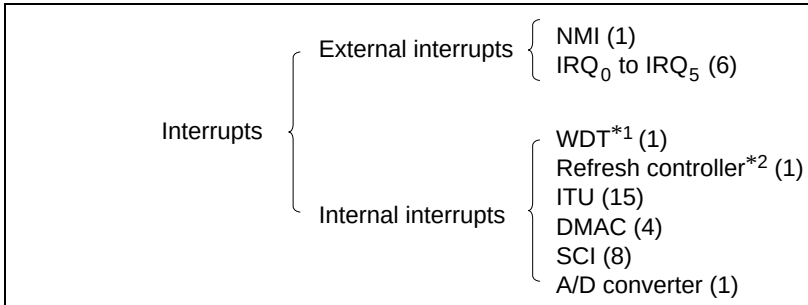
4.2.3 Interrupts after Reset

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.

For details on interrupts see section 5, Interrupt Controller.

Note: * For the H8/3048F-ONE (single power supply with flash memory), the NMI
be prohibited. For details, refer to section 18.8.4, NMI Input Disable Condit

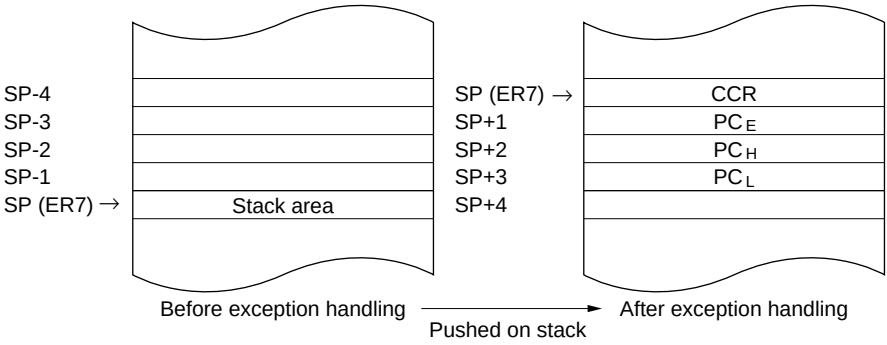


Notes: Numbers in parentheses are the number of interrupt sources.
1. When the watchdog timer is used as an interval timer, it generates an int
request at every counter overflow.
2. When the refresh controller is used as an interval timer, it generates an i
request at compare match.

Figure 4.5 Interrupt Sources and Number of Interrupts

4.5 Stack Status after Exception Handling

Figure 4.6 shows the stack after completion of trap instruction exception handling and exception handling.



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

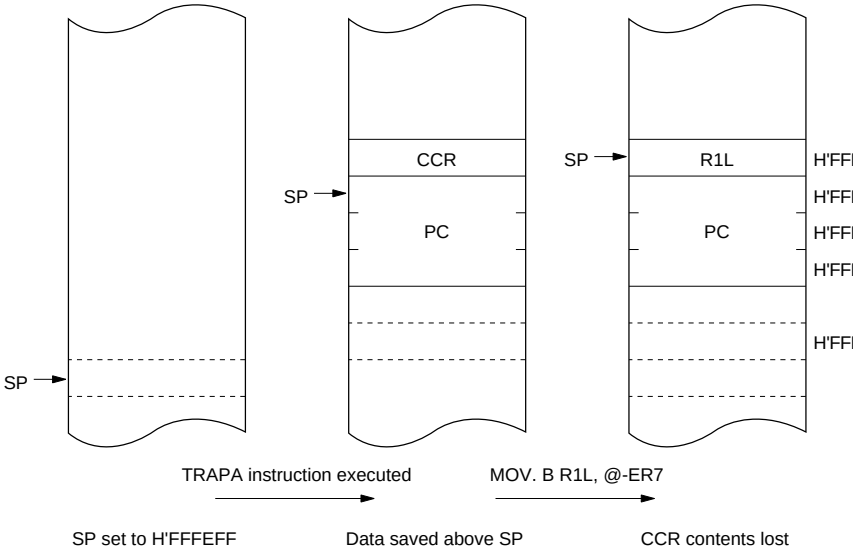
Notes: 1. PC indicates the address of the first instruction that will be executed after return.
2. Registers must be saved in word or longword size at even addresses.

Figure 4.6 Stack after Completion of Exception Handling

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

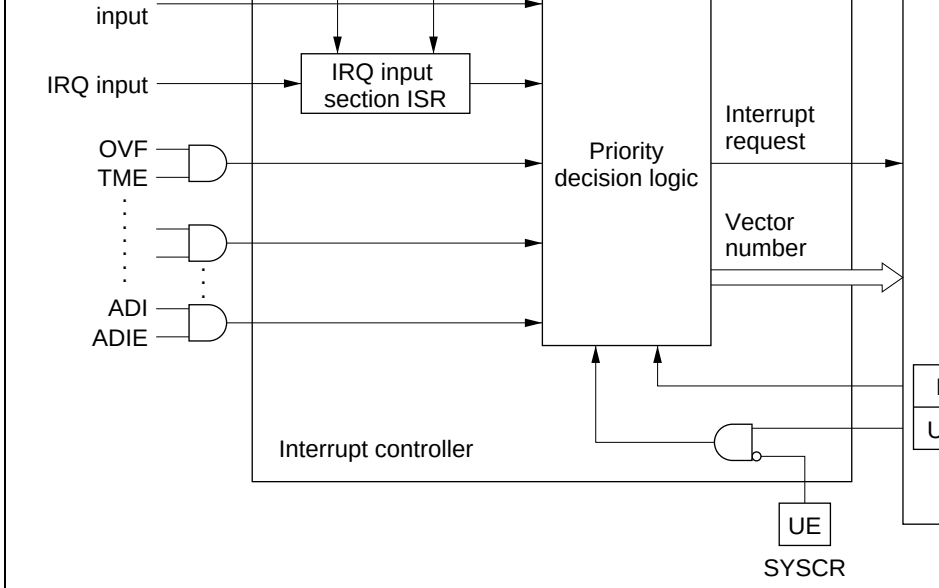
- Interrupt priority registers (IPRs) for setting interrupt priorities

  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
- Independent vector addresses

  All interrupts are independently vectored; the interrupt service routine does not ha
  identify the interrupt source.
- 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: * For the H8/3048F-ONE (single power supply with flash memory), the NM
be prohibited. For details, refer to section 18.8.4, NMI Input Disable Cond



Legend:

ISCR:	IRQ sense control register
IER:	IRQ enable register
ISR:	IRQ status register
IPRA:	Interrupt priority register A
IPRB:	Interrupt priority register B
SYSCR:	System control register

Figure 5.1 Interrupt Controller Block Diagram

External interrupt request 5 to 0	$\overline{IRQ}_5$ to $\overline{IRQ}_0$	Input	Maskable interrupts, falling level sensing selectable

Note: * For the H8/3048F-ONE (single power supply with flash memory), the NMI i prohibited. For details, refer to section 18.8.4, NMI Input Disable Conditions

5.1.4 Register Configuration

Table 5.2 lists the registers of the interrupt controller.

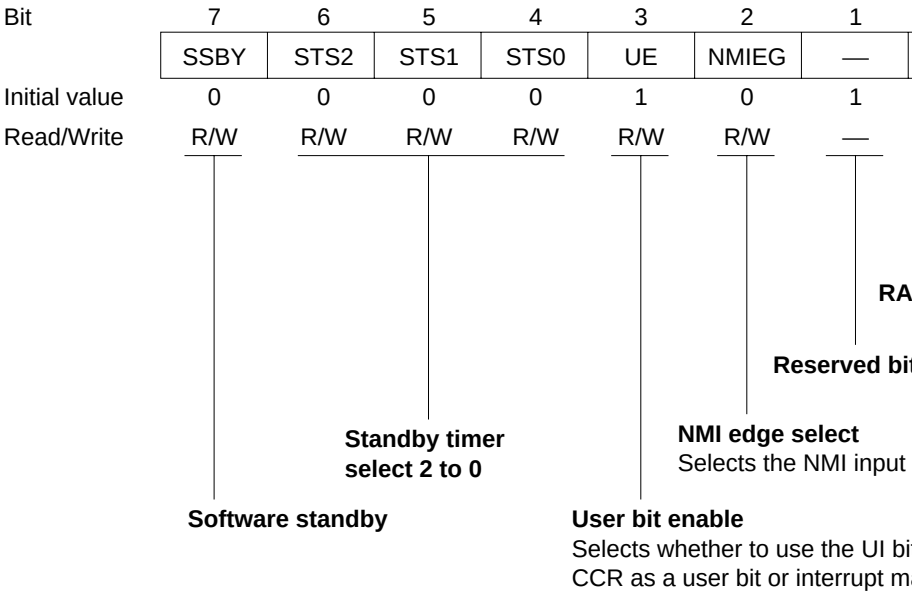
Table 5.2 Interrupt Controller Registers

Address*1	Name	Abbreviation	R/W	In
H'FFF2	System control register	SYSCR	R/W	H'
H'FFF4	IRQ sense control register	ISCR	R/W	H'
H'FFF5	IRQ enable register	IER	R/W	H'
H'FFF6	IRQ status register	ISR	R/(W)*2	H'
H'FFF8	Interrupt priority register A	IPRA	R/W	H'
H'FFF9	Interrupt priority register B	IPRB	R/W	H'

Notes: 1. Lower 16 bits of the address.
2. Only 0 can be written, to clear flags.

(SYSCR).

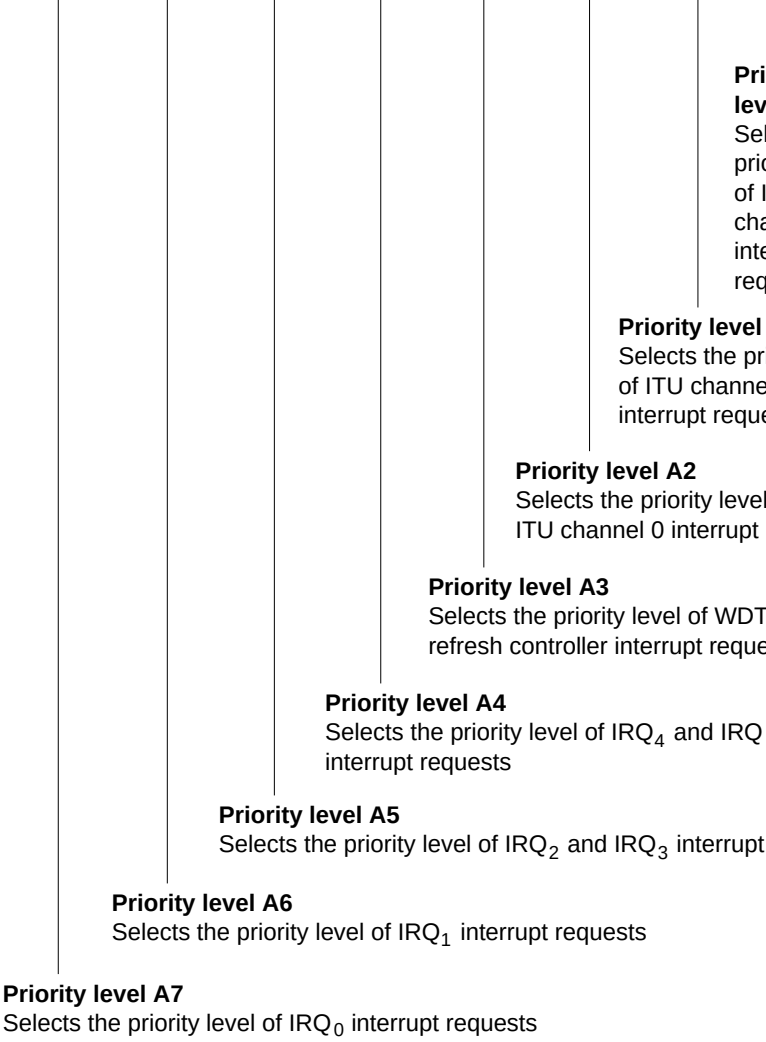
SYSCR is initialized to H'0B by a reset and in hardware standby mode. It is not initializ
software standby mode.



Bit 2: NMIEG	Description
0	Interrupt is requested at falling edge of NMI input
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.



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) (
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 interr

Bit 5: IPRA5	Description
0	IRQ_2 and IRQ_3 interrupt requests have priority level 0 (low priority) (
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 interr

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

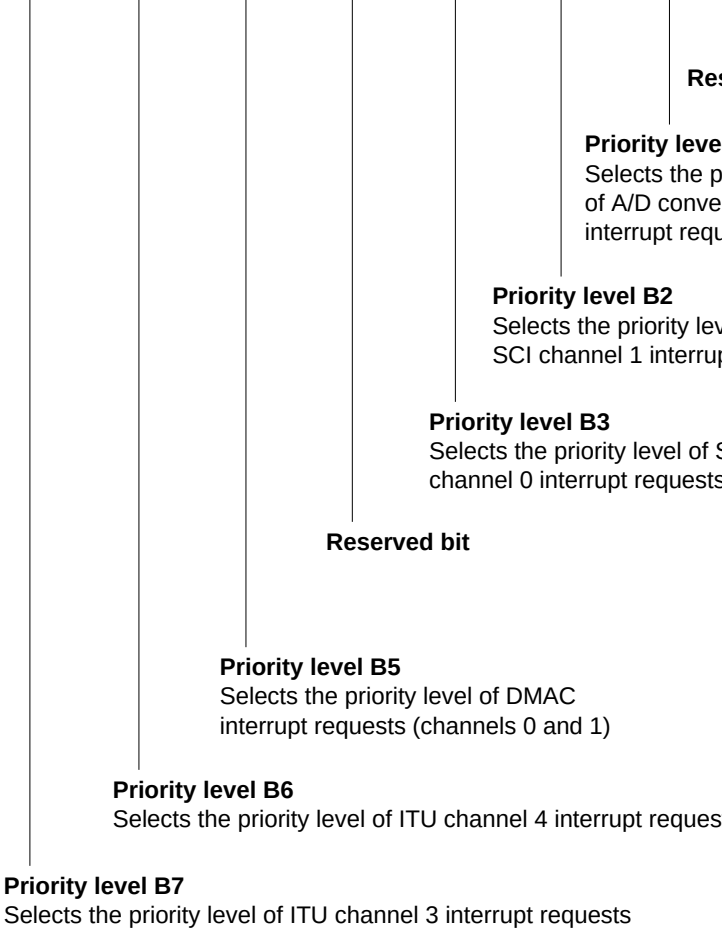
Bit 3—Priority Level A3 (IPRA3): Selects the priority level of WDT and refresh co interrupt requests.

Bit 3: IPRA3	Description
0	WDT and refresh controller interrupt requests have priority level 0 (
1	WDT and refresh controller interrupt requests have priority level 1 priority)

Bit 1: IPRA1	Description
0	ITU channel 1 interrupt requests have priority level 0 (low priority) (I
1	ITU channel 1 interrupt requests have priority level 1 (high priority)

Bit 0—Priority Level A0 (IPRA0): Selects the priority level of ITU channel 2 interru

Bit 0: IPRA0	Description
0	ITU channel 2 interrupt requests have priority level 0 (low priority) (I
1	ITU 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: IPRB6	Description
0	ITU channel 4 interrupt requests have priority level 0 (low priority) (I
1	ITU channel 4 interrupt requests have priority level 1 (high priority)

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

Bit 5: IPRB5	Description
0	DMAC interrupt requests (channels 0 and 1) have priority level 0 (lo (I
1	DMAC interrupt requests (channels 0 and 1) have priority level 1 (hi

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

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

Bit 1—Priority Level B1 (IPRB1): Selects the priority level of A/D converter interru

Bit 1: IPRB1	Description
0	A/D converter interrupt requests have priority level 0 (low priority) (
1	A/D converter interrupt requests have priority level 1 (high priority)

Bit 0—Reserved: This bit can be written and read, but it does not affect interrupt prio

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.

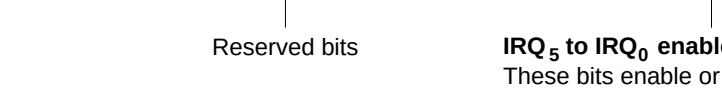
ISR is initialized to H'00 by a reset and in hardware standby mode.

Bits 7 and 6—Reserved: Read-only bits, 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, $\overline{\text{IRQn}}$ input is high, and interrupt exception handling is
	IRQnSC = 1 and IRQn interrupt exception handling is carried out.
1	[Setting conditions]
	IRQnSC = 0 and $\overline{\text{IRQn}}$ input is low.
	IRQnSC = 1 and $\overline{\text{IRQn}}$ input changes from high to low.

Note: n = 5 to 0



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_0 interrupts.

Bits 5 to 0: IRQ5E to IRQ0E	Description
0	IRQ_5 to IRQ_0 interrupts are disabled (Initial value)
1	IRQ_5 to IRQ_0 interrupts are enabled

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

Reserved bits

IRQ$_5$ to IRQ$_0$ sense control
These bits select level sensing or fall
sensing for IRQ$_5$ to IRQ$_0$ interrupts

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 $\overline{\text{IRQ}}_5$ to $\overline{\text{IRQ}}_0$, or by fallin
sensing.

Bits 5 to 0: IRQ5SC to IRQ0SC	Description
0	Interrupts are requested when $\overline{\text{IRQ}}_5$ to $\overline{\text{IRQ}}_0$ inputs are low (I
1	Interrupts are requested by falling-edge input at $\overline{\text{IRQ}}_5$ to $\overline{\text{IRQ}}_0$

NMI

NMI is the highest-priority interrupt and is always accepted, regardless of the states of bits in CCR. The NMIEG bit in SYSCR selects whether an interrupt is requested by the falling edge of the input at the NMI pin*. NMI interrupt exception handling has vector

Note: * For the H8/3048F-ONE (single power supply with flash memory), the NMI be prohibited. For details, refer to section 18.8.4, NMI Input Disable Condi

IRQ_0 to IRQ_5 Interrupts

These interrupts are requested by input signals at pins $\overline{IRQ}_0$ to $\overline{IRQ}_5$. The IRQ_0 to IRQ_5 have the following features.

- ISCR settings can select whether an interrupt is requested by the low level of the in $\overline{IRQ}_0$ to $\overline{IRQ}_5$, or by the falling edge.
- IER settings can enable or disable the IRQ_0 to IRQ_5 interrupts. Interrupt priority lev 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 can to 0 by software.

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

Note: n = 5 to 0

Figure 5.2 Block Diagram of Interrupts IRQ_0 to IRQ_5

Figure 5.3 shows the timing of the setting of the interrupt flags (IRQnF).

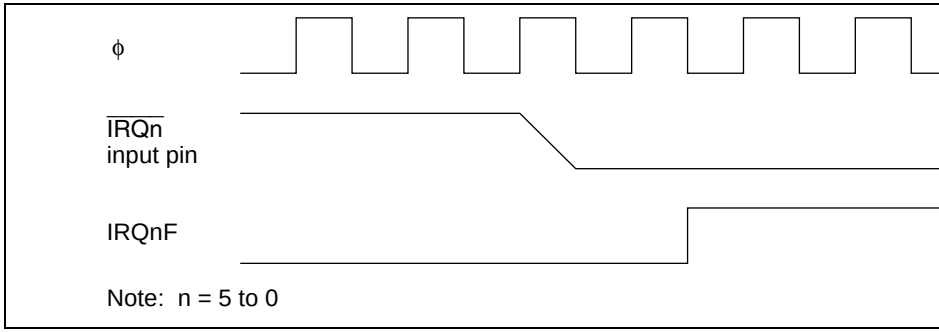


Figure 5.3 Timing of Setting of IRQnF

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 or output.

sent to the interrupt controller, and the I and UI bits are disregarded.

5.3.3 Interrupt Vector Table

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

IRQ_3		15	H'003C to H'003F	
IRQ_4		16	H'0040 to H'0043	IPRA4
IRQ_5		17	H'0044 to H'0047	
Reserved	—	18	H'0048 to H'004B	
		19	H'004C to H'004F	
WOVI (interval timer)	Watchdog timer	20	H'0050 to H'0053	IPRA3
CMI (compare match)	Refresh controller	21	H'0054 to H'0057	
Reserved	—	22	H'0058 to H'005B	
		23	H'005C to H'005F	
IMIA0 (compare match/ input capture A0)	ITU channel 0	24	H'0060 to H'0063	IPRA2
IMIB0 (compare match/ input capture B0)		25	H'0064 to H'0067	
OVI0 (overflow 0)		26	H'0068 to H'006B	
Reserved	—	27	H'006C to H'006F	
IMIA1 (compare match/ input capture A1)	ITU channel 1	28	H'0070 to H'0073	IPRA1
IMIB1 (compare match/ input capture B1)		29	H'0074 to H'0077	
OVI1 (overflow 1)		30	H'0078 to H'007B	
Reserved	—	31	H'007C to H'007F	

Reserved	—	35	H'008C to H'008F	
IMIA3 (compare match/ input capture A3)	ITU channel 3	36	H'0090 to H'0093	IPRB7
IMIB3 (compare match/ input capture B3)		37	H'0094 to H'0097	
OVI3 (overflow 3)		38	H'0098 to H'009B	
Reserved	—	39	H'009C to H'009F	
IMIA4 (compare match/ input capture A4)	ITU channel 4	40	H'00A0 to H'00A3	IPRB6
IMIB4 (compare match/ input capture B4)		41	H'00A4 to H'00A7	
OVI4 (overflow 4)		42	H'00A8 to H'00AB	
Reserved	—	43	H'00AC to H'00AF	
DEND0A	DMAC	44	H'00B0 to H'00B3	IPRB5
DEND0B		45	H'00B4 to H'00B7	
DEND1A		46	H'00B8 to H'00BB	
DEND1B		47	H'00BC to H'00BF	
Reserved	—	48	H'00C0 to H'00C3	—
		49	H'00C4 to H'00C7	
		50	H'00C8 to H'00CB	
		51	H'00CC to H'00CF	

TEI0 (transmit end 0)		55	H'00DC to H'00DF	
ERI1 (receive error 1)	SCI channel 1	56	H'00E0 to H'00E3	IPRB2
RXI1 (receive data full 1)		57	H'00E4 to H'00E7	
TXI1 (transmit data empty 1)		58	H'00E8 to H'00EB	
TEI1 (transmit end 1)		59	H'00EC to H'00EF	
ADI (A/D end)	A/D	60	H'00F0 to H'00F3	IPRB1

Note: * Lower 16 bits of the address.

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

Note: * For the H8/3048F-ONE (single power supply with flash memory), the NM
be prohibited. For details, refer to section 18.8.4, NMI Input Disable Cond

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 h priority.
	1	—	No interrupts are accepted except NMI.
0	0	—	All interrupts are accepted. Interrupts with priority level 1 h 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 can all be
the I bit in the CPU's CCR. Interrupts are masked when the I bit is set to 1, and unmas
the I bit is cleared to 0. Interrupts with priority level 1 have higher priority. Figure 5.4
flowchart showing how interrupts are accepted when UE = 1.

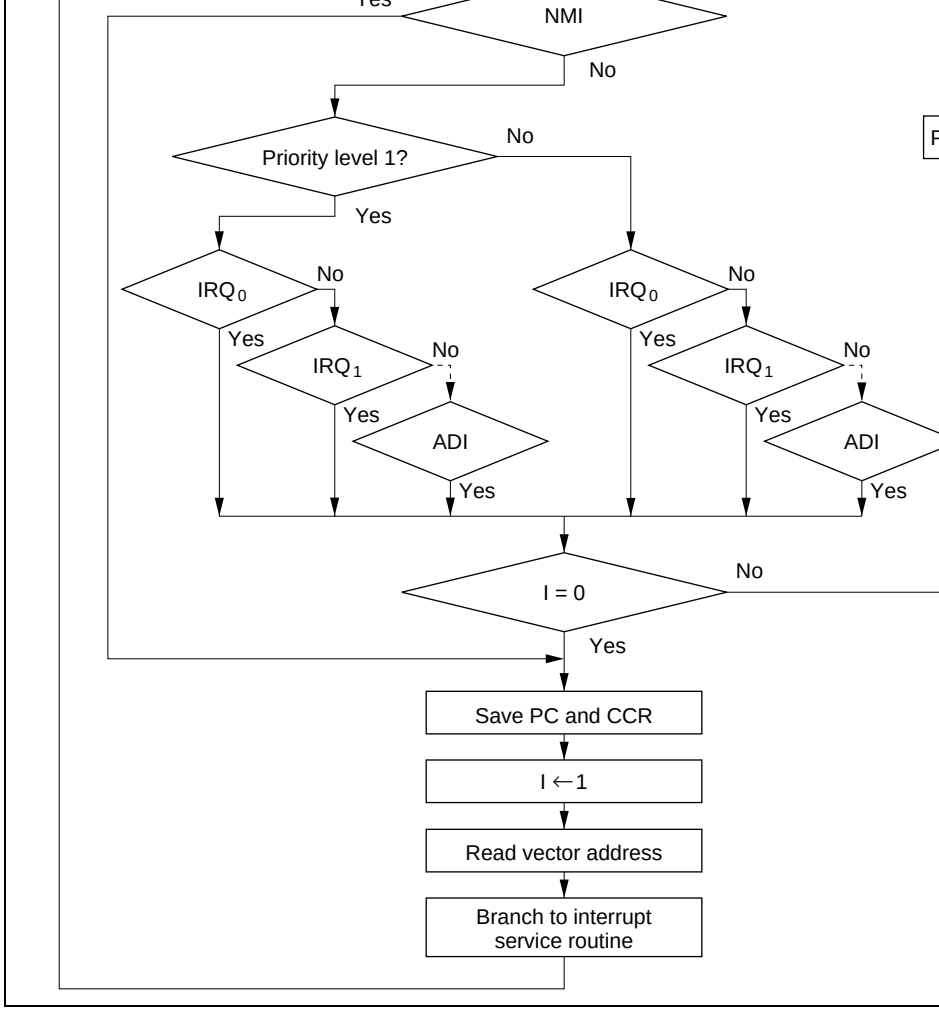


Figure 5.4 Process Up to Interrupt Acceptance when UE = 1

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.
- 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 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 masking of IR 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 ar when the I bit is cleared to 0.
- Interrupt requests with priority level 1 are masked when the I and UI bits are both 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 H'20, and IPRB is set to H'00 (giving IRQ_2 and IRQ_3 interrupt requests priority ov 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 shows the transitions among the above states.

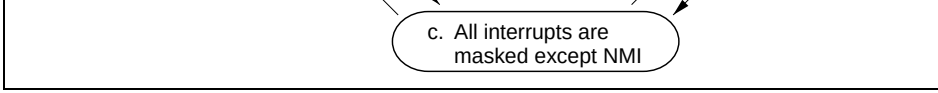


Figure 5.5 Interrupt Masking State Transitions (Example)

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 1 interrupt request is sent to the interrupt controller.
- When the interrupt controller receives one or more interrupt requests, it selects the h priority request, following the IPR interrupt priority settings, and holds other reques If two or more interrupts with the same IPR setting are requested simultaneously, th 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 inter is accepted regardless of its IPR setting, and regardless of the UI bit. If the I bit is se the UI bit is cleared to 0, only NMI and interrupts with priority level 1 are accepted 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 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.
- 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 r starts executing from the address indicated by the contents of the vector address.

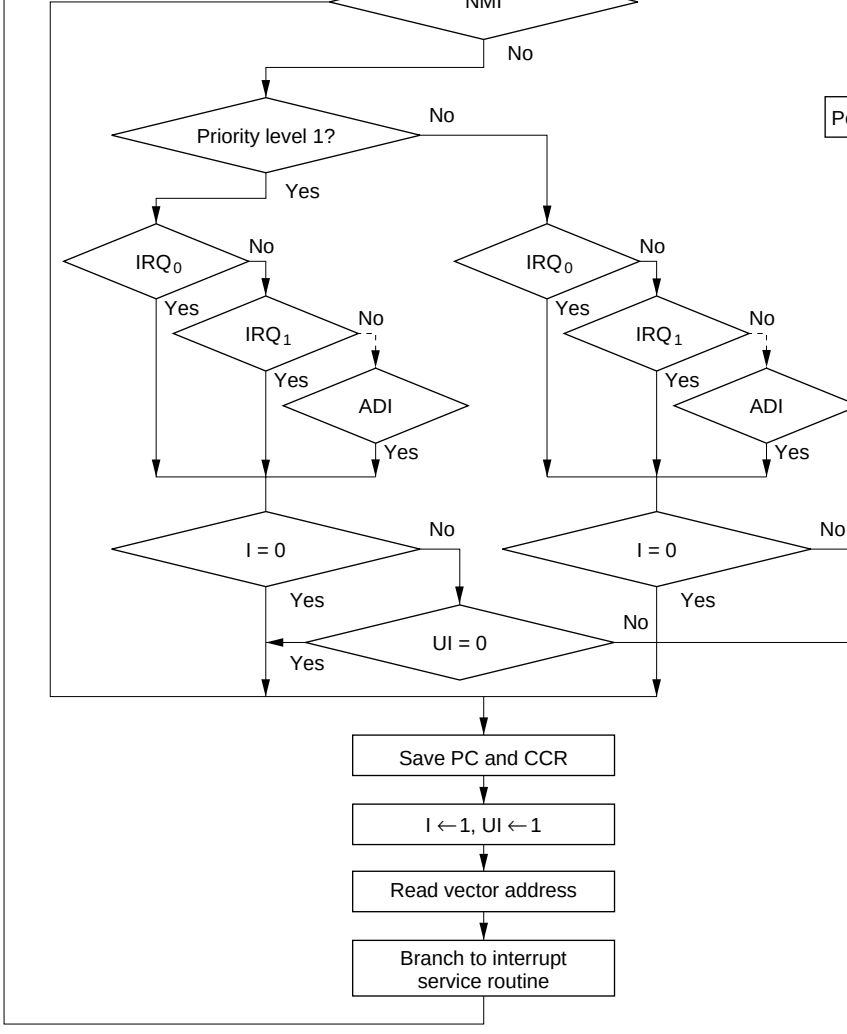


Figure 5.6 Process Up to Interrupt Acceptance when UE = 0

Figure 5.7 Interrupt Sequence (Mode 2, Two-State Access, Stack in External M

No.	Item	On-Chip Memory	2 States	3 States	2 States
1	Interrupt priority decision	2*1	2*1	2*1	2*1
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
3	Saving PC and CCR to stack	4	8	12*4	4
4	Vector fetch	4	8	12*4	4
5	Instruction prefetch*2	4	8	12*4	4
6	Internal processing*3	4	4	4	4
Total		19 to 41	31 to 57	43 to 83	19 to 41

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 a
4. The number of states increases if wait states are inserted in external memo
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 is carried out. If a higher-priority interrupt is also requested, however, interrup
handling for the higher-priority interrupt is carried out, and the lower-priority interrupt
This also applies to the clearing of an interrupt flag.

Figure 5.8 shows an example in which an IMIEA bit is cleared to 0 in TIER of the ITU

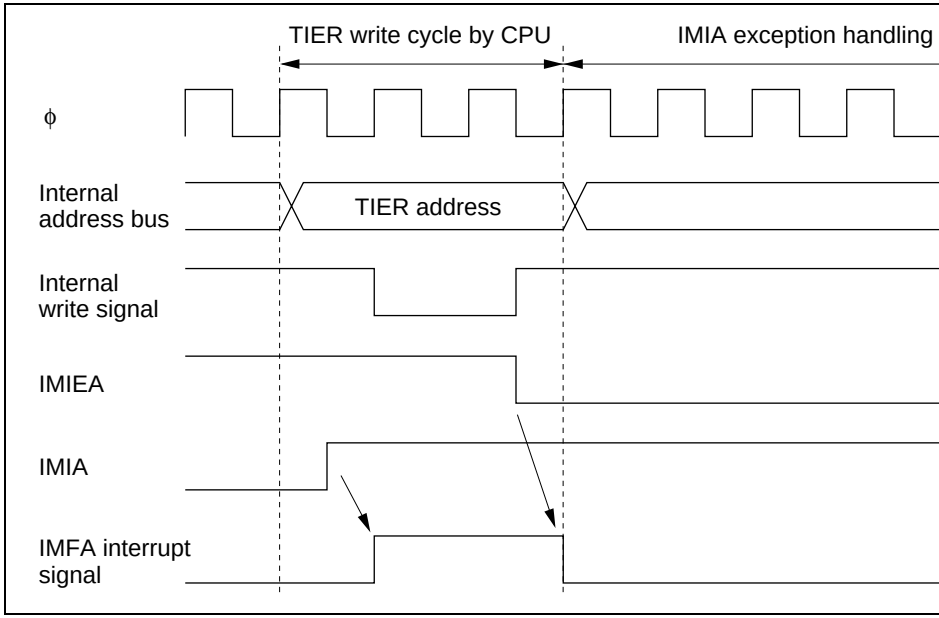


Figure 5.8 Contention between Interrupt and Interrupt-Disabling Instruc

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

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

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

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

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

5.5.4 Usage Notes on External Interrupts

The IRQnF flag specification calls for the flag to be cleared by writing 0 to it after it h
while set to 1. However, it is possible for the IRQnF flag to be cleared by mistake sim
writing 0 to it, irrespective of whether it has been read while set to 1, with the result th
exception handling is not executed. This occurs when the following conditions are ful

- Setting conditions
  1. Multiple external interrupts (IRQa, IRQb) are being used.
  2. Different clearing methods are being used: clearing by writing 0 for the IRQaF
     clearing by hardware for the IRQbF flag.
  3. A bit manipulation instruction is used on the IRQ status register to clear the IR
     else ISR is read as a byte unit, the IRQaF flag bit is cleared, and the values rea
     bits are written as a byte unit.

clears in error during ISR write for occurrence condition 2 and interrupt processing is n
out. However, if IRQbF flag reaches 0 between occurrence conditions 1 and 2, IRQbF
not clear in error.

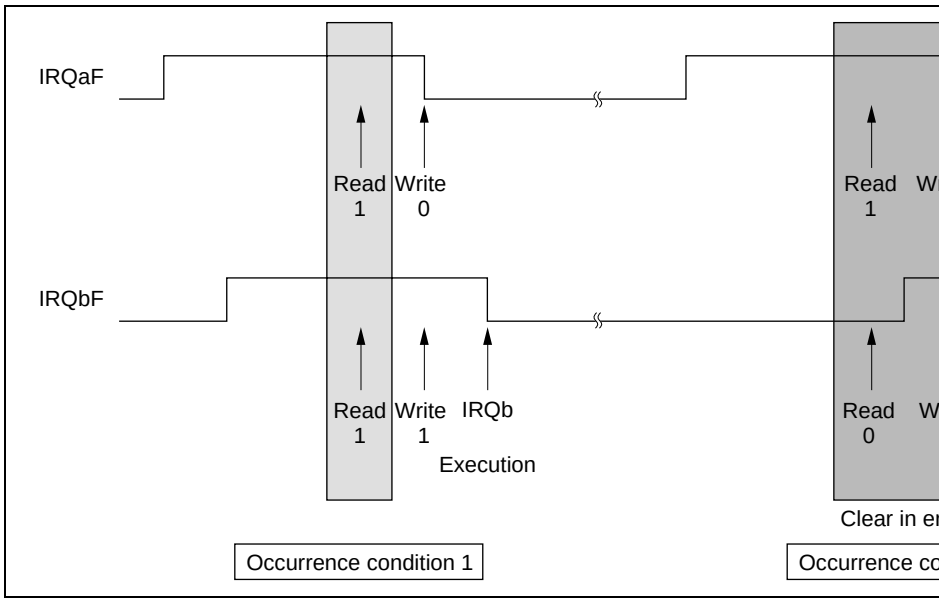


Figure 5.9 IRQnF Flag When Interrupt Processing Is Not Conducted

```
MOV.B R0L,@ISR
```

Countermeasure 2: During IRQb interrupt exception processing, carry out IRQbF fla
dummy processing.

For example, if b = 1

```
IRQB  MOV.B #HFD,R0L
      MOV.B R0L,@ISR
            ⋮
```

5.5.5 Notes on Non-Maskable Interrupts (NMI)

NMI is an exception processing that can be executed by the interrupt controller and C
chip internal circuits are operating normally under a specified electrical characteristics
is executed when the circuits are not operating normally due to some factors such as s
abnormal interrupt of input to the pins (runaway execution), the operation will not be

Incorrect NMI Operation Factors: Software

1. When an interrupt exception processing is executed in an H8/300H CPU, it is assu
   stack pointer (SP(ER7)) has already been set by software, and that the stack pointe
   points to the stack area set in a system such as RAM. If the program is in a runawa
   the stack pointer may be overflowed and updated illegally. Therefore, normal oper
   not be guaranteed.

2. Requests for NMIs can be accepted on the rising or falling edge of a pin. Acceptan
   rising or falling edge depends on the setting of the bit NMIEG in the system contro
   (SYSCR). It is necessary for the customer to set the bit according to the designated
   When the program is in a runaway execution, this bit may be rewritten illegally. T
   system may not operate as expected.

execution will resume after the reset exception processing has been executed.

Incorrect NMI Operation Factors: Abnormal Interrupts Input to the Chip Pins

If an abnormal interrupt which was not specified in the electrical characteristics is inpu
during a chip operation, the chip may be destroyed. In this case, the operation of the ch
be guaranteed.

When an abnormal interrupt has been input to a pin, the chip may not be destroyed; hov
internal circuits of the chip may partially or wholly malfunction, and the CPU may ente
unimagined undefined state when the CPU was designed. If this occurs, it will be impo
control the operation of the chip by external pins other than the external reset and stand
and the operation of the NMI will not be guaranteed. In this case, after some specified s
been input to the pins, input an external reset so that the chip can enter the normal prog
execution state again.

A bus arbitration function of the bus controller controls the operation of the DMA con
(DMAC) and refresh controller. The bus controller can also release the bus to an exter

6.1.1 Features

Features of the bus controller are listed below.

- Independent settings for address areas 7 to 0
  - 128-kbyte areas in 1-Mbyte modes; 2-Mbyte areas in 16-Mbyte modes.
  - Chip select signals ($\overline{CS}_7$ to $\overline{CS}_0$) can be output for areas 7 to 0.
  - Areas can be designated for 8-bit or 16-bit access.
  - Areas can be designated for two-state or three-state access.
- Four wait modes
  - Programmable wait mode, pin auto-wait mode, and pin wait modes 0 and 1 can
  - Zero to three wait states can be inserted automatically.
- Bus arbitration function
  - A built-in bus arbiter arbitrates the bus right to the CPU, DMAC, refresh contr
    external bus master.

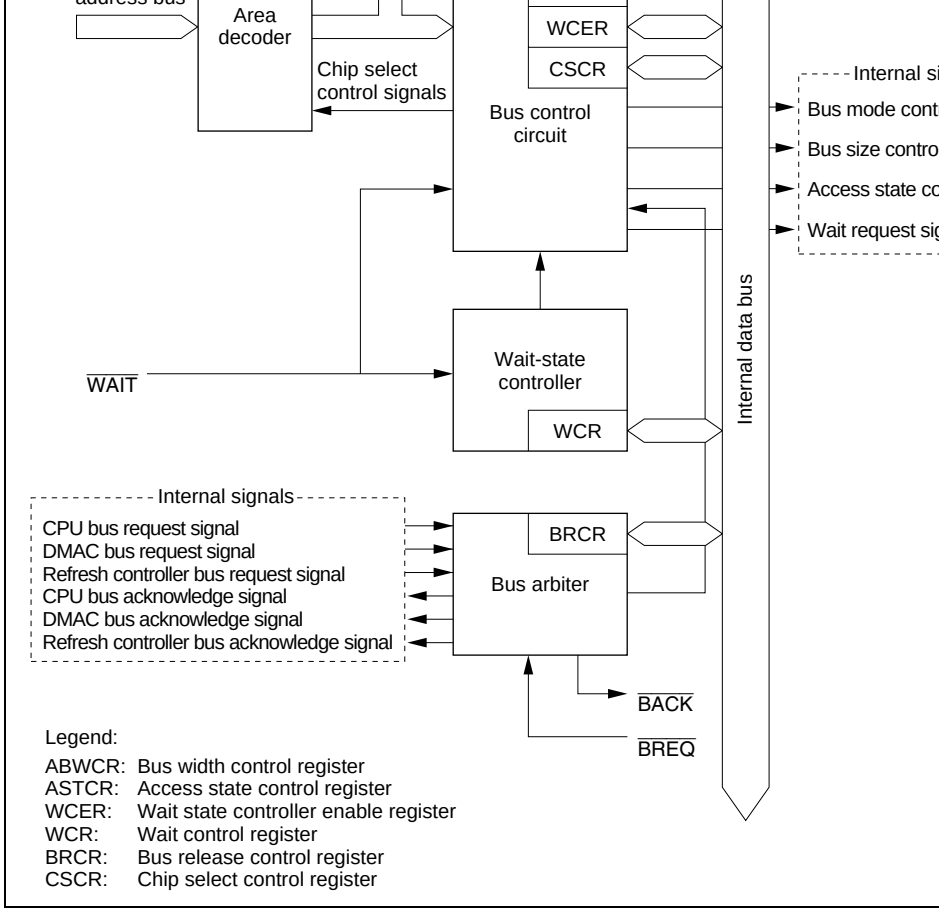


Figure 6.1 Block Diagram of Bus Controller

Address strobe	$\overline{AS}$	Output	Strobe signal indicating valid add on the address bus
Read	$\overline{RD}$	Output	Strobe signal indicating reading f external address space
High write	$\overline{HWR}$	Output	Strobe signal indicating writing to external address space, with val the upper data bus (D_{15} to D_8)
Low write	$\overline{LWR}$	Output	Strobe signal indicating writing to external address space, with val the lower data bus (D_7 to D_0)
Wait	$\overline{WAIT}$	Input	Wait request signal for access to three-state-access areas
Bus request	$\overline{BREQ}$	Input	Request signal for releasing the external device
Bus acknowledge	$\overline{BACK}$	Output	Acknowledge signal indicating th released to an external device

H'FFEC	Bus width control register	ABWCR	R/W	H'FF	H
H'FFED	Access state control register	ASTCR	R/W	H'FF	H
H'FFEE	Wait control register	WCR	R/W	H'F3	H
H'FFEF	Wait state controller enable register	WCER	R/W	H'FF	H
H'FFF3	Bus release control register	BRCR	R/W	H'FE	H
H'FF5F	Chip select control register	CSCR	R/W	H'0F	H

Note: * Lower 16 bits of the address.

6.2 Register Descriptions

6.2.1 Bus Width Control Register (ABWCR)

ABWCR is an 8-bit readable/writable register that selects 8-bit or 16-bit access for each

Bit		7	6	5	4	3	2	1
		ABW7	ABW6	ABW5	ABW4	ABW3	ABW2	ABW1
Initial value	Modes 1, 3, 5, 6	1	1	1	1	1	1	1
	Modes 2, 4, 7	0	0	0	0	0	0	0
Read/Write		R/W	R/W	R/W	R/W	R/W	R/W	R/W

Bits selecting bus width for each area

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 a 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, and 6 ABWCR is initialized to H'FF by a reset and in hardware st mode. In modes 2, 4, and 7 ABWCR is initialized to H'00 by a reset and in hardware st mode. ABWCR is not initialized in software standby mode.

ABWCR specifies the bus width of external memory areas. The bus width of on-chip internal I/O registers is fixed and does not depend on ABWCR settings. These settings therefore meaningless in single-chip mode (mode 7).

6.2.2 Access State Control Register (ASTCR)

ASTCR is an 8-bit readable/writable register that selects whether each area is accessed states or three states.

Bit	7	6	5	4	3	2	1
	AST7	AST6	AST5	AST4	AST3	AST2	AST1
Initial value	1	1	1	1	1	1	1
Read/Write	R/W	R/W	R/W	R/W	R/W	R/W	R/W

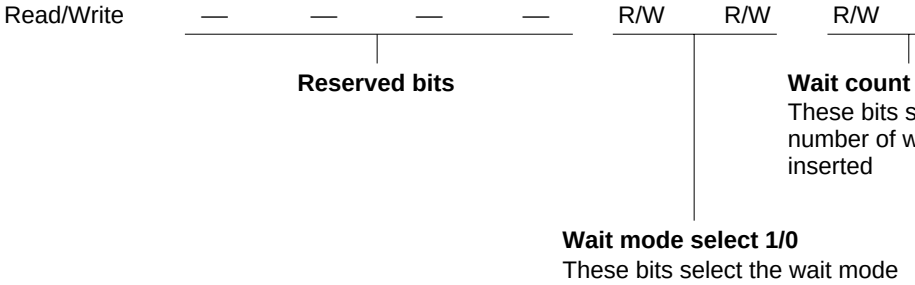
Bits selecting number of states for access to each area

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

Bits 7 to 0—Areas 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 r internal I/O registers are accessed in a fixed number of states that does not depend on settings. These settings are therefore meaningless in single-chip mode (mode 7).



WCR is initialized to H'F3 by a reset and in hardware standby mode. It is not initialized software standby mode.

Bits 7 to 4—Reserved: Read-only bits, always read as 1.

Bits 3 and 2—Wait Mode Select 1 and 0 (WMS1, WMS0): These bits select the wait

Bit 3: WMS1	Bit 2: WMS0	Description
0	0	Programmable wait mode (In
	1	No wait states inserted by wait-state controller
1	0	Pin wait mode 1
	1	Pin auto-wait mode

Bits 1 and 0—Wait Count 1 and 0 (WC1, WC0): These bits select the number of wai inserted in access to external three-state-access areas.

Bit 1: WC1	Bit 0: WC0	Description
0	0	No wait states inserted by wait-state controller
	1	1 state inserted
1	0	2 states inserted
	1	3 states inserted (In

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

Wait-state controller enable 7 to 0
These bits enable or disable wait-state control

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

Bits 7 to 0—Wait-State Controller Enable 7 to 0 (WCE7 to WCE0): These bits en
disable wait-state control of external three-state-access areas.

Bits 7 to 0: WCE7 to WCE0	Description
0	Wait-state control disabled (pin wait mode 0)
1	Wait-state control enabled (

Since WCER enables or disables wait-state control of external three-state-access areas
settings are meaningless in single-chip mode (mode 7).

Read/ Write	Modes 1, 2, 5, 7	—	—	—	—	—	—	—
	Modes 3, 4, 6	R/W	R/W	R/W	—	—	—	—

Address 23 to 21 enable
These bits enable PA_6 to PA_4 to be used for A_{23} to A_{21} address output

Reserved bits

Bus relea
Enables o
release o
an extern

BRCR is initialized to H'FE by a reset and in hardware standby mode. It is not initialize
software standby mode.

Bit 7—Address 23 Enable (A23E): Enables PA_4 to be used as the A_{23} address output p
0 in this bit enables A_{23} address output from PA_4. In modes other than 3, 4, and 6 this bi
modified and PA_4 has its ordinary input/output functions.

Bit 7: A23E	Description	
0	PA_4 is the A_{23} address output pin	
1	PA_4 is the PA_4/TP_4/TIOCA_1 input/output pin	(In

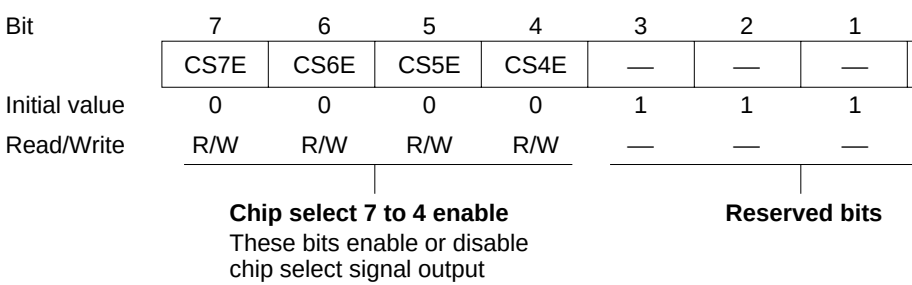
Bit 6—Address 22 Enable (A22E): Enables PA_5 to be used as the A_{22} address output p
0 in this bit enables A_{22} address output from PA_5. In modes other than 3, 4, and 6 this bi
modified and PA_5 has its ordinary input/output functions.

Bit 6: A22E	Description	
0	PA_5 is the A_{22} address output pin	
1	PA_5 is the PA_5/TP_5/TIOCB_1 input/output pin	(In

Bits 4 to 1—Reserved: Read-only bits, always read as 1.

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

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



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

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

Bit n: CSnE	Description	
0	Output of chip select signal CS_n is disabled	(In
1	Output of chip select signal CS_n is enabled	

Note: n = 7 to 4

Bits 3 to 0—Reserved: Read-only bits, always read as 1.

a. 1-Mbyte modes with on-chip ROM disabled (modes 1 and 2)

b. 16-Mbyte modes with on-chip ROM disabled (modes 3 and 4)

c. 1-Mbyte mode with on-chip ROM enabled (mode 5)

d. 16-M
on-ch
(mode

Notes: 1. The on-chip ROM, on-chip RAM, and internal I/O registers have a fixed bus width and are accessed in a fixed number of states.
2. When the RAME bit is cleared to 0 in SYSCR, this area conforms to the specifications of area 7.
3. This external address area conforms to the specifications of area 7.

Figure 6.2 Access Area Map for Modes 1 to 6

	1	0	—	—	16	3	Pin wait mode
		1	0	0	16	3	Programmable
				1	16	3	Disabled
			1	0	16	3	Pin wait mode
				1	16	3	Pin auto-wait m
1	0	—	—	—	8	2	Disabled
	1	0	—	—	8	3	Pin wait mode
		1	0	0	8	3	Programmable
				1	8	3	Disabled
			1	0	8	3	Pin wait mode
				1	8	3	Pin auto-wait m

Note: n = 0 to 7

In the expanded modes with on-chip ROM disabled, a reset leaves pin $\overline{CS}_0$ in the outp
pins $\overline{CS}_3$ to $\overline{CS}_1$ in the input state. To output chip select signals $\overline{CS}_3$ to $\overline{CS}_1$, the corresp
bits must be set to 1. In the expanded modes with on-chip ROM enabled, a reset leave
$\overline{CS}_0$ in the input state. To output chip select signals $\overline{CS}_3$ to $\overline{CS}_0$, the corresponding DDI
be set to 1. For details see section 9, I/O Ports.

Output of $\overline{CS}_7$ to $\overline{CS}_4$: Output of $\overline{CS}_7$ to $\overline{CS}_4$ is enabled or disabled in the chip select c
register (CSCR). A reset leaves pins $\overline{CS}_7$ to $\overline{CS}_4$ in the input state. To output chip selec
to $\overline{CS}_4$, the corresponding CSCR bits must be set to 1. For details see section 9, I/O Po

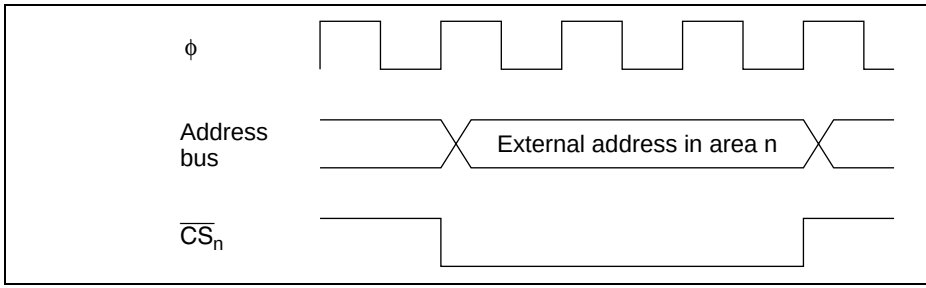


Figure 6.3 $\overline{CS}_n$ Output Timing (n = 7 to 0)

When the on-chip ROM, on-chip RAM, and internal I/O registers are accessed, $\overline{CS}_7$ ar
remain high. The $\overline{CS}_n$ signals are decoded from the address signals. They can be used
select signals for SRAM and other devices.

Table 6.4 indicates how the two parts of the data bus are used under different access co

Table 6.4 Access Conditions and Data Bus Usage

Area	Access Size	Read/ Write	Address	Valid Strobe	Upper Data Bus (D_{15} to D_8)	Lower D (D_7 to D_0
8-bit-access area	—	Read	—	$\overline{RD}$	Valid	Invalid
		Write	—	$\overline{HWR}$		Undeterr
16-bit-access area	Byte	Read	Even	$\overline{RD}$	Valid	Invalid
			Odd		Invalid	Valid
		Write	Even	$\overline{HWR}$	Valid	Undeterr
			Odd	$\overline{LWR}$	Undetermined data	Valid
	Word	Read	—	$\overline{RD}$	Valid	Valid
		Write	—	$\overline{HWR}$, $\overline{LWR}$	Valid	Valid

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

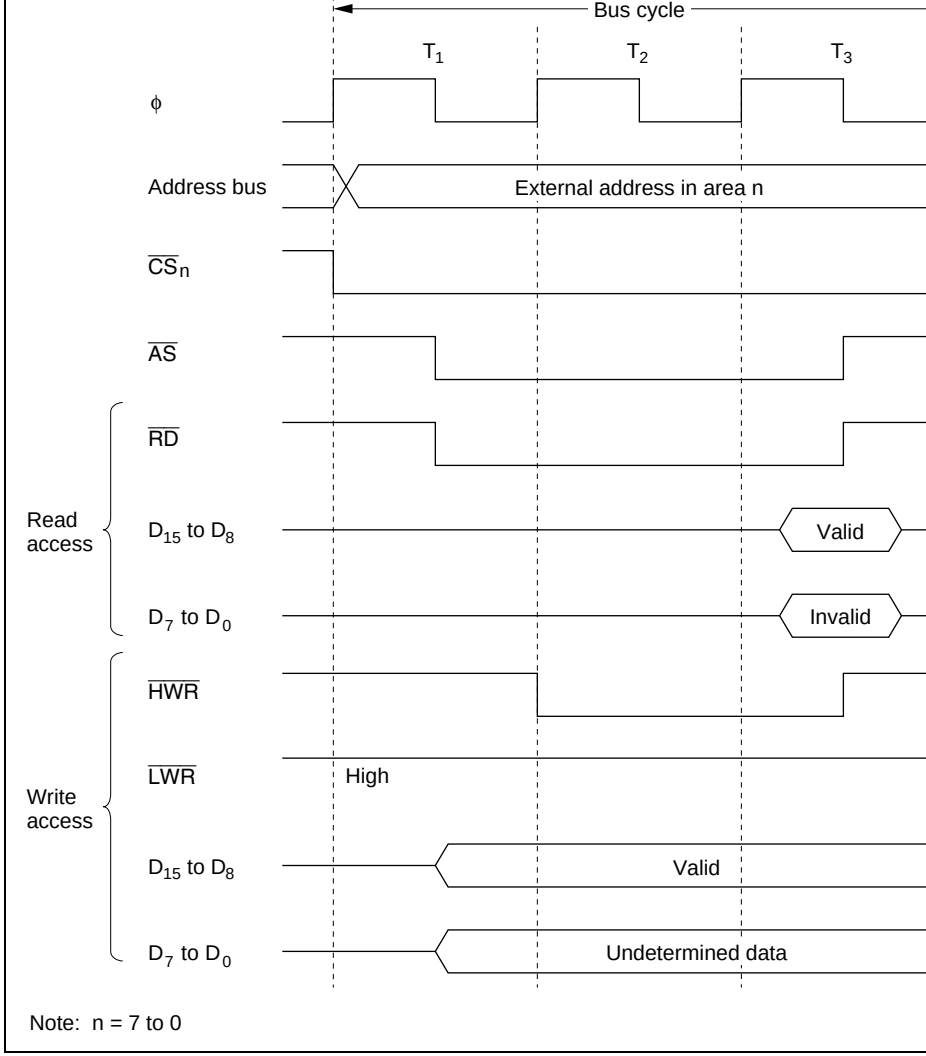


Figure 6.4 Bus Control Signal Timing for 8-Bit, Three-State-Access Ar

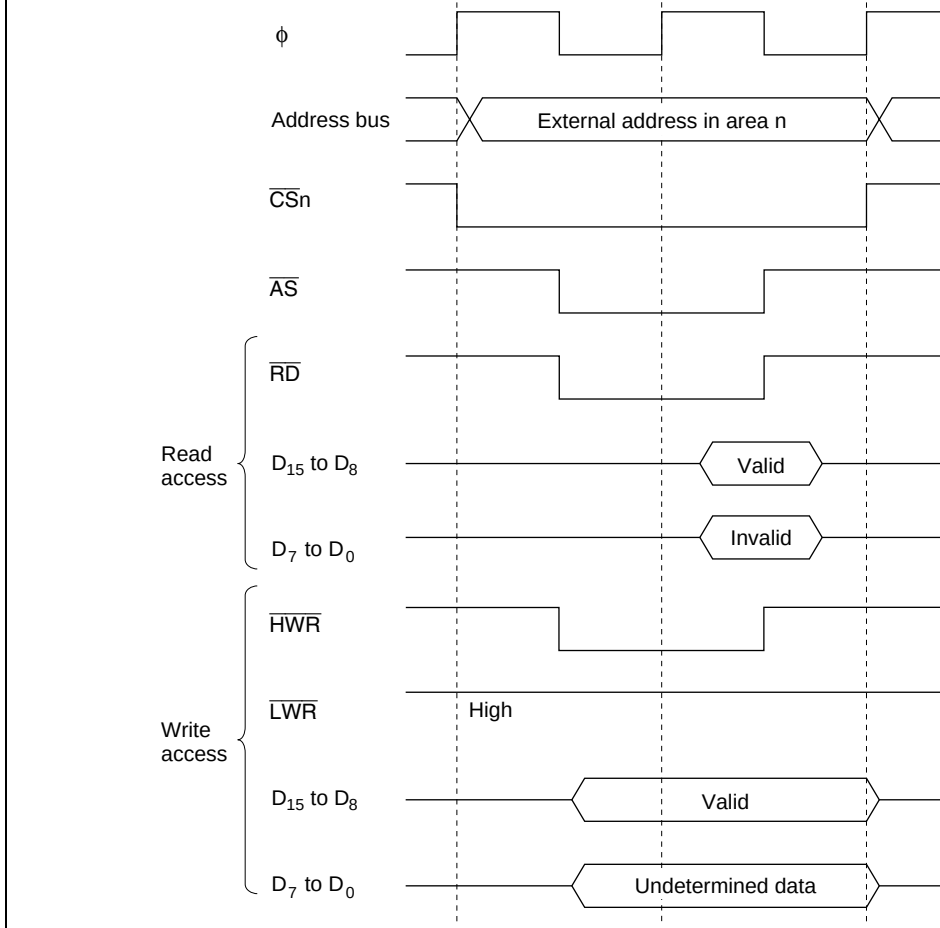


Figure 6.5 Bus Control Signal Timing for 8-Bit, Two-State-Access Area

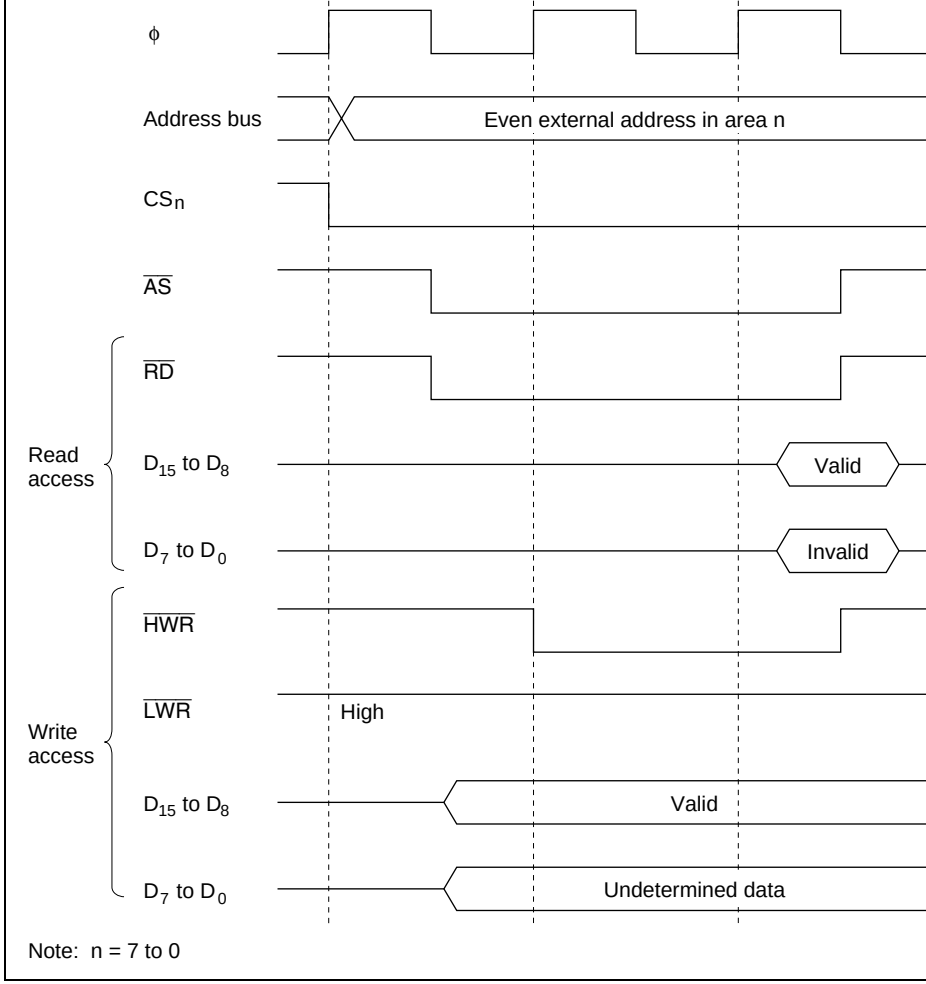


Figure 6.6 Bus Control Signal Timing for 16-Bit, Three-State-Access Are (Byte Access to Even Address)

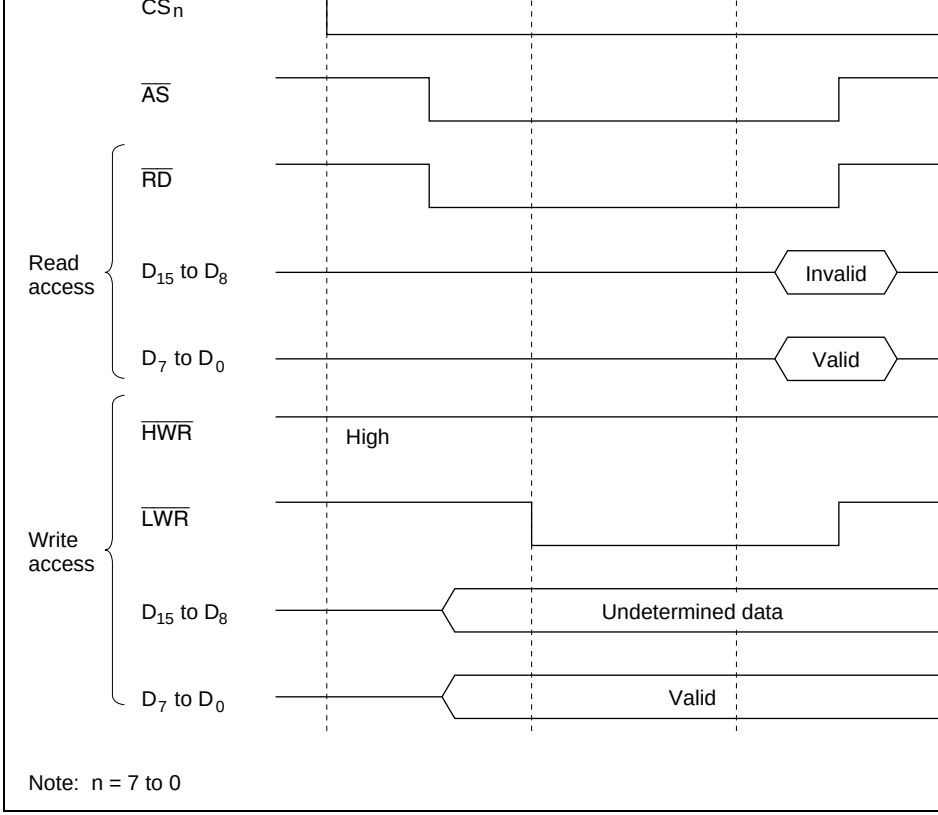


Figure 6.7 Bus Control Signal Timing for 16-Bit, Three-State-Access Area (Byte Access to Odd Address)

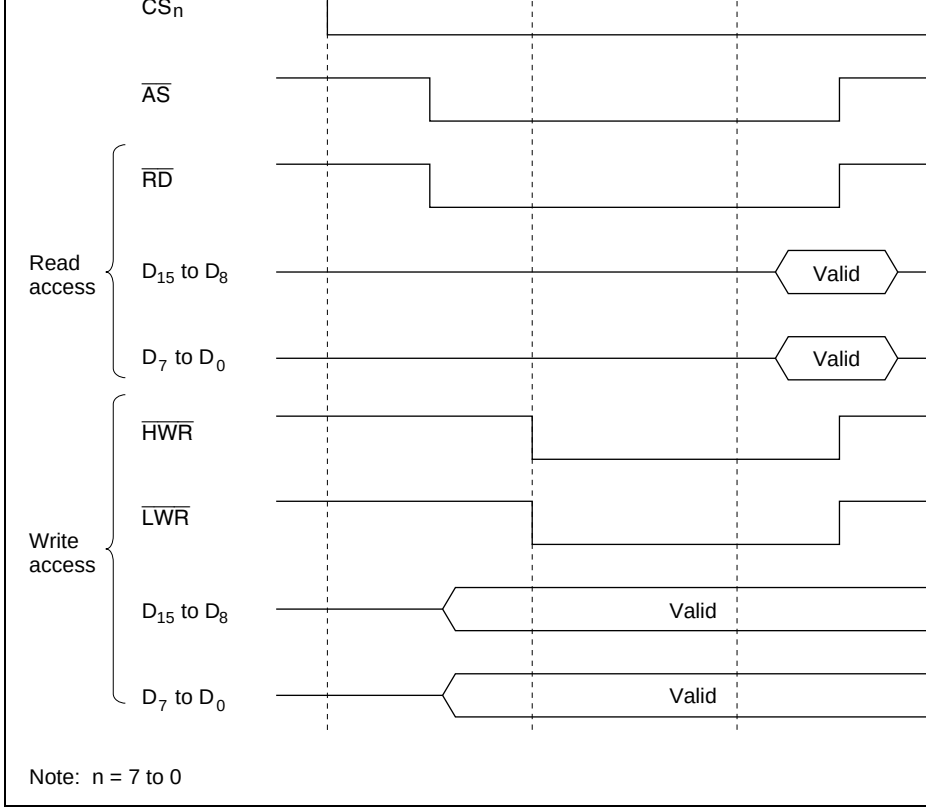


Figure 6.8 Bus Control Signal Timing for 16-Bit, Three-State-Access Are (Word Access)

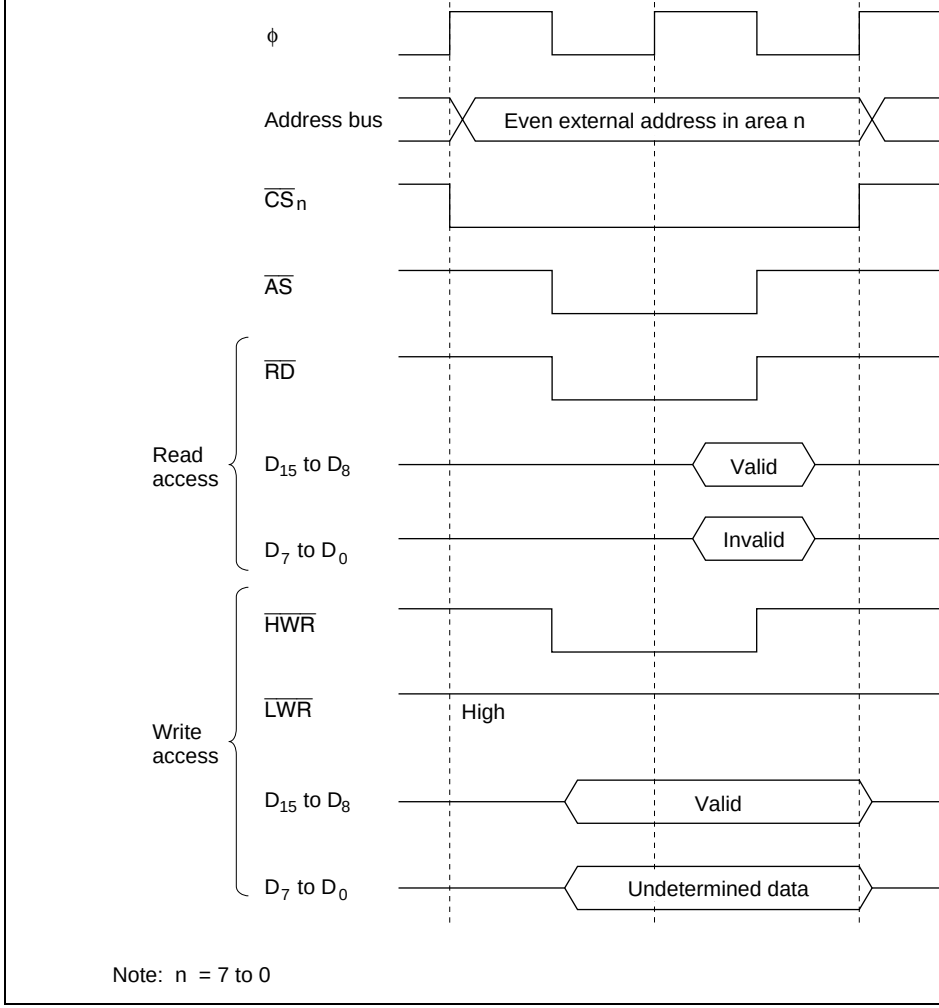


Figure 6.9 Bus Control Signal Timing for 16-Bit, Two-State-Access Area (Byte Access to Even Address)

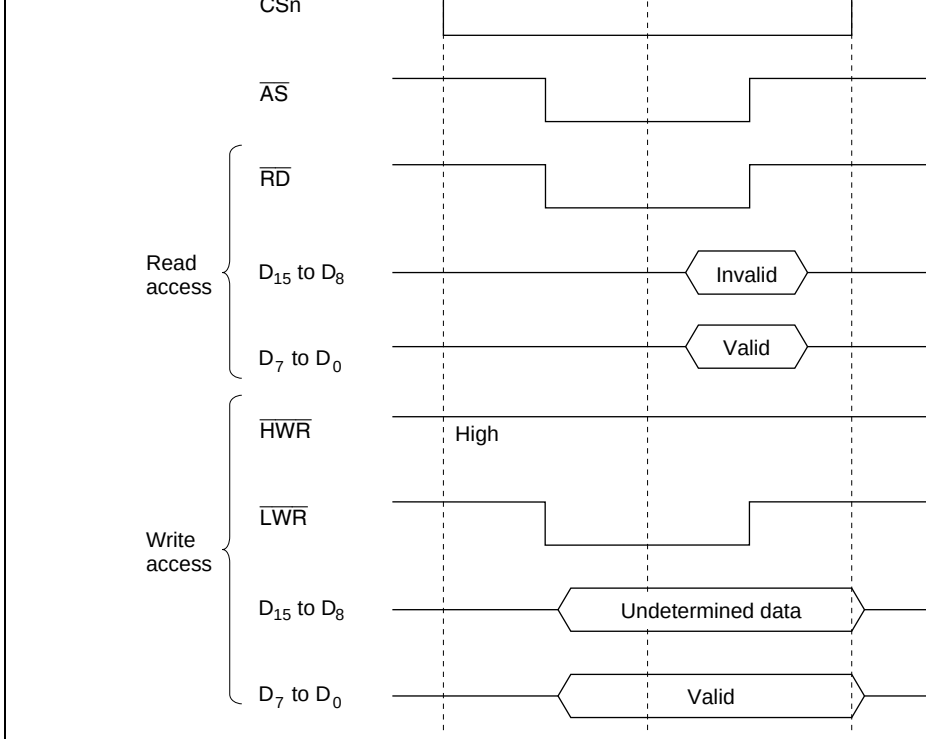


Figure 6.10 Bus Control Signal Timing for 16-Bit, Two-State-Access Are (Byte Access to Odd Address)

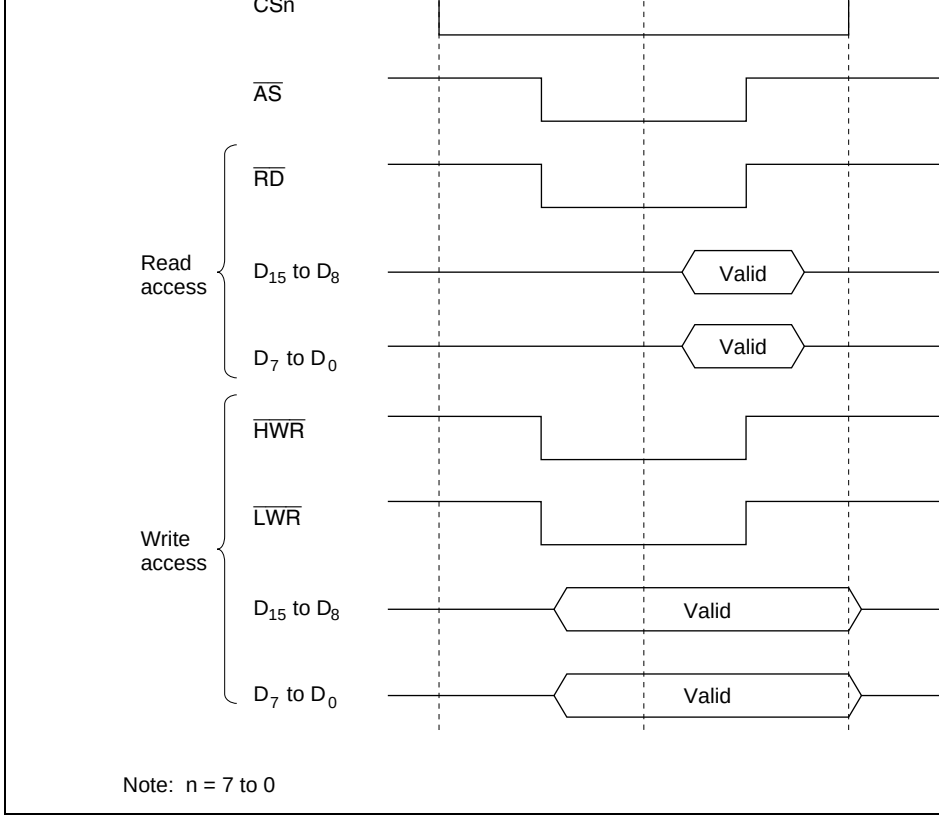


Figure 6.11 Bus Control Signal Timing for 16-Bit, Two-State-Access Area (Word Access)

1	0	—	—	Disabled	Pin wait mode 0
	1	0	0	Enabled	Programmable w
			1	Enabled	No wait states
		1	0	Enabled	Pin wait mode 1
			1	Enabled	Pin auto-wait mo

Note: n = 7 to 0

Wait Mode in Areas Where Wait-State Controller is Disabled

External three-state access areas in which the wait-state controller is disabled (ASTn = 0) operate in pin wait mode 0. The other wait modes are unavailable. The settings of b and WMS0 are ignored in these areas.

Pin Wait Mode 0: Wait states can only be inserted by $\overline{\text{WAIT}}$ pin control. During acce external three-state-access area, if the $\overline{\text{WAIT}}$ pin is low at the fall of the system clock state, a wait state (T_w) is inserted. If the $\overline{\text{WAIT}}$ pin remains low, wait states continue t until the $\overline{\text{WAIT}}$ signal goes high. Figure 6.12 shows the timing.

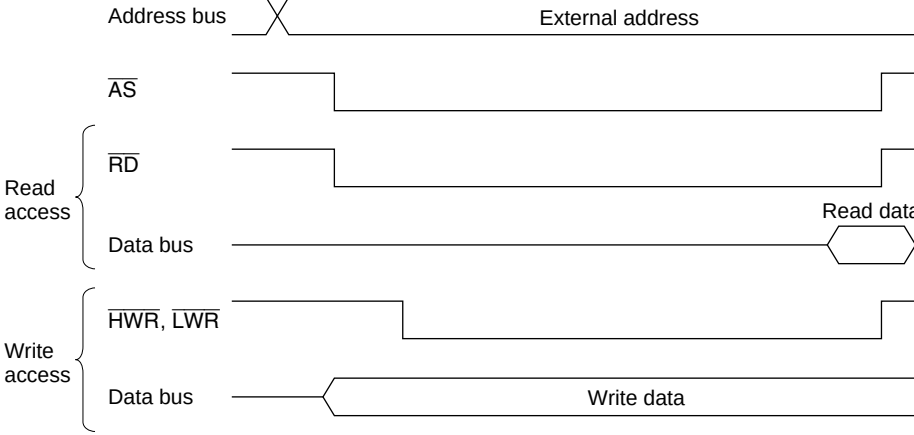


Figure 6.12 Pin Wait Mode 0

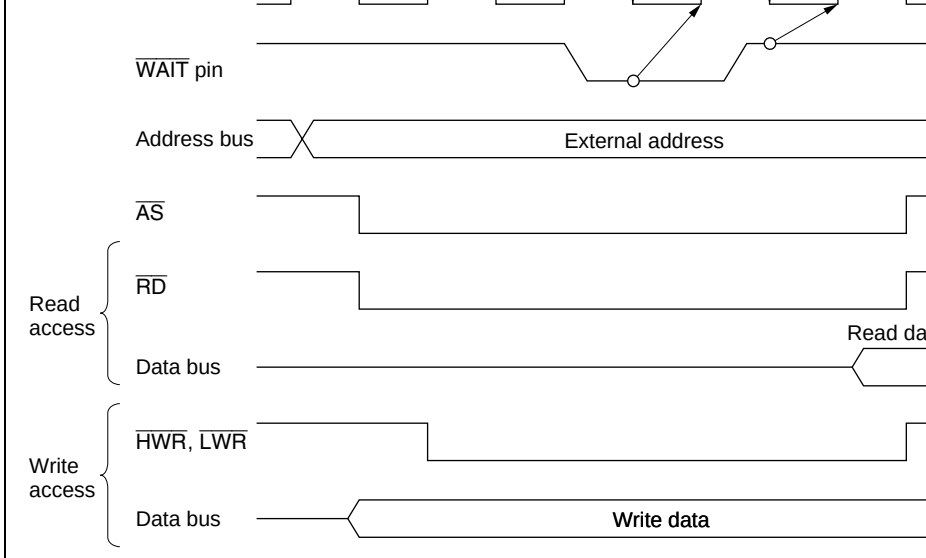
Wait Modes in Areas Where Wait-State Controller is Enabled

External three-state access areas in which the wait-state controller is enabled (ASTn = 1) can operate in pin wait mode 1, pin auto-wait mode, or programmable wait mode, as bits WMS1 and WMS0. Bits WMS1 and WMS0 apply to all areas, so all areas in which state controller is enabled operate in the same wait mode.

Pin Wait Mode 1: In all accesses to external three-state-access areas, the number of wa (T_w) selected by bits WC1 and WC0 are inserted. If the $\overline{\text{WAIT}}$ pin is low at the fall of t clock (ϕ) in the last of these wait states, an additional wait state is inserted. If the $\overline{\text{WAIT}}$ remains low, wait states continue to be inserted until the $\overline{\text{WAIT}}$ signal goes high.

Pin wait mode 1 is useful for inserting four or more wait states, or for inserting differen of wait states for different external devices.

If the wait count is 0, this mode operates in the same way as pin wait mode 0.



Note: * Arrows indicate time of sampling of the WAIT pin.

Figure 6.13 Pin Wait Mode 1

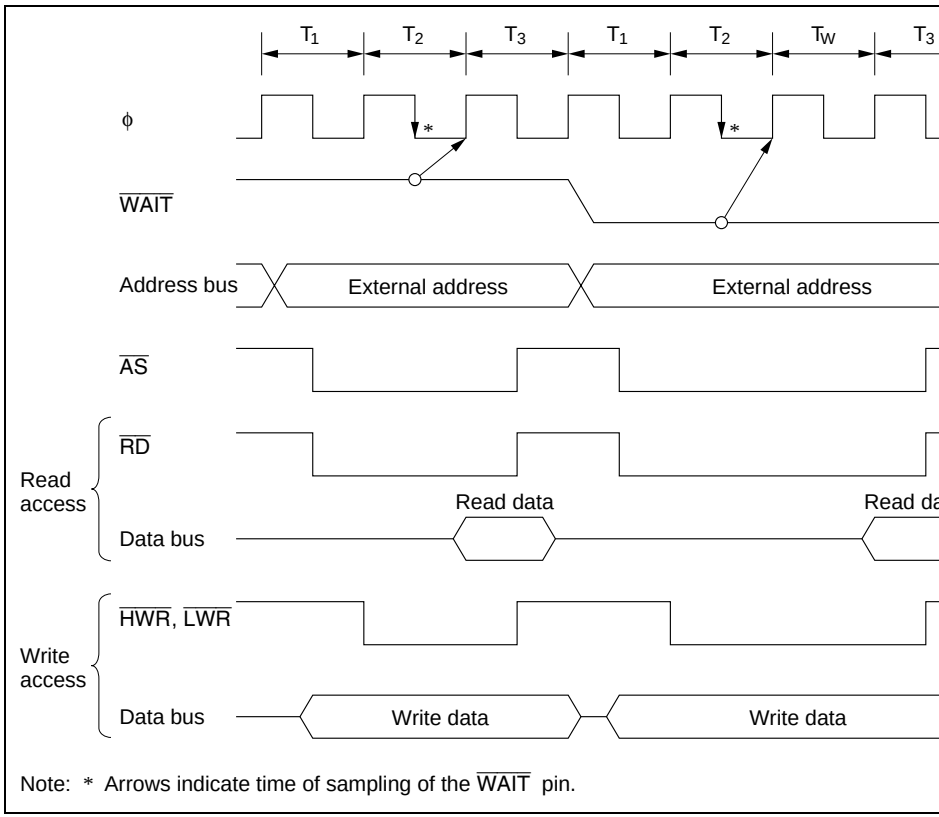


Figure 6.14 Pin Auto-Wait Mode

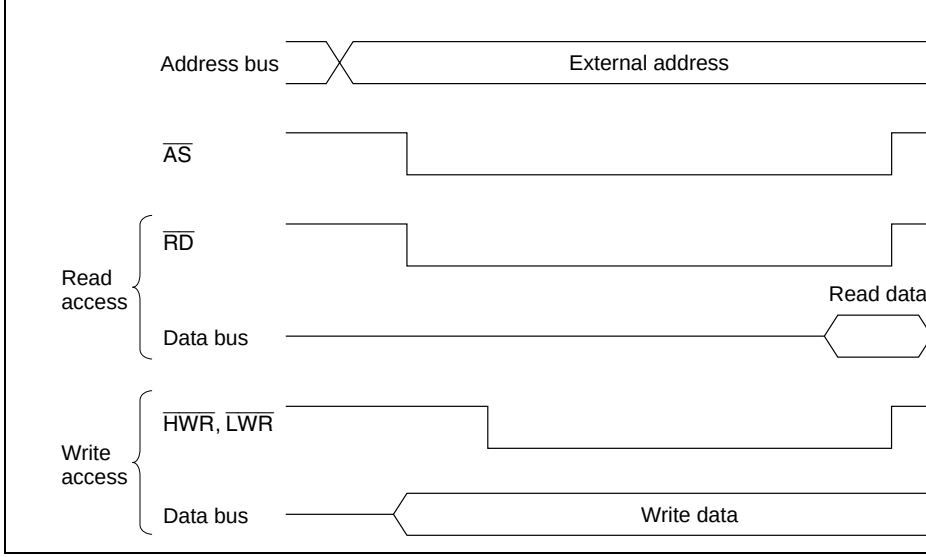
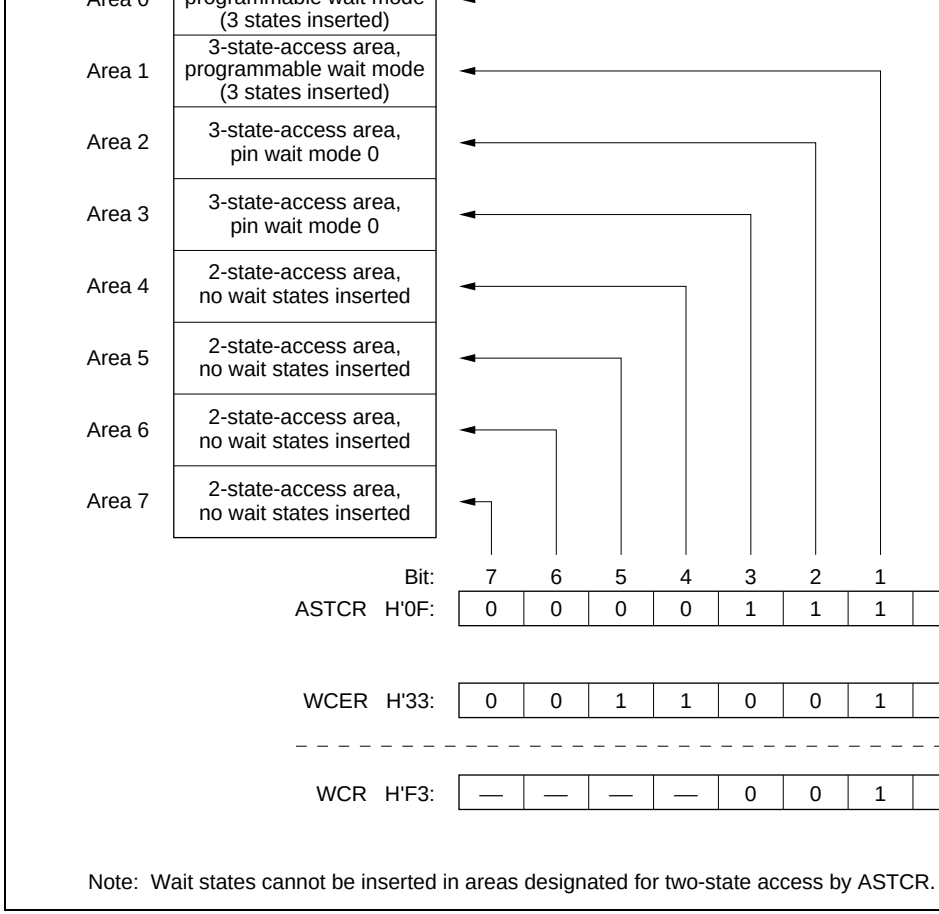


Figure 6.15 Programmable Wait Mode



Note: Wait states cannot be inserted in areas designated for two-state access by ASTCR.

Figure 6.16 Wait Mode Settings (Example)

A 256-kword × 16-bit EPROM is connected to area 0. This device is accessed in three 16-bit bus.

Two 32-kword × 8-bit SRAM devices (SRAM1 and SRAM2) are connected to area 1. devices are accessed in two states via a 16-bit bus.

One 32-kword × 8-bit SRAM (SRAM3) is connected to area 2. This device is accessed bus, using three-state access with an additional wait state inserted in pin auto-wait mo

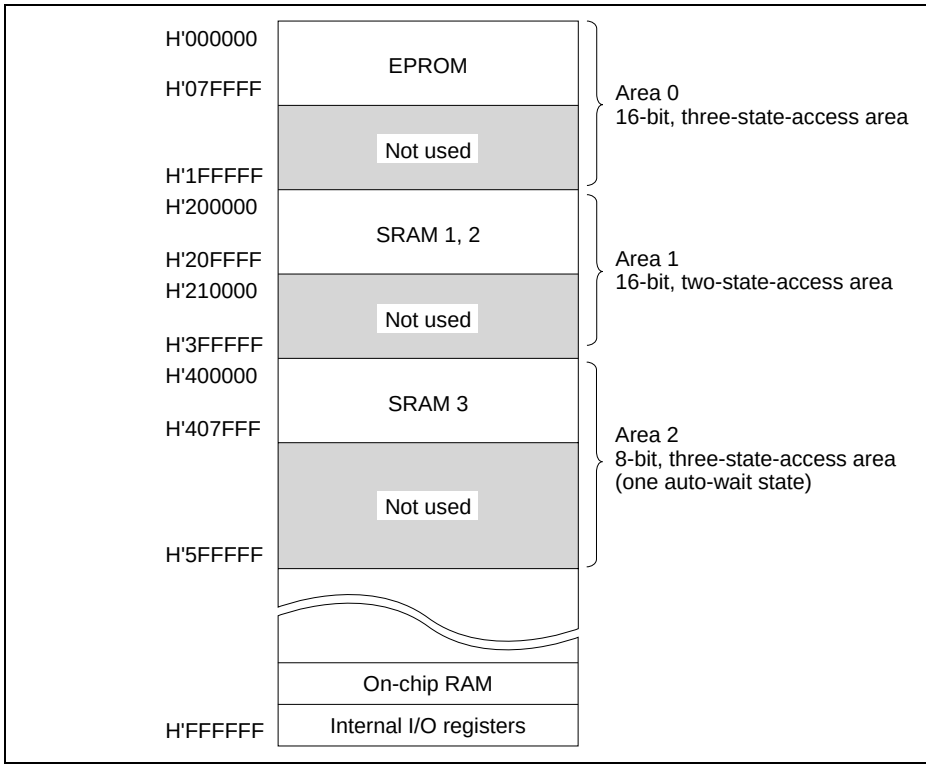


Figure 6.17 Memory Map (Example)

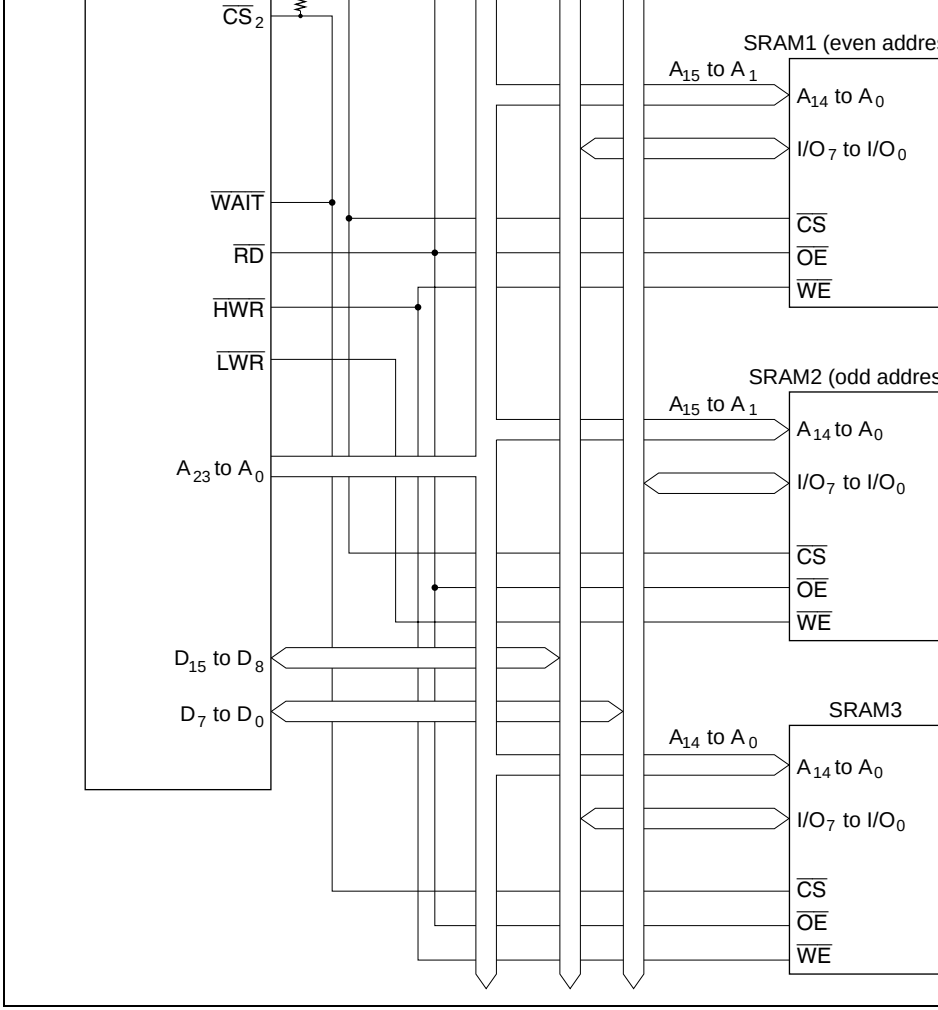


Figure 6.18 Interconnections with Memory (Example)

The bus arbiter checks whether the bus request signal from a bus master is active or in
returns an acknowledge signal to the bus master if the bus request signal is active. Wh
more bus masters request the bus, the highest-priority bus master receives an acknowl
The bus master that receives an acknowledge signal can continue to use the bus until t
acknowledge signal is deactivated.

The bus master priority order is:

(High) External bus master > refresh controller > DMAC > CPU (Lo

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

CPU

The CPU is the lowest-priority bus master. If the DMAC, refresh controller, or an exte
master requests the bus while the CPU has the bus right, the bus arbiter transfers the b
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 tw
accesses.
- If another bus master requests the bus while the CPU is performing internal operat
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 righ
transferred immediately.

the read cycle and the write cycle.

There is a priority order among the DMAC channels. For details see section 8.4.9, DMA Multiple-Channel Operation.

Refresh Controller

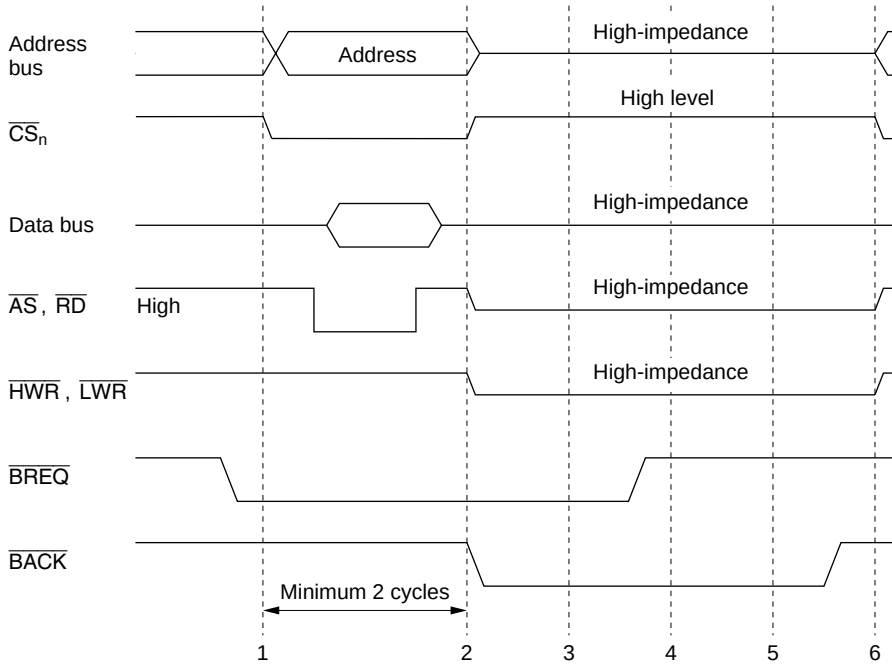
When a refresh cycle is requested, the refresh controller requests the bus right from the When the refresh cycle is completed, the refresh controller releases the bus. For details 7, Refresh Controller.

External Bus Master

When the BRLE bit is set to 1 in BRCR, the bus can be released to an external bus mas external bus master has highest priority, and requests the bus right from the bus arbiter the $\overline{\text{BREQ}}$ signal low. Once the external bus master gets the bus, it keeps the bus right u $\overline{\text{BREQ}}$ signal goes high. While the bus is released to an external bus master, the H8/304 holds the address bus and data bus control signals ($\overline{\text{AS}}$, $\overline{\text{RD}}$, $\overline{\text{HWR}}$, and $\overline{\text{LWR}}$) in the hi impedance state, holds the chip select signals high ($\overline{\text{CS}}_n$: n = 7 to 0), and holds the $\overline{\text{BAC}}$ 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 lo is released to the external bus master at the appropriate opportunity. The $\overline{\text{BREQ}}$ signal held low until the $\overline{\text{BACK}}$ signal goes low.

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



1 Low $\overline{BREQ}$ signal is sampled at rise of T_1 state.
2 $\overline{BACK}$ signal goes low at end of CPU read cycle, releasing bus right to external bus mast
3 $\overline{BREQ}$ pin continues to be sampled while bus is released to external bus master.
4, 5 High $\overline{BREQ}$ signal is sampled twice consecutively.
6 $\overline{BACK}$ signal goes high, ending bus-release cycle.

Note: n = 7 to 0

Figure 6.19 External-Bus-Released State (Two-State-Access Area, During Re

ABWCR, ASTCR, and WCER Write Timing

Data written to ABWCR, ASTCR, or WCER takes effect starting from the next bus cyc
6.20 shows the timing when an instruction fetched from area 0 changes area 0 from thr
access to two-state access.

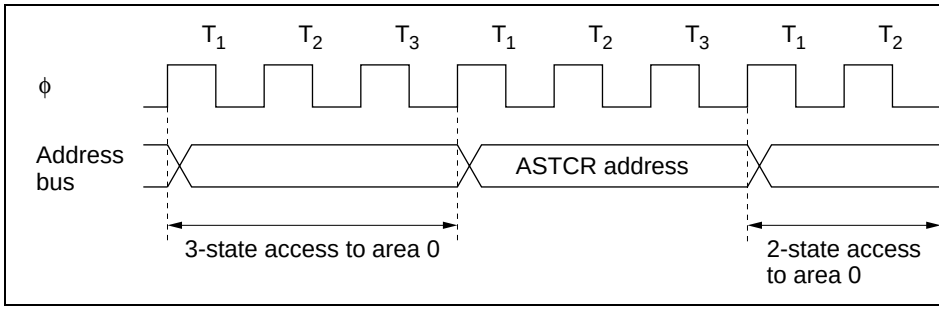


Figure 6.20 ASTCR Write Timing

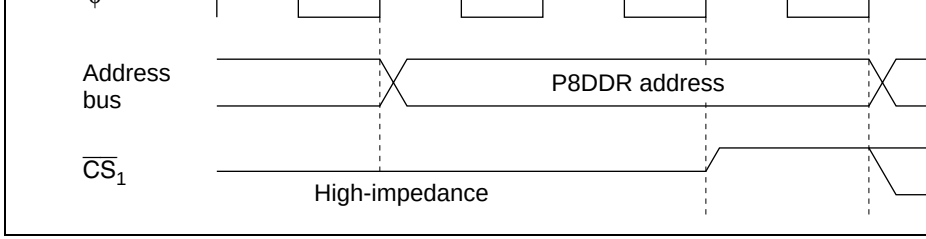


Figure 6.21 DDR Write Timing

BRCR Write Timing

Data written to switch between A_{23}, A_{22}, or A_{21} output and generic input or output takes starting from the T_3 state of the BRCR write cycle. Figure 6.22 shows the timing when changed from generic input to A_{23}, A_{22}, or A_{21} output.

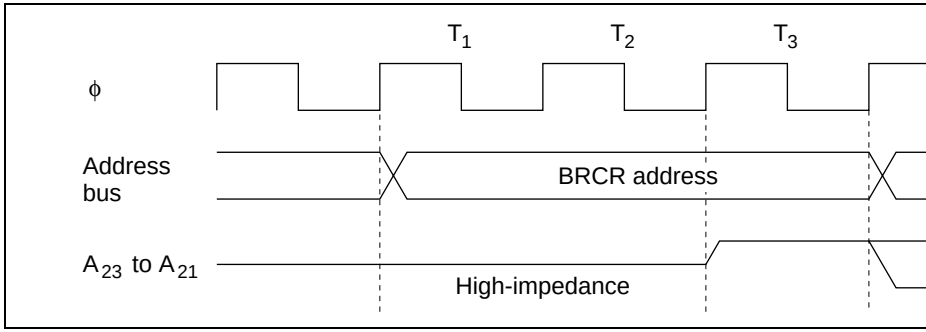


Figure 6.22 BRCR Write Timing

6.4.4 Transition To Software Standby Mode

If contention occurs between a transition to software standby mode and a bus request fr
external bus master, the bus may be released for one state just before the transition to so
standby mode (see figure 6.23). When using software standby mode, clear the BRLE bi
BRCR before executing the SLEEP instruction.

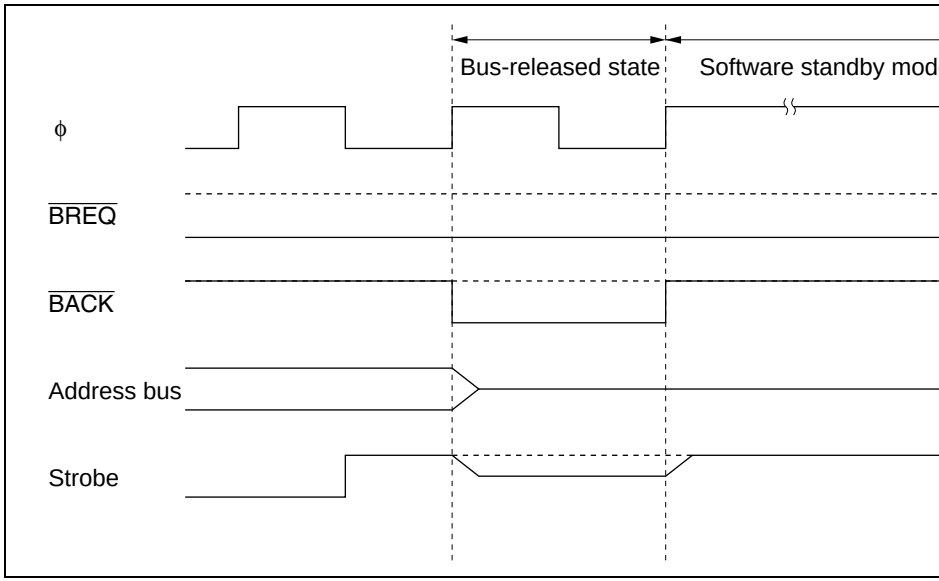


Figure 6.23 Contention between Bus-Released State and Software Standby

DRAM or pseudo-static RAM can be directly connected to area 3 of the external addr

A maximum 128 kbytes can be connected in modes 1, 2, and 5 (1-Mbyte modes). A m
2 Mbytes can be connected in modes 3, 4, and 6 (16-Mbyte modes).

Systems that do not need to refresh DRAM or pseudo-static RAM can use the refresh
an 8-bit interval timer.

When the refresh controller is not used, it can be independently halted to conserve pow
details see section 20.6, Module Standby Function.

7.1.1 Features

The refresh controller can be used for one of three functions: DRAM refresh control, p
RAM refresh control, or 8-bit interval timing. Features of the refresh controller are lis

Features as a DRAM Refresh Controller:

- Enables direct connection of 16-bit-wide DRAM
- Selection of 2$\overline{\text{CAS}}$ or 2$\overline{\text{WE}}$ mode
- Selection of 8-bit or 9-bit column address multiplexing for DRAM address input
  Examples:
  — 1-Mbit DRAM: 8-bit row address × 8-bit column address
  — 4-Mbit DRAM: 9-bit row address × 9-bit column address
  — 4-Mbit DRAM: 10-bit row address × 8-bit column address
- $\overline{\text{CAS}}$-before-$\overline{\text{RAS}}$ refresh control
- Software-selectable refresh interval
- Software-selectable self-refresh mode
- Wait states can be inserted

- Refresh timer counter (RTCNT) can be used as an 8-bit up-counter
- Selection of seven counter clock sources: ϕ/2, ϕ/8, ϕ/32, ϕ/128, ϕ/512, ϕ/2048, ϕ/40
- Interrupts can be generated by compare match between RTCNT and the refresh tim register (RTCOR)

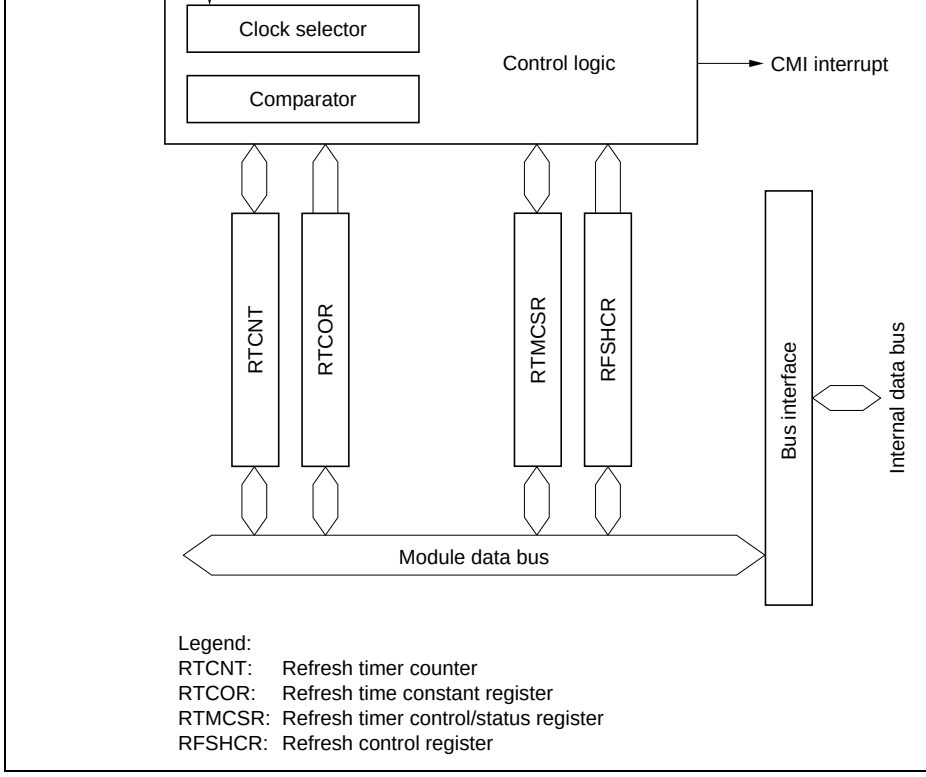


Figure 7.1 Block Diagram of Refresh Controller

				used to refresh DRAM an
$\overline{HWR}$	Upper write/upper column address strobe	UW/UCAS	Output	Connects to the $\overline{UW}$ pin o DRAM or $\overline{UCAS}$ pin of 2C
$\overline{LWR}$	Lower write/lower column address strobe	LW/LCAS	Output	Connects to the $\overline{LW}$ pin o DRAM or $\overline{LCAS}$ pin of 2C
$\overline{RD}$	Column address strobe/ write enable	CAS/WE	Output	Connects to the $\overline{CAS}$ pin DRAM or $\overline{WE}$ pin of 2CA
$\overline{CS_3}$	Row address strobe	RAS	Output	Connects to the $\overline{RAS}$ pin

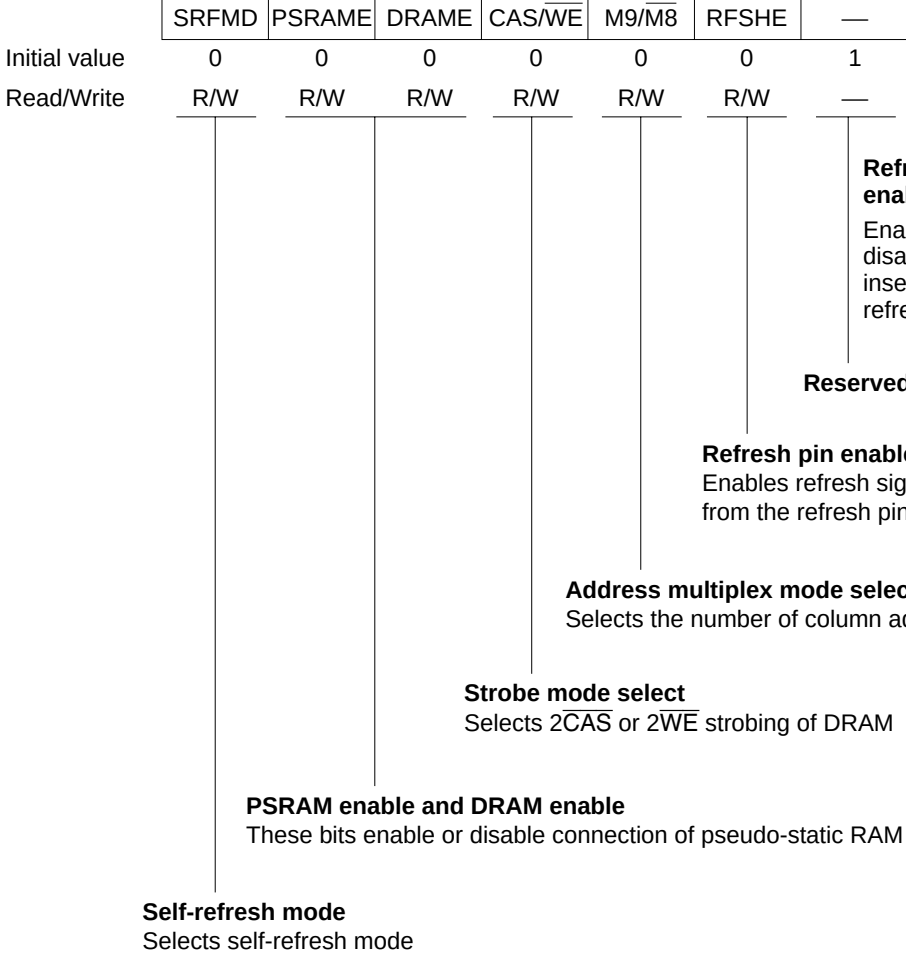
7.1.4 Register Configuration

Table 7.2 summarizes the refresh controller's registers.

Table 7.2 Refresh Controller Registers

Address*	Name	Abbreviation	R/W	Ini
H'FFAC	Refresh control register	RFSHCR	R/W	H'0
H'FFAD	Refresh timer control/status register	RTMCSR	R/W	H'0
H'FFAE	Refresh timer counter	RTCNT	R/W	H'0
H'FFAF	Refresh time constant register	RTCOR	R/W	H'F

Note: * Lower 16 bits of the address.



RFSHCR is initialized to H'02 by a reset and in hardware standby mode.

0	DRAM or PSRAM self-refresh is disabled in software standby mod (In
1	DRAM or PSRAM self-refresh is enabled in software standby mod

Bit 6—PSRAM Enable (PSRAME) and
Bit 5—DRAM Enable (DRAME): These bits enable or disable connection of pseudo- and DRAM to area 3 of the external address space.

When DRAM or pseudo-static RAM is connected, the bus cycle and refresh cycle of ar of three states, regardless of the setting in the access state control register (ASTCR). If in ASTCR, wait states cannot be inserted.

When the PSRAME or DRAME bit is set to 1, bits 0, 2, 3, and 4 in RFSHCR and regis RTMCSR, RTCNT, and RTCOR are write-disabled, except that the CMF flag in RTM cleared by writing 0.

Bit 6: PSRAME	Bit 5: DRAME	Description
0	0	Can be used as an interval timer (In
		(DRAM and PSRAM cannot be directly conne
	1	DRAM can be directly connected
1	0	PSRAM can be directly connected
	1	Illegal setting

Bit 4—Strobe Mode Select (CAS/$\overline{\text{WE}}$): Selects 2$\overline{\text{CAS}}$ or 2$\overline{\text{WE}}$ mode. The setting of th valid when PSRAME = 0 and DRAME = 1. This bit is write-disabled when the PSRAM DRAME bit is set to 1.

Bit 4: CAS/$\overline{\text{WE}}$	Description
0	2$\overline{\text{WE}}$ mode (In
1	2$\overline{\text{CAS}}$ mode

Bit 2—Refresh Pin Enable (RFSHE): Enables or disables refresh signal output from pin. This bit is write-disabled when the PSRAME or DRAME bit is set to 1.

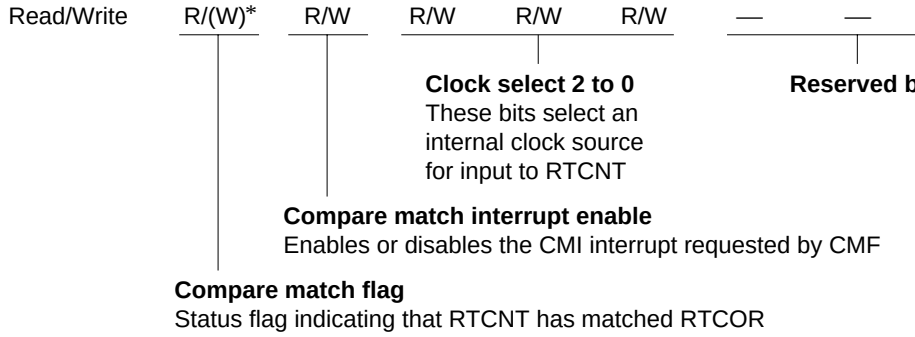
Bit 2: RFSHE	Description
0	Refresh signal output at the $\overline{\text{RFSH}}$ pin is disabled (the $\overline{\text{RFSH}}$ pin as a generic input/output port) (
1	Refresh signal output at the $\overline{\text{RFSH}}$ pin is enabled

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

Bit 0—Refresh Cycle Enable (RCYCE): Enables or disables insertion of refresh cyc

The setting of this bit is valid when PSRAME = 1 or DRAME = 1. When PSRAME = DRAME = 0, refresh cycles are not inserted regardless of the setting of this bit.

Bit 0: RCYCE	Description
0	Refresh cycles are disabled (
1	Refresh cycles are enabled for area 3



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

Bits 7 and 6 are initialized by a reset and in standby mode. Bits 5 to 3 are initialized by in hardware standby mode, but retain their previous values on transition to software sta

Bit 7—Compare Match Flag (CMF): This status flag indicates that the RTCNT and R values have matched.

Bit 7: CMF	Description
0	[Clearing condition]
	Cleared by reading CMF when CMF = 1, then writing 0 in CMF
1	[Setting condition]
	When RTCNT = RTCOR

Bits 5 to 3—Clock Select 2 to 0 (CKS2 to CKS0): These bits select an internal clock
input to RTCNT. When used for refresh control, the refresh controller outputs a refres
periodic intervals determined by compare match between RTCNT and RTCOR. When
interval timer, the refresh controller generates CMI interrupts at periodic intervals det
compare match. These bits are write-disabled when the PSRAME bit or DRAME bit i

Bit 5: CKS2	Bit 4: CKS1	Bit 3: CKS0	Description
0	0	0	Clock input is disabled (
		1	φ/2 clock source
	1	0	φ/8 clock source
		1	φ/32 clock source
1	0	0	φ/128 clock source
		1	φ/512 clock source
	1	0	φ/2048 clock source
		1	φ/4096 clock source

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

RTCNT is an up-counter that is incremented by an internal clock selected by bits CKS2 in RTMCSR. When RTCNT matches RTCOR (compare match), the CMF flag is set to RTCNT is cleared to H'00.

RTCNT is write-disabled when the PSRAME bit or DRAME bit is set to 1. RTCNT is to H'00 by a reset and in standby mode.

7.2.4 Refresh Time Constant Register (RTCOR)

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

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

RTCOR is write-disabled when the PSRAME bit or DRAME bit is set to 1. RTCOR is to H'FF by a reset and in hardware standby mode. In software standby mode it retains it value.

Table 7.3 Refresh Controller Settings

Register Settings		Usage		
		DRAM Interface	**PSRAM Interface**	**Interval**
RFSHCR	SRFMD	Selects self-refresh mode	Selects self-refresh mode	Cleared
	PSRAME	Cleared to 0	Set to 1	Cleared
	DRAME	Set to 1	Cleared to 0	Cleared
	CAS/$\overline{WE}$	Selects 2$\overline{CAS}$ or 2$\overline{WE}$ mode	—	—
	M9/$\overline{M8}$	Selects column addressing mode	—	—
	RFSHE	Selects $\overline{RFSH}$ signal output	Selects $\overline{RFSH}$ signal output	Cleared
	RCYCE	Selects insertion of refresh cycles	Selects insertion of refresh cycles	—
RTCOR		Refresh interval setting	Refresh interval setting	Interrupt setting
RTMCSR	CKS2 to CKS0			
	CMF	Set to 1 when RTCNT = RTCOR	Set to 1 when RTCNT = RTCOR	Set to 1 RTCNT
	CMIE	Cleared to 0	Cleared to 0	Enables interrupt
P8DDR	$P8_1$DDR	Set to 1 ($\overline{CS_3}$ output)	Set to 1 ($\overline{CS_3}$ output)	Set to 0
ABWCR	ABW3	Cleared to 0	—	—

To set up area 3 for connection to pseudo-static RAM, initialize RTCOR, RTMCSR, ar
RFSHCR in that order, setting bit PSRAME to 1 and clearing bit DRAME to 0. Set bit
1 in P8DDR to enable $\overline{CS}_3$ output.

Interval Timer

When PSRAME = 0 and DRAME = 0, the refresh controller operates as an interval tim
setting RTCOR, select an input clock in RTMCSR and set the CMIE bit to 1. CMI inte
be requested at compare match intervals determined by RTCOR and bits CKS2 to CKS
RTMCSR.

When setting RTCOR, RTMCSR, and RFSHCR, make sure that PSRAME = 0 and DR
Writing is disabled when either of these bits is set to 1.

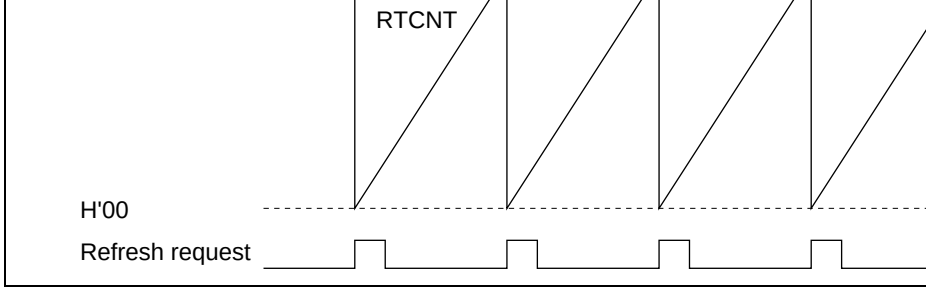


Figure 7.2 Refresh Request Interval (RCYCE = 1)

Refresh requests are generated at regular intervals as shown in figure 7.2, but the refre
not actually executed until the refresh controller gets the bus right.

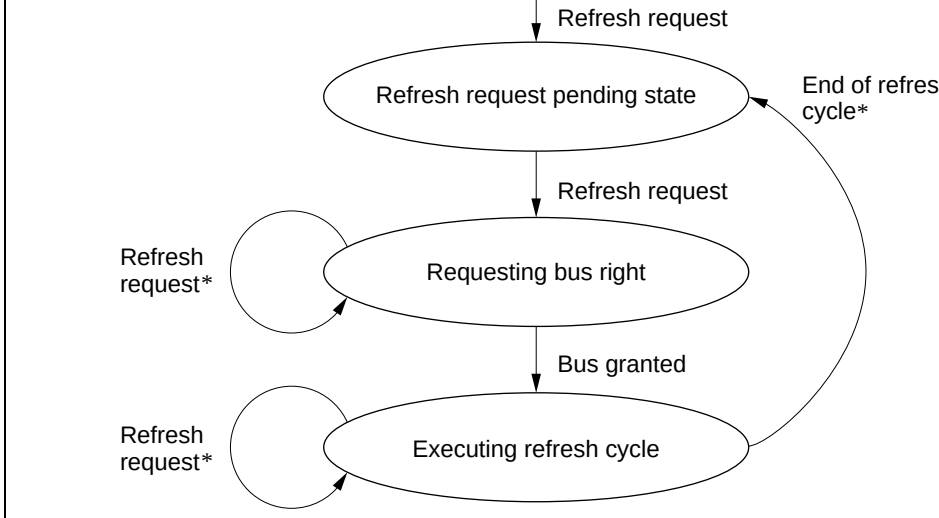
Table 7.4 summarizes the relationship among area 3 settings, DRAM read/write cycle
refresh cycles.

Table 7.4 Area 3 Settings, DRAM Access Cycles, and Refresh Cycles

Area 3 Settings	Read/Write Cycle by CPU or DMAC	Refresh Cycle
2-state-access area (AST3 = 0)	• 3 states • Wait states cannot be inserted	• 3 states • Wait states cannot
3-state-access area (AST3 = 1)	• 3 states • Wait states can be inserted	• 3 states • Wait states can be

To insert refresh cycles, set the RCYCE bit to 1 in RFSHCR. Figure 7.3 shows the sta
for execution of refresh cycles.

When the first refresh request occurs after exit from the reset state or standby mode, th
controller does not execute a refresh cycle, but goes into the refresh request pending s
this point when using a DRAM that requires a refresh cycle for initialization.



Note: * A refresh request is ignored if it occurs while the refresh controller is reques
bus right or executing a refresh cycle.

Figure 7.3 State Transitions for Refresh Cycle Execution

Address Multiplexing

Address multiplexing depends on the setting of the M9/$\overline{\text{M8}}$ bit in RFSHCR, as describe
7.5. Figure 7.4 shows the address output timing. Address output is multiplexed only in

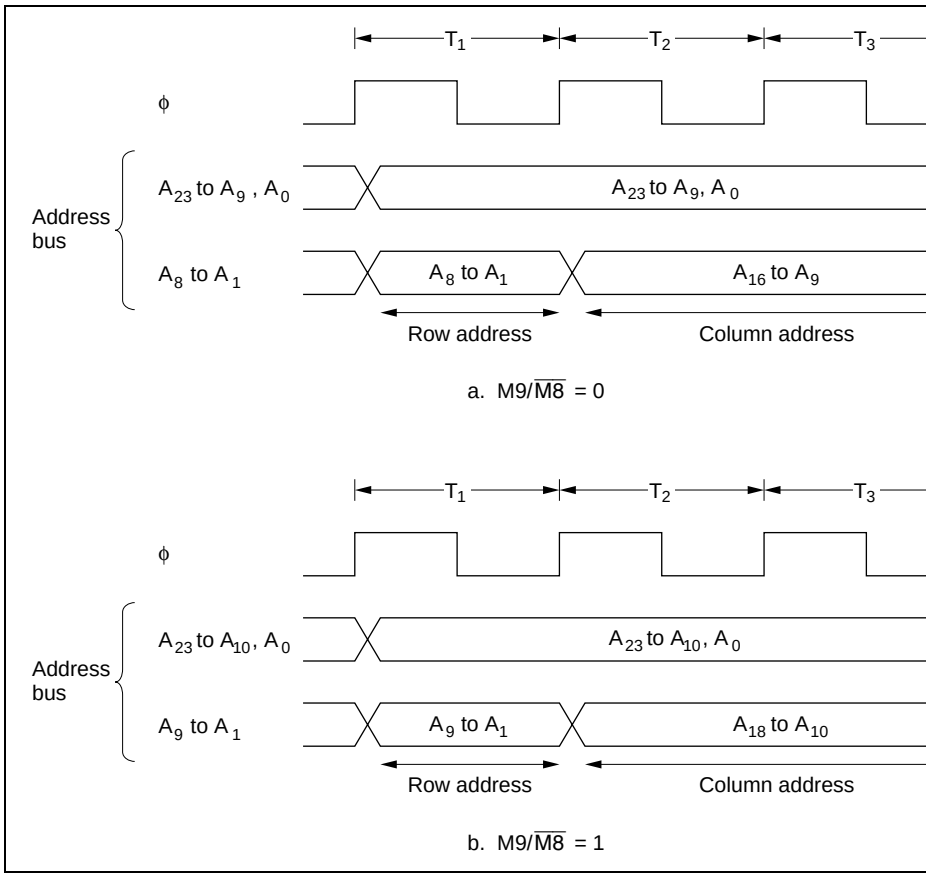


Figure 7.4 Multiplexed Address Output (Example without Wait State

H8/3048B Group Pin	CAS/$\overline{WE}$ = 0 (2$\overline{WE}$ Mode)	CAS/$\overline{WE}$ = 1 (2$\overline{CAS}$ Mo
$\overline{HWR}$	$\overline{UW}$	$\overline{UCAS}$
$\overline{LWR}$	$\overline{LW}$	$\overline{LCAS}$
$\overline{RD}$	$\overline{CAS}$	$\overline{WE}$
$\overline{CS_3}$	$\overline{RAS}$	$\overline{RAS}$

Figure 7.5 (1) shows the interface timing for 2$\overline{WE}$ DRAM. Figure 7.5 (2) shows the int
timing for 2$\overline{CAS}$ DRAM.

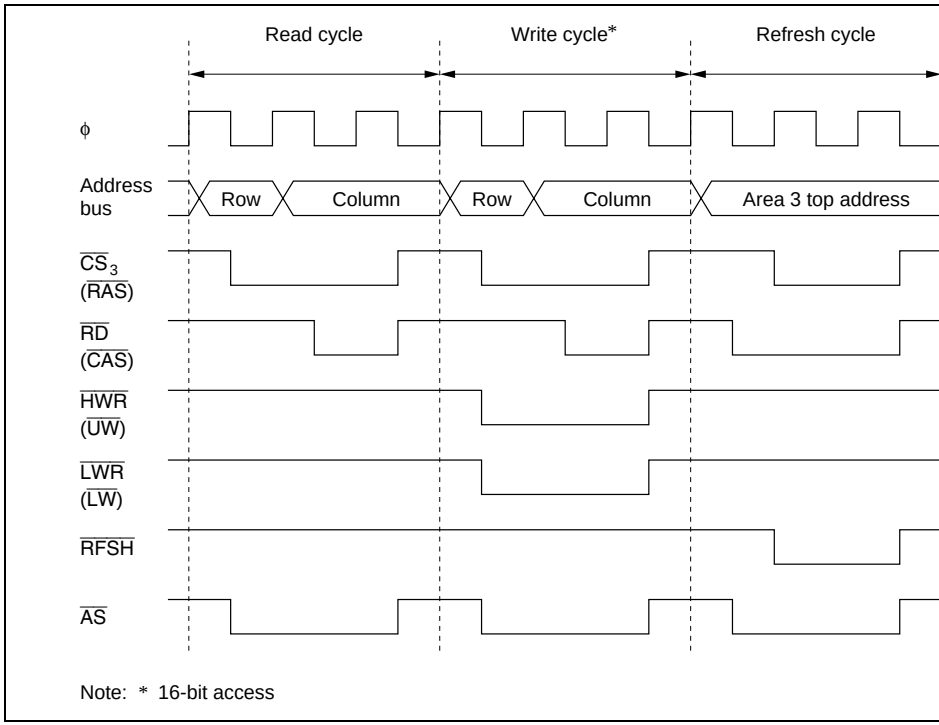


Figure 7.5(1) DRAM Control Signal Output Timing (2$\overline{WE}$ Mode)

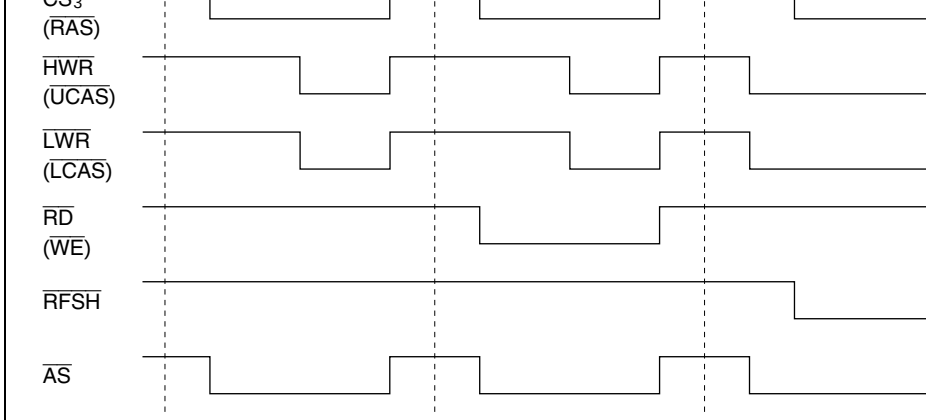


Figure 7.5(2) DRAM Control Signal Output Timing (2CAS Mode)

Refresh Cycle Priority Order

When there are simultaneous bus requests, the priority order is:

(High) External bus master > refresh controller > DMA controller > CPU

For details see section 6.3.7, Bus Arbiter Operation.

Wait State Insertion

When bit AST3 is set to 1 in ASTCR, bus controller settings can cause wait states to b
into bus cycles and refresh cycles. For details see section 6.3.5, Wait Modes.

Table 7.7 Pin States in Software Standby Mode (1) (PSRAME = 0, DRAME = 1

	Software Standby Mode			
	SRFMD = 0		SRFMD = 1 (self-refres	
Signal	**CAS/$\overline{WE}$ = 0**	**CAS/$\overline{WE}$ = 1**	**CAS/$\overline{WE}$ = 0**	**CAS/$\overline{W}$**
$\overline{HWR}$	High-impedance	High-impedance	High	Low
$\overline{LWR}$	High-impedance	High-impedance	High	Low
$\overline{RD}$	High-impedance	High-impedance	Low	High
$\overline{CS_3}$	High	High	Low	Low
$\overline{RFSH}$	High	High	Low	Low

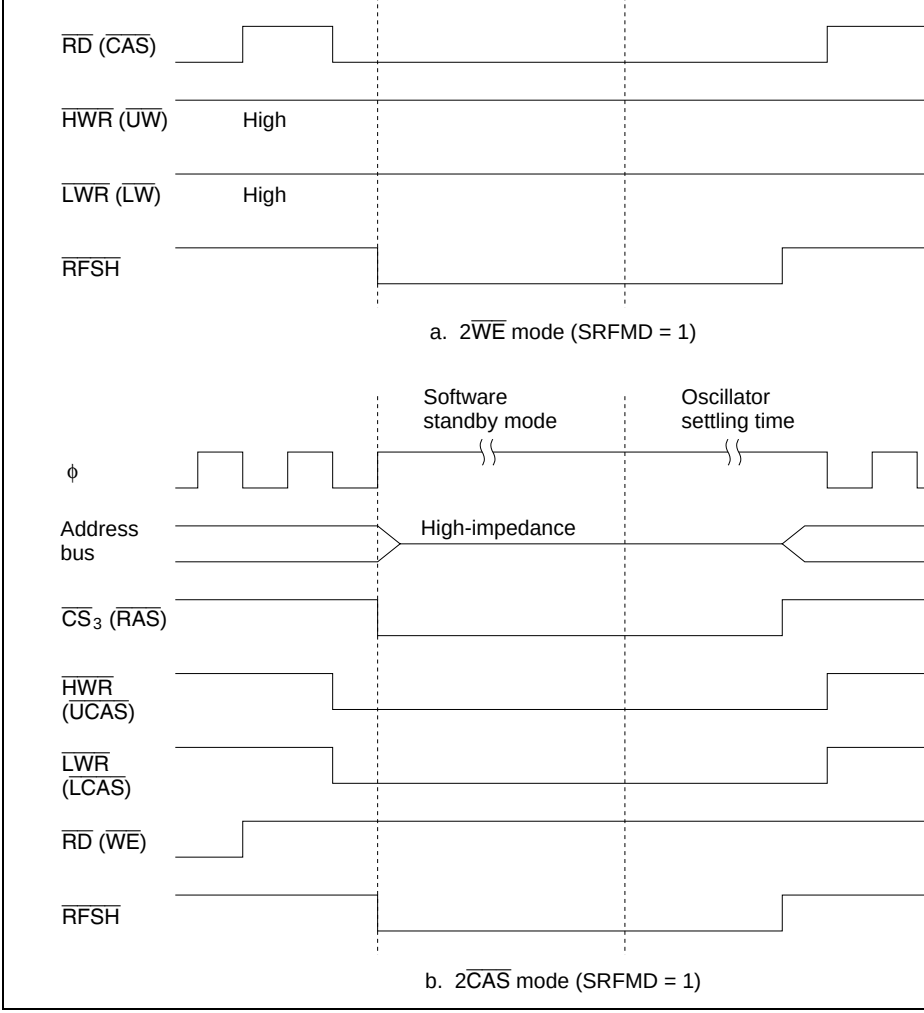


Figure 7.6 Signal Output Timing in Self-Refresh Mode (PSRAME = 0, DRA

Figure 7.7 shows typical interconnections to a 2WE 1-Mbit DRAM, and the correspond address map. Figure 7.8 shows a setup procedure to be followed by a program for this e After power-up the DRAM must be refreshed to initialize its internal state. Initializatio certain length of time, which can be measured by using an interrupt from another timer by counting the number of times RTMCSR bit 7 (CMF) is set. Note that no refresh cyc executed for the first refresh request after exit from the reset state or standby mode (the the CMF flag is set; see figure 7.3). When using this example, check the DRAM device characteristics carefully and use a procedure that fits them.

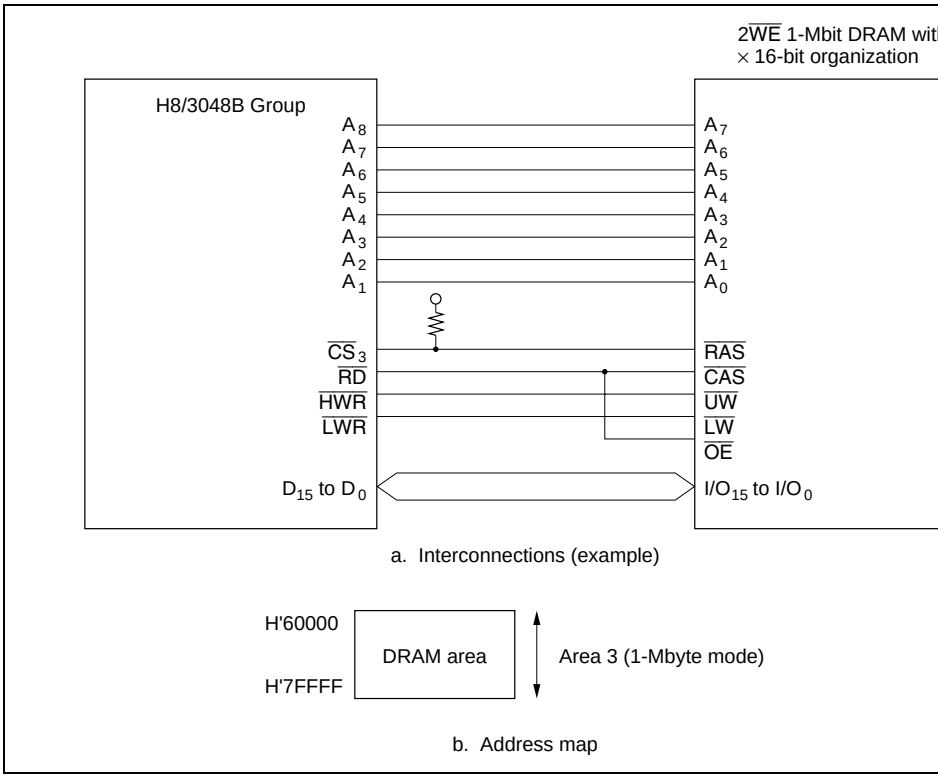


Figure 7.7 Interconnections and Address Map for 2WE 1-Mbit DRAM (Exa

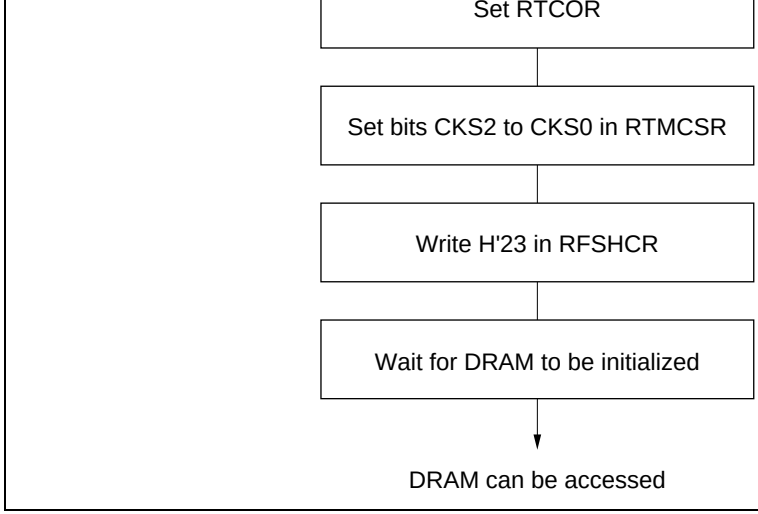
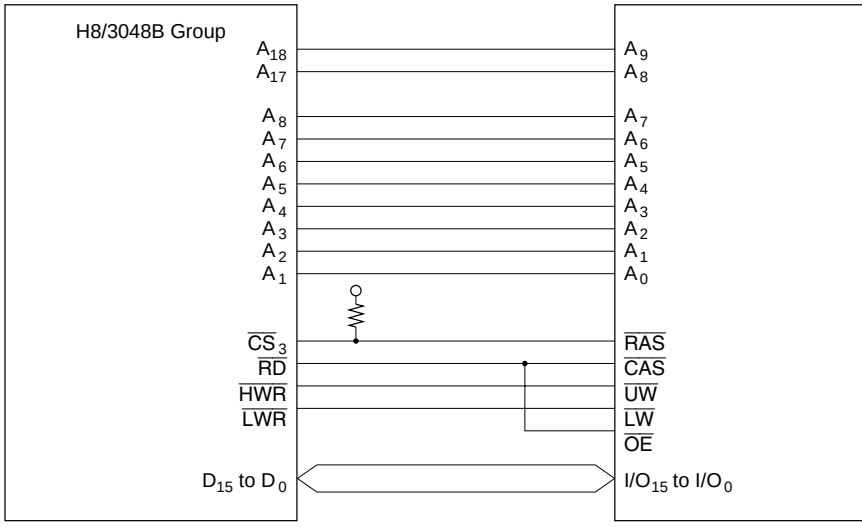
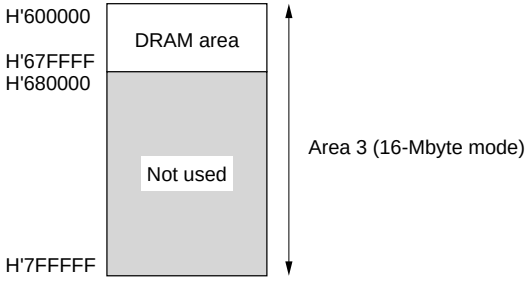


Figure 7.8 Setup Procedure for 2$\overline{\text{WE}}$ 1-Mbit DRAM (1-Mbyte Mode

2WE 4-Mbit DRAM with 10-bi
row address, 8-bit column ad
and ×16-bit organization



a. Interconnections (example)



b. Address map

Figure 7.9 Interconnections and Address Map for 2$\overline{\text{WE}}$ 4-Mbit DRAM (Exa

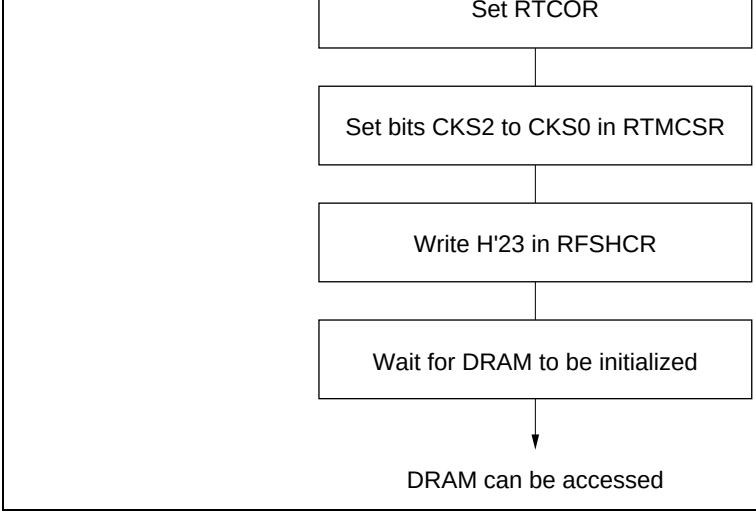


Figure 7.10 Setup Procedure for 2$\overline{WE}$ 4-Mbit DRAM with 10-Bit Row Addres Column Address (16-Mbyte Mode)

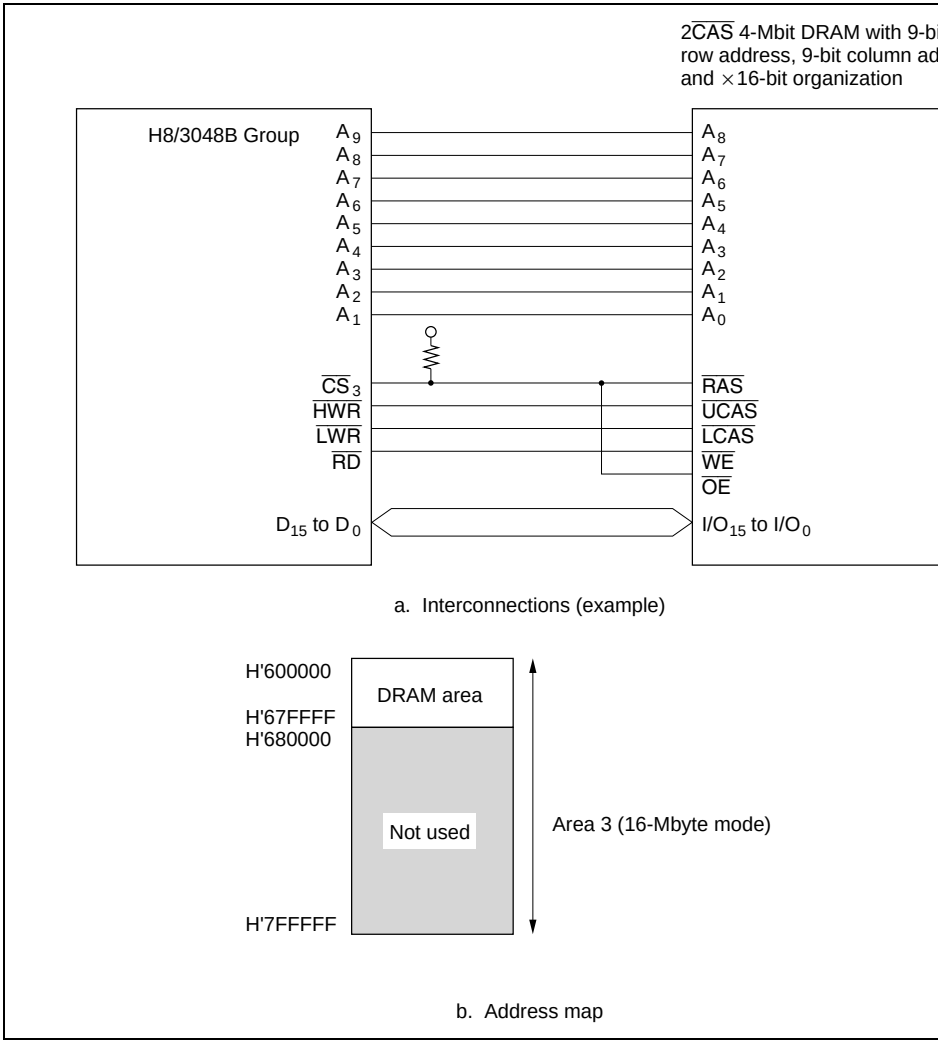


Figure 7.11 Interconnections and Address Map for 2CAS 4-Mbit DRAM (Ex

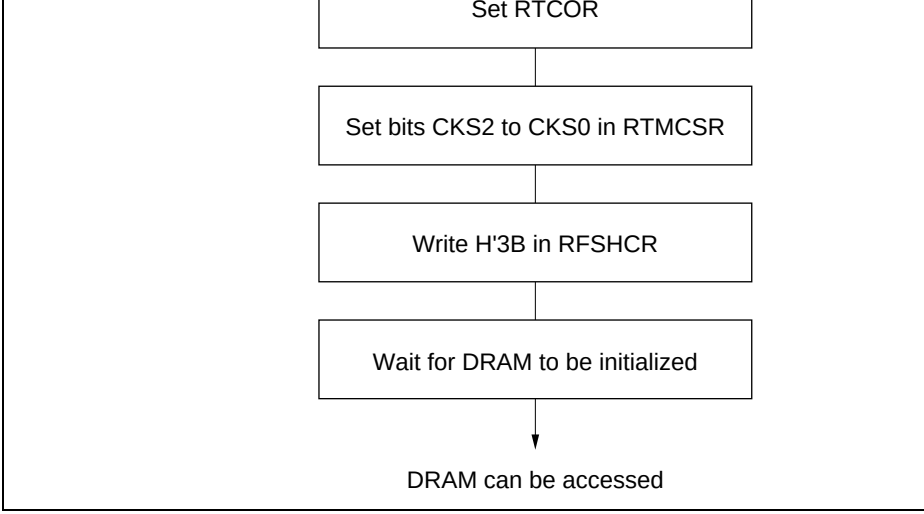
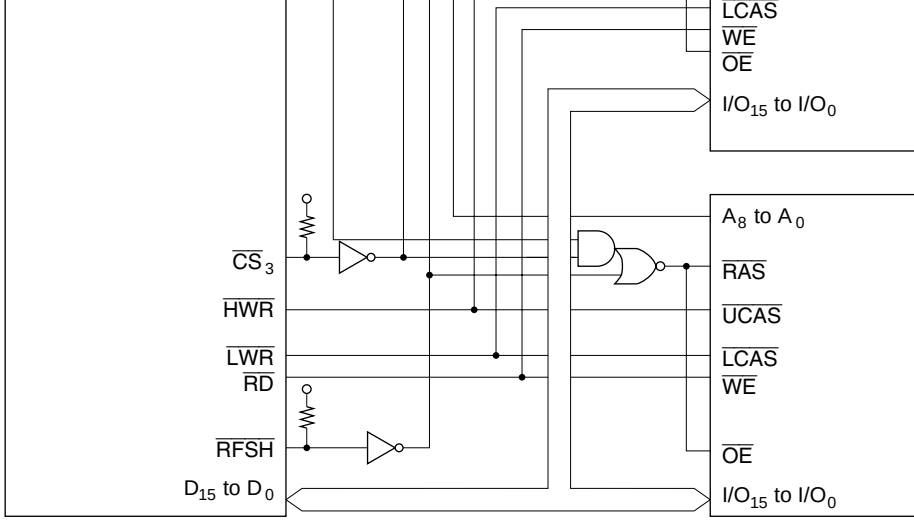


Figure 7.12 Setup Procedure for 2$\overline{\text{CAS}}$ 4-Mbit DRAM with 9-Bit Row Addres Column Address (16-Mbyte Mode)

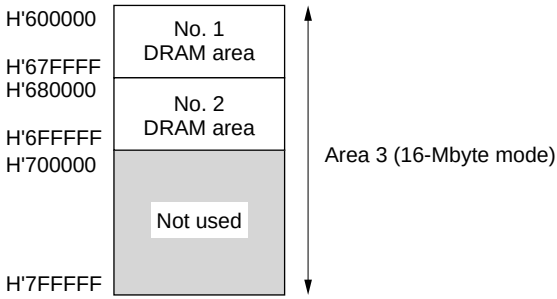
Example 4: Connection to Multiple 4-Mbit DRAM Chips (16-Mbyte Mode)

Figure 7.13 shows an example of interconnections to two 2$\overline{\text{CAS}}$ 4-Mbit DRAM chips, corresponding address map. Up to four DRAM chips can be connected to area 3 by de upper address bits A_{19} and A_{20}.

Figure 7.14 shows a setup procedure to be followed by a program for this example. Th this example has 9-bit row addresses and 9-bit column addresses. Both chips must be simultaneously, so the $\overline{\text{RFSH}}$ pin must be used.



a. Interconnections (example)



b. Address map

Figure 7.13 Interconnections and Address Map for Multiple 2$\overline{\text{CAS}}$ 4-Mbit DRA (Example)

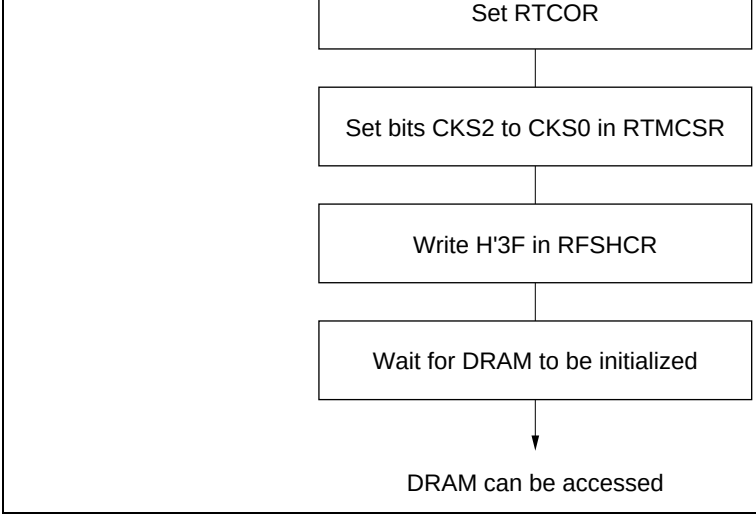


Figure 7.14 Setup Procedure for Multiple 2$\overline{\text{CAS}}$ 4-Mbit DRAM Chips with 9 Address and 9-Bit Column Address (16-Mbyte Mode)

7.3.3 Pseudo-Static RAM Refresh Control

Refresh Request Interval and Refresh Cycle Execution

The refresh request interval is determined as in a DRAM interface, by the settings of I bits CKS2 to CKS0 in RTMCSR. The numbers of states required for pseudo-static RA read/write cycles and refresh cycles are the same as for DRAM (see table 7.4). The sta transitions are as shown in figure 7.3.

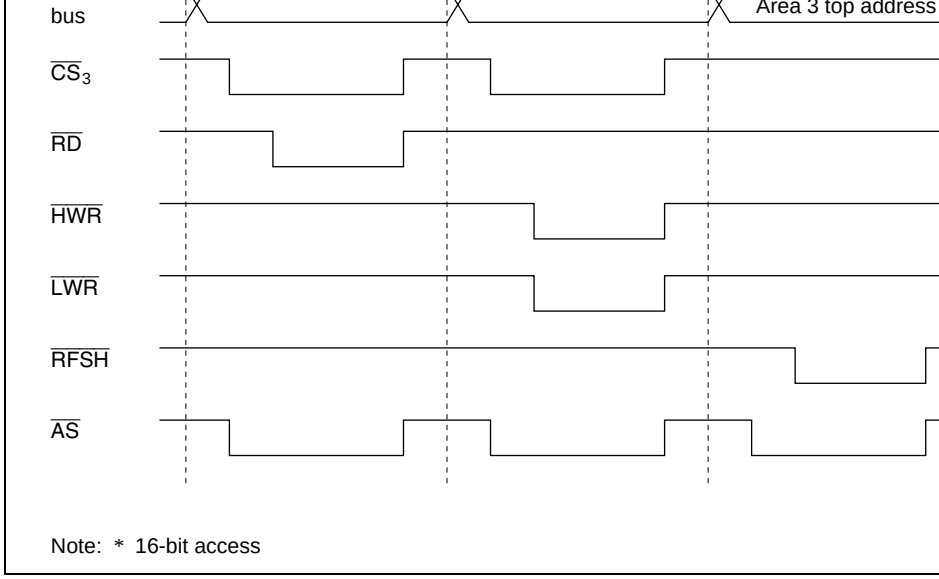


Figure 7.15 Pseudo-Static RAM Control Signal Output Timing

Refresh Cycle Priority Order

When there are simultaneous bus requests, the priority order is:

(High) External bus master > refresh controller > DMA controller > CPU

For details see section 6.3.7, Bus Arbiter Operation.

Wait State Insertion

When bit AST3 is set to 1 in ASTCR, the wait state controller (WSC) can insert wait st
bus cycles and refresh cycles. For details see section 6.3.5, Wait Modes.

Table 7.8 Pin States in Software Standby Mode (2) (PSRAME = 1, DRAME =

	Software Standby Mode	
Signal	**SRFMD = 0**	**SRFMD = 1 (Self-Refresh Mod**
$\overline{CS_3}$	High	High
$\overline{RD}$	High-impedance	High-impedance
$\overline{HWR}$	High-impedance	High-impedance
$\overline{LWR}$	High-impedance	High-impedance
$\overline{RFSH}$	High	Low

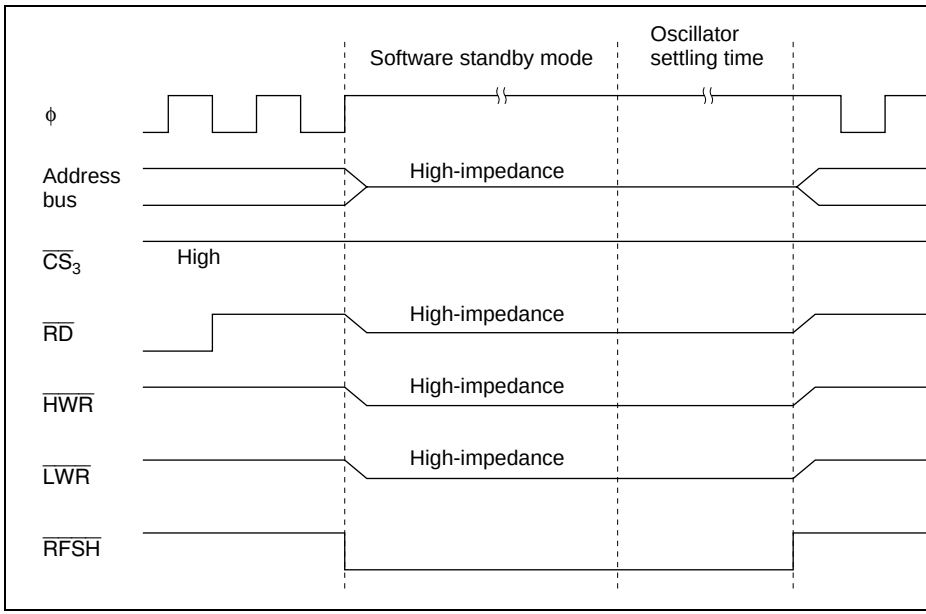


Figure 7.16 Signal Output Timing in Self-Refresh Mode (PSRAME = 1, DRA

Pseudo-static RAM may have separate OE and RFSH pins, or these may be combined
$\overline{OE}$/$\overline{RFSH}$ pin. Figure 7.17 shows an example of a circuit for generating an $\overline{OE}$/$\overline{RFSH}$ s
Check the device characteristics carefully, and design a circuit that fits them. Figure 7.1
setup procedure to be followed by a program.

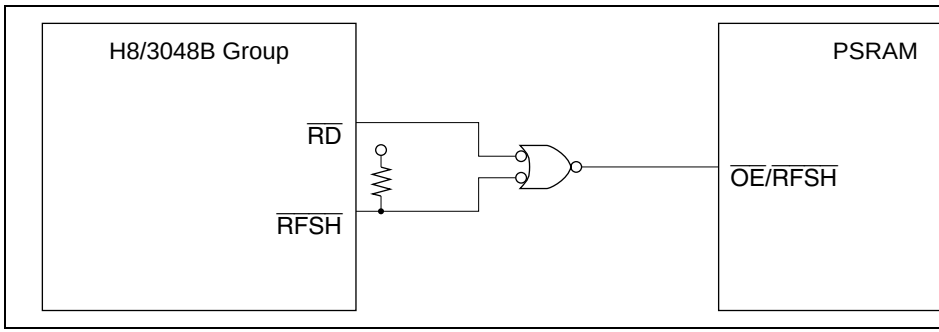


Figure 7.17 Interconnection to Pseudo-Static RAM with $\overline{OE}$/$\overline{RFSH}$ Signal (Ex

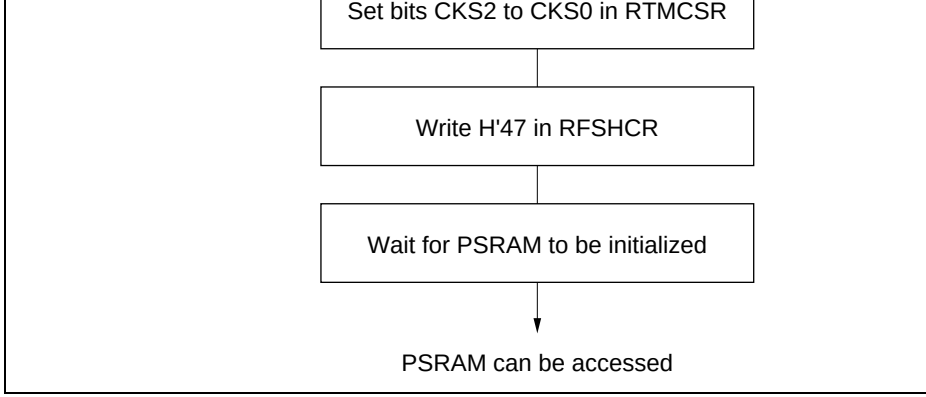


Figure 7.18 Setup Procedure for Pseudo-Static RAM

The CMF flag in RTCSR is set to 1 by a compare match signal output when the RTCO
RTCNT values match. The compare match signal is generated in the last state in which
match (when RTCNT is updated from the matching value to a new value). Accordingly
RTCNT and RTCOR match, the compare match signal is not generated until the next c
clock pulse. Figure 7.19 shows the timing.

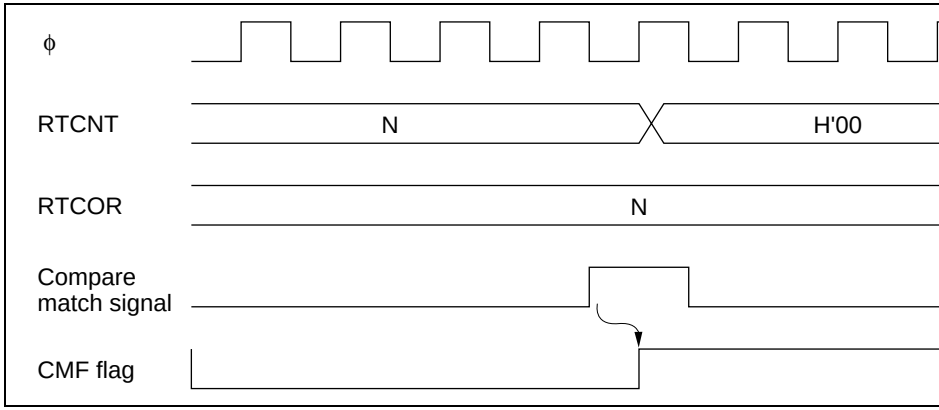


Figure 7.19 Timing of Setting of CMF Flag

Operation in Power-Down State

The interval timer function operates in sleep mode. It does not operate in hardware stan
In software standby mode RTCNT and RTMCSR bits 7 and 6 are initialized, but RTM
to 3 and RTCOR retain their settings prior to the transition to software standby mode.

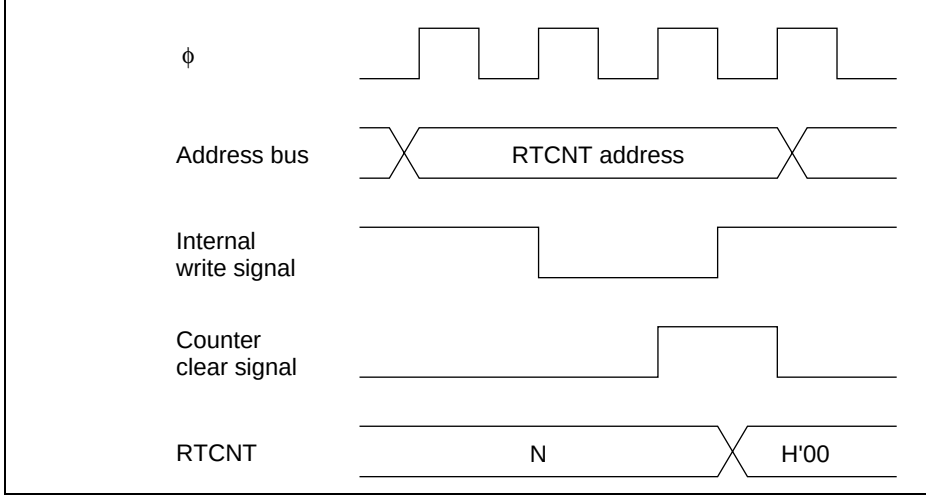


Figure 7.20 Contention between RTCNT Write and Clear

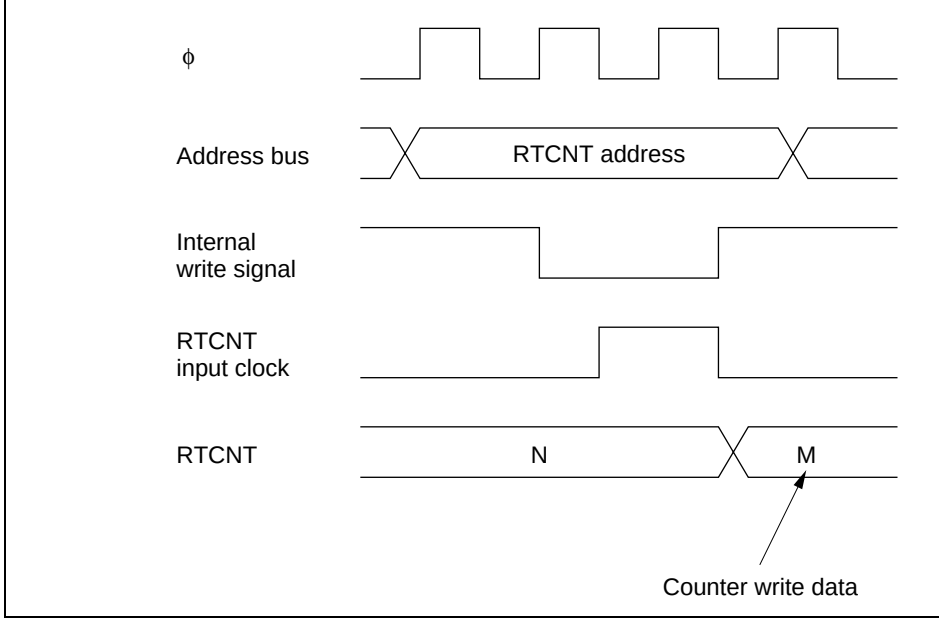


Figure 7.21 Contention between RTCNT Write and Increment

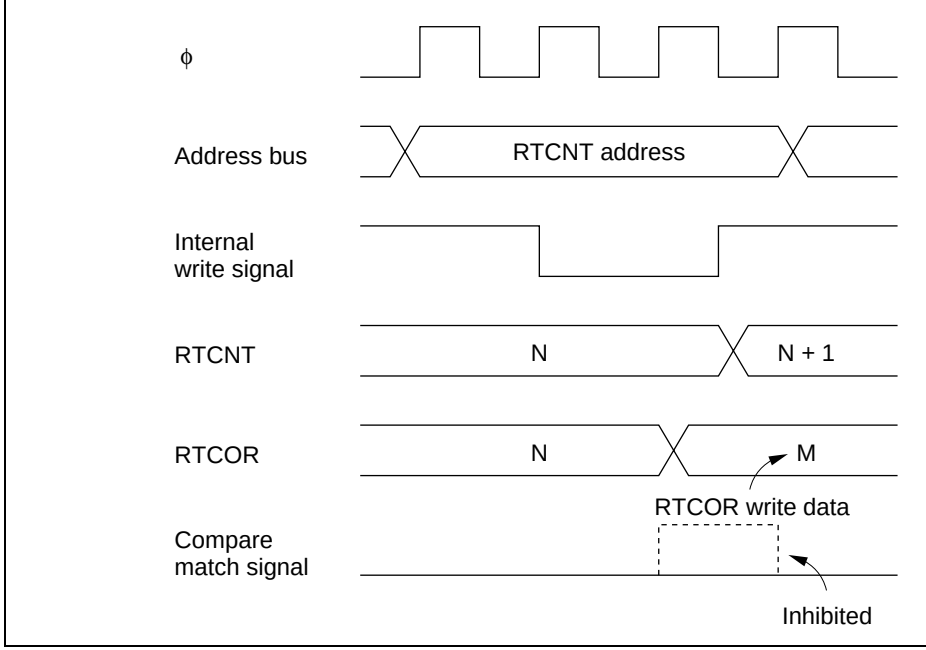
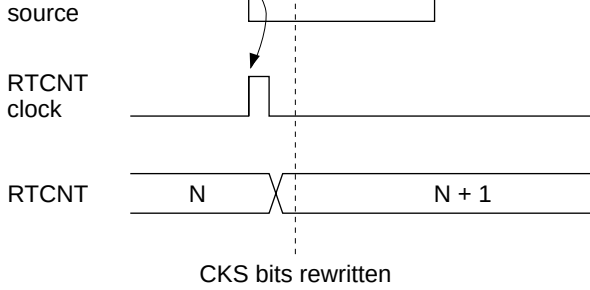


Figure 7.22 Contention between RTCOR Write and Compare Matc

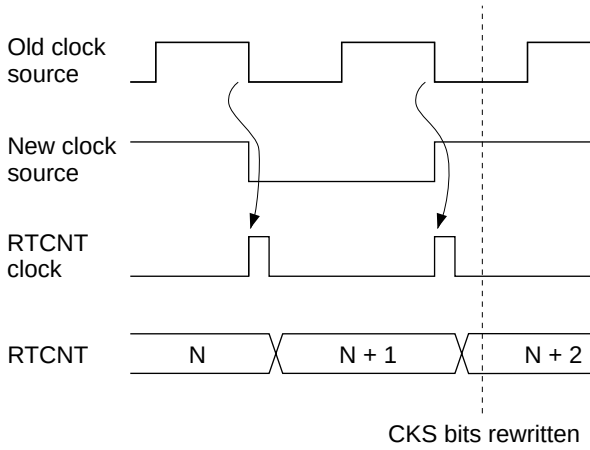
RTCNT Operation at Internal Clock Source Switchover

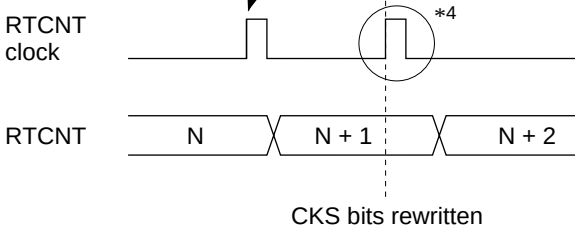
Switching internal clock sources may cause RTCNT to increment, depending on the s timing. Table 7.9 shows the relation between the time of the switchover (by writing to to CKS0) and the operation of RTCNT.

The RTCNT input clock is generated from the internal clock source by detecting the f of the internal clock. If a switchover is made from a high clock source to a low clock s case No. 3 in table 7.9, the switchover will be regarded as a falling edge, an RTCNT c will be generated, and RTCNT will be incremented.

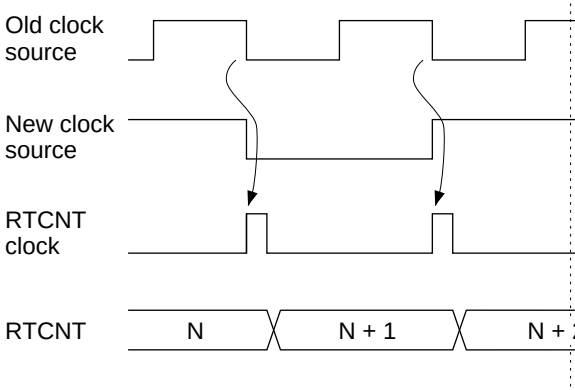


2 Low → high switchover*2





4 High → high switchover*4



Notes: 1. Including switchovers from a low clock source to the halted state, and from 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

When using the DRAM or pseudo-static RAM refresh function, note the following poi

- With the refresh controller, if directly connected DRAM or PSRAM is disconnecte $P8_0/\overline{RFSH}/\overline{IRQ_0}$ pin and the $P8_1/\overline{CS_3}/\overline{IRQ_1}$ pin may both become low-level outputs simultaneously.

Note: * When the DRAM enable bit (DRAME) or PSRAM enable bit (PSRAME) i refresh control register (RFSHCR) is cleared to 0 after being set to 1.

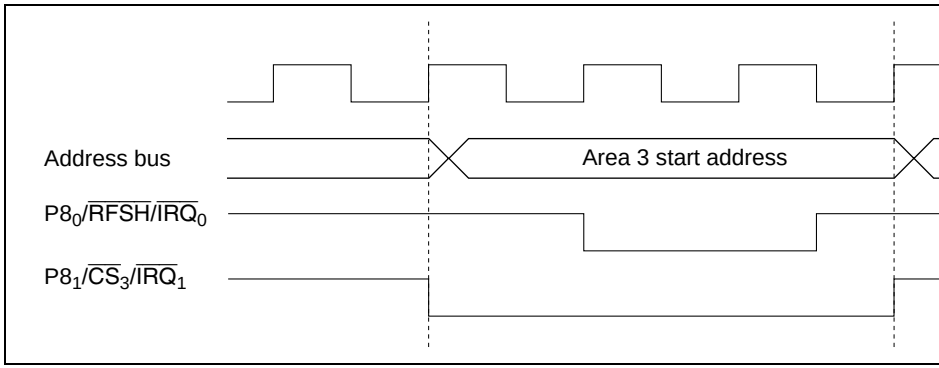


Figure 7.23 Operation when DRAM/PSRAM Connection Is Switched

- Refresh cycles are not executed while the bus is released, during software standby m when a bus cycle is greatly prolonged by insertion of wait states. When these condi other means of refreshing are required.
- If refresh requests occur while the bus is released, the first request is held and one r is executed after the bus-released state ends. Figure 7.24 shows the bus cycles in thi

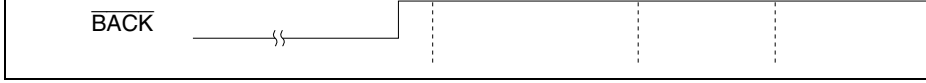


Figure 7.24 Refresh Cycles when Bus Is Released

- If a bus cycle is prolonged by insertion of wait states, the first refresh request is he bus-released state.
- If there is contention with a bus request from an external bus master when making to software standby mode, a one-state bus-released state may occur immediately b transition to software standby mode (see figure 7.25).

  When using software standby mode, clear the BRLE bit to 0 in BRCR before exec SLEEP instruction.

  When making a transition to self-refresh mode, the strobe waveform output may n guaranteed due to the same kind of contention. This, too, can be prevented by clea BRLE bit to 0 in BRCR.

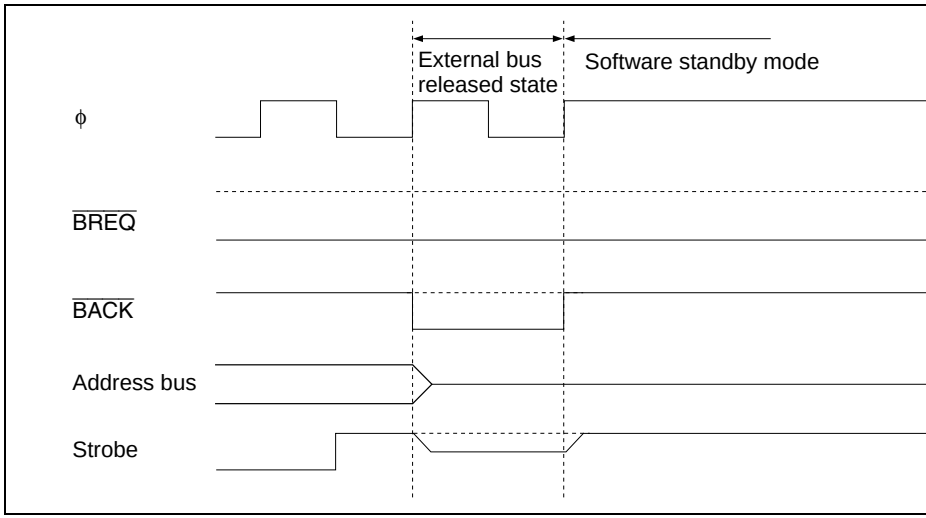


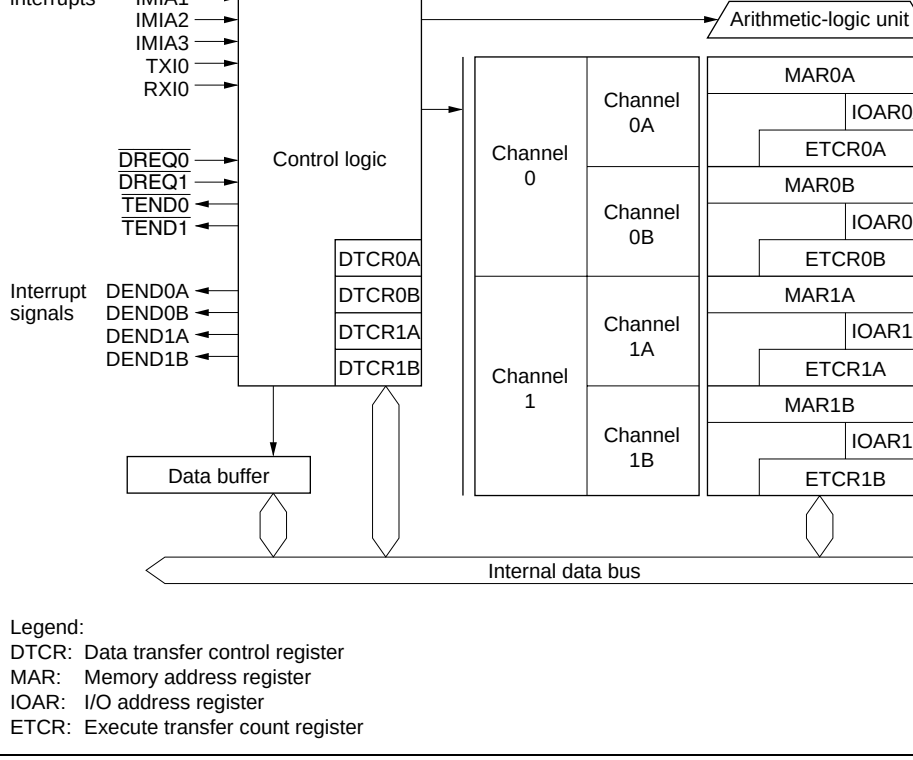
Figure 7.25 Contention between Bus-Released State and Software Standby

When the DMAC controller is not used, it can be independently halted to conserve pow
details see section 20.6, Module Standby Function.

8.1.1 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 integrated timer unit (ITU) compare match/input capture interrupts (four
  — Serial communication interface (SCI channel 0) transmit-data-empty/receive-d
    interrupts
  — External requests
  — Auto-request



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

Figure 8.1 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 ITU channels 0 to 3 • Transmit-data-empty interrupt from SCI channel 0	24
		• Receive-data-full interrupt from SCI channel 0	8
		• External request	24

	(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		
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 ITU channels 0 to 3 • External request	24

	Transfer end 0	$TEND_0$	Output	Transfer end on DMAC ch
1	DMA request 1	$DREQ_1$	Input	External request for DMAC
	Transfer end 1	$TEND_1$	Output	Transfer end on DMAC ch

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

8.1.5 Register Configuration

Table 8.3 lists the DMAC registers.

	H'FF24	Execute transfer count register 0AH	ETCR0AH	R/W	Un
	H'FF25	Execute transfer count register 0AL	ETCR0AL	R/W	Un
	H'FF27	Data transfer control register 0A	DTCR0A	R/W	H'0
	H'FF28	Memory address register 0BR	MAR0BR	R/W	H'F
	H'FF29	Memory address register 0BE	MAR0BE	R/W	Un
	H'FF2A	Memory address register 0BH	MAR0BH	R/W	Un
	H'FF2B	Memory address register 0BL	MAR0BL	R/W	Un
	H'FF2E	I/O address register 0B	IOAR0B	R/W	Un
	H'FF2C	Execute transfer count register 0BH	ETCR0BH	R/W	Un
	H'FF2D	Execute transfer count register 0BL	ETCR0BL	R/W	Un
	H'FF2F	Data transfer control register 0B	DTCR0B	R/W	H'0
1	H'FF30	Memory address register 1AR	MAR1AR	R/W	H'F
	H'FF31	Memory address register 1AE	MAR1AE	R/W	Un
	H'FF32	Memory address register 1AH	MAR1AH	R/W	Un
	H'FF33	Memory address register 1AL	MAR1AL	R/W	Un
	H'FF36	I/O address register 1A	IOAR1A	R/W	Un
	H'FF34	Execute transfer count register 1AH	ETCR1AH	R/W	Un
	H'FF35	Execute transfer count register 1AL	ETCR1AL	R/W	Un
	H'FF37	Data transfer control register 1A	DTCR1A	R/W	H'0
	H'FF38	Memory address register 1BR	MAR1BR	R/W	H'F
	H'FF39	Memory address register 1BE	MAR1BE	R/W	Un
	H'FF3A	Memory address register 1BH	MAR1BH	R/W	Un
	H'FF3B	Memory address register 1BL	MAR1BL	R/W	Un
	H'FF3E	I/O address register 1B	IOAR1B	R/W	Un
	H'FF3C	Execute transfer count register 1BH	ETCR1BH	R/W	Un
	H'FF3D	Execute transfer count register 1BL	ETCR1BL	R/W	Un
	H'FF3F	Data transfer control register 1B	DTCR1B	R/W	H'0

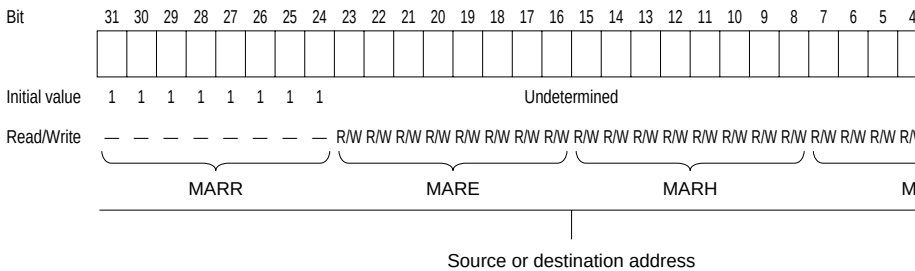
Note: * The lower 16 bits of the address are indicated.

Channel	DTS2A	DTS1A	Description
0	1	1	DMAC channel 0 operates as one channel in full mode
	Other than above		DMAC channels 0A and 0B operate as two indep channels in short address mode
1	1	1	DMAC channel 1 operates as one channel in full mode
	Other than above		DMAC channels 1A and 1B operate as two indep channels in short address mode

8.2.1 Memory Address Registers (MAR)

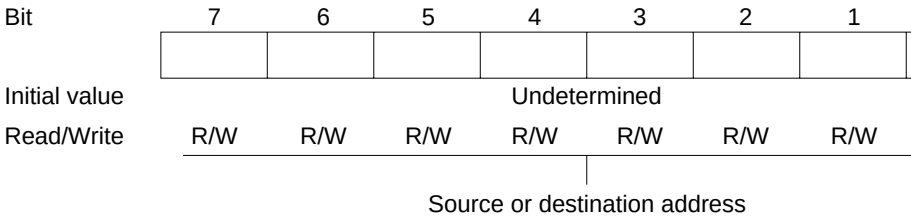
A memory address register (MAR) is a 32-bit readable/writable register that specifies destination address. The transfer direction is determined automatically from the activa

An MAR consists of four 8-bit registers designated MARR, MARE, MARH, and MA 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 D activated: as a destination address register if activation is by a receive-data-full interru serial communication interface (SCI) (channel 0), and as a source address register oth

An I/O address register (IOAR) is an 8-bit readable/writable register that specifies a sou
destination address. The IOAR value is the lower 8 bits of the address. The upper 16 ad
are all 1 (H'FFFF).



An IOAR functions as a source or destination address register depending on how the D
activated: as a source address register if activation is by a receive-data-full interrupt fro
(channel 0), and as a destination address register otherwise.

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

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

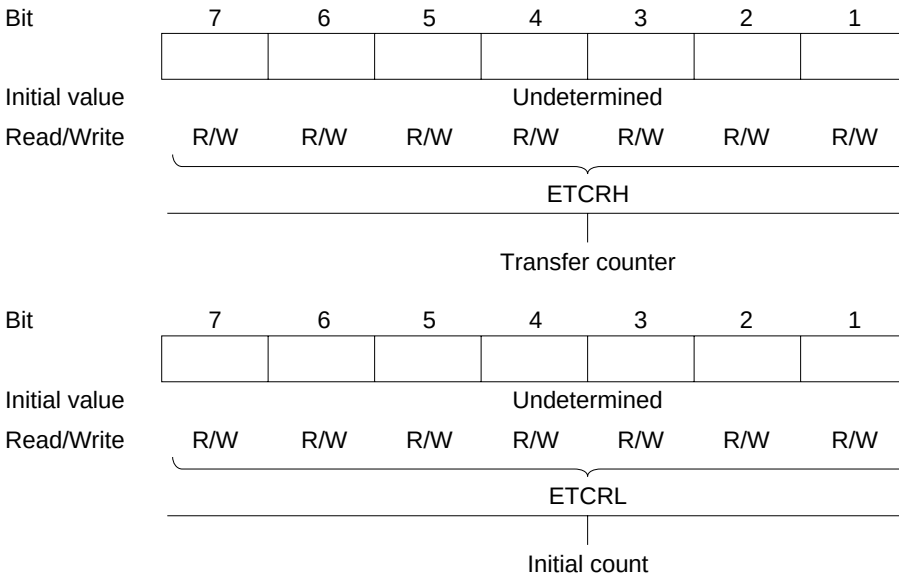
8.2.3 Execute Transfer Count Registers (ETCR)

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

In I/O mode and idle mode, ETCR functions as a 16-bit counter. The count is decreme

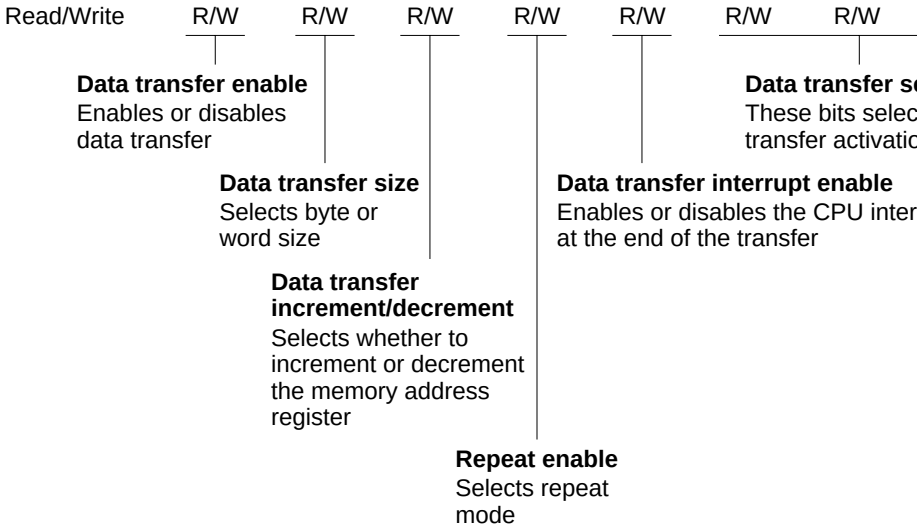
1 each time one transfer is executed. The transfer ends when the count reaches H'0000

Repeat mode



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

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. W
DTE bit is set to 1, the channel waits for a transfer to be requested, and executes the tra
activated as specified by bits DTS2 to DTS0. When DTE is 0, the channel is disabled a
accept transfer requests. DTE is set to 1 by reading the register when DTE is 0, then wr

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

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

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
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 mod
mode.

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

Operations in these modes are described in sections 8.4.2, I/O Mode, 8.4.3, Idle Mode
Repeat Mode.

Bit 3—Data Transfer Interrupt Enable (DTIE): Enables or disables the CPU interr
requested when the DTE bit is cleared to 0.

Bit 3: DTIE	Description
0	The DEND interrupt requested by DTE is disabled
1	The DEND interrupt requested by DTE is enabled

			channel 1
	1	0	Compare match/input capture A interru channel 2
		1	Compare match/input capture A interru channel 3
1	0	0	Transmit-data-empty interrupt from SC
		1	Receive-data-full interrupt from SCI ch
	1	0	Falling edge of $\overline{\text{DREQ}}$ input (channel B
			Transfer in full address mode (channel
		1	Low level of $\overline{\text{DREQ}}$ input (channel B)
			Transfer in full address mode (channel

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 channel
the priority order, see section 8.4.9, DMAC Multiple-Channel Operation.

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

source address register of the transfer, and MARB as the destination address register.

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

Bit	31	30	29	28	27	26	25	24	23	22	21	20	19	18	17	16	15	14	13	12	11	10	9	8	7	6	5	4
Initial value	1	1	1	1	1	1	1	1	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	R/W	R/W	R/W	R/W	R/W	R/
	MARR								MARE								MARH								M			

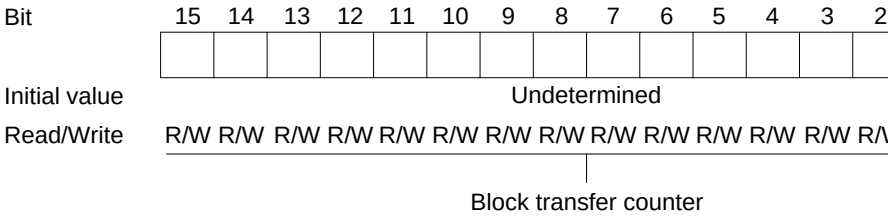
Source or destination address

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

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

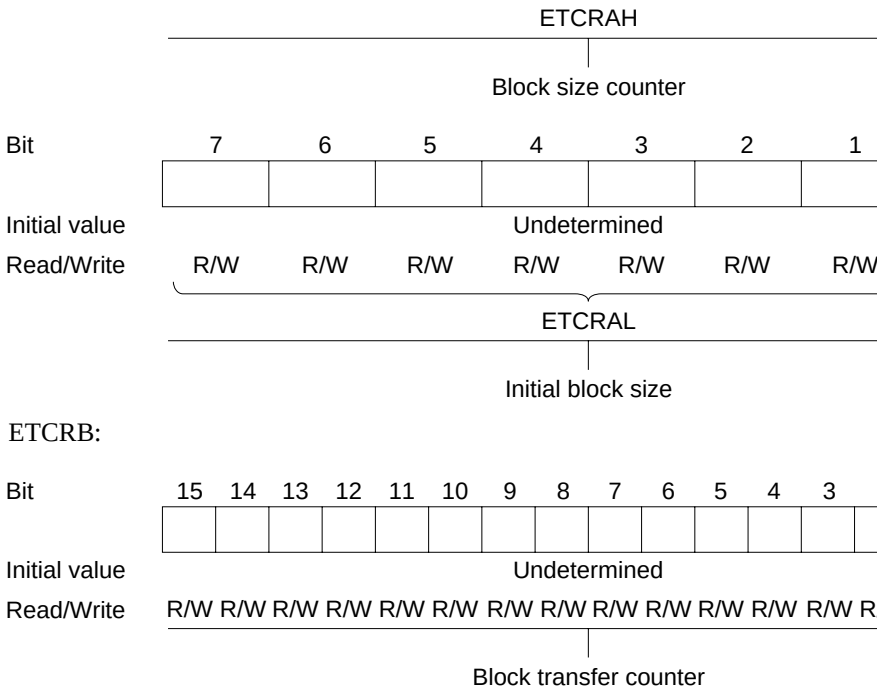
8.3.2 I/O Address Registers (IOAR)

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 decremente each time one transfer is executed. The transfer ends when the count reaches H'0000. E not used.

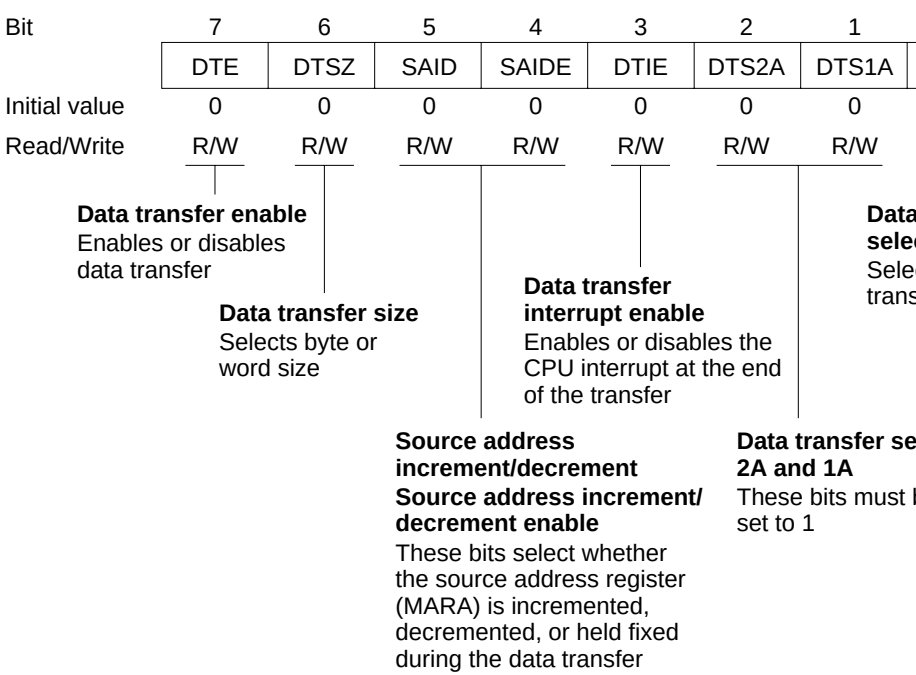


ETCRB:

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

Bit 7—Data Transfer Enable (DTE): Together with the DTME bit in DTCRB, this bi
or disables data transfer on the channel. When the DTME and DTE bits are both set to
channel is enabled. If auto-request is specified, data transfer begins immediately. Other
channel waits for transfers to be requested. When the specified number of transfers hav
completed, the DTE bit is automatically cleared to 0. When DTE is 0, the channel is dis
does not accept transfer requests. DTE is set to 1 by reading the register when DTE is 0
writing 1.

Bit 6: DTSZ	Description
0	Byte-size transfer
1	Word-size transfer

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

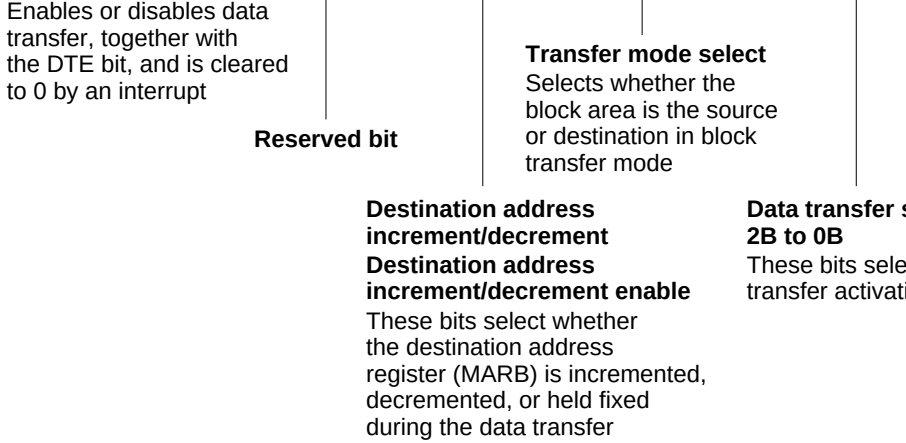
Bit 5: SAID	Bit 4: SAIDE	Description
0	0	MARA is held fixed
	1	MARA is incremented after each data transfer • If DTSZ = 0, MARA is incremented by 1 af transfer • If DTSZ = 1, MARA is incremented by 2 af transfer
1	0	MARA is held fixed
	1	MARA is decremented after each data transfe • If DTSZ = 0, MARA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

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 8.4.5, Normal Mode, and 8.4.6, Bl
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 DTCE enables or disables data transfer. When the DTME and DTE bits are both set to 1, the enabled. When an NMI interrupt occurs DTME is cleared to 0, suspending the transfe CPU can use the bus. The suspended transfer resumes when DTME is set to 1 again. F information on operation in block transfer mode, see section 8.6.6, 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 interr (
1	Data transfer is enabled

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

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 (In
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 channel 0 (I
		1	Compare match/input capture A interrupt channel 1
	1	0	Compare match/input capture A interrupt channel 2
		1	Compare match/input capture A interrupt channel 3
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 char activated in a priority order, highest priority first. For the priority order, see section 8.4 Multiple-Channel Operation.

Transfer Mode		Activation	Notes
Short address mode	I/O mode Idle mode Repeat mode	Compare match/input capture A interrupt from ITU channels 0 to 3	• Up to four chan operate indepen • Only the B chan support externa
		Transmit-data-empty and receive-data-full interrupts from SCI channel 0	
		External request	
Full address mode	Normal mode	Auto-request	• A and B channe paired; up to tw are available • Burst mode or c mode can be se auto-requests
		External request	
	Block transfer mode	Compare match/input capture A interrupt from ITU channels 0 to 3	
		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 transfers are A CPU interrupt can be requested at completion of the designated number of transfers. address and one 8-bit address are specified. The transfer direction is determined automa from the activation source.

Idle Mode

One byte or word is transferred per request. A designated number of these transfers are A CPU interrupt can be requested at completion of the designated number of transfers. address and one 8-bit address are specified. The addresses are held fixed. The transfer d determined automatically from the activation source.

Normal Mode

Auto-request: The DMAC is activated by register setup alone, and continues executi
until the designated number of transfers have been completed. A CPU interrupt can be
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 the d
  number of transfers have been completed.

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

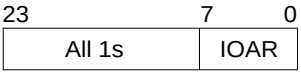
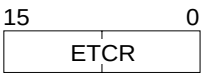
Block Transfer Mode

One block of a specified size is transferred per request. A designated number of block
executed. At the end of each block transfer, one address is restored to its initial value.
designated number of blocks have been transferred, a CPU interrupt can be requested.
addresses are 24-bit addresses.

in MAR to the address specified in IOAR otherwise.

Table 8.6 indicates the register functions in I/O mode.

Table 8.6 Register Functions in I/O Mode

Register	Function: Activated by SCI0 Receive-Data-Full Interrupt	Function: Other Activation	Initial Setting	Ope
	Destination address register	Source address register	Destination or source address	Incre decr once trans
	Source address register	Destination address register	Source or destination address	Held
	Transfer counter	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, which is incremented or decremented as each byte or word is trans
IOAR specifies the lower 8 bits of a fixed address. The upper 16 bits are all 1s. IOAR i
incremented or decremented.

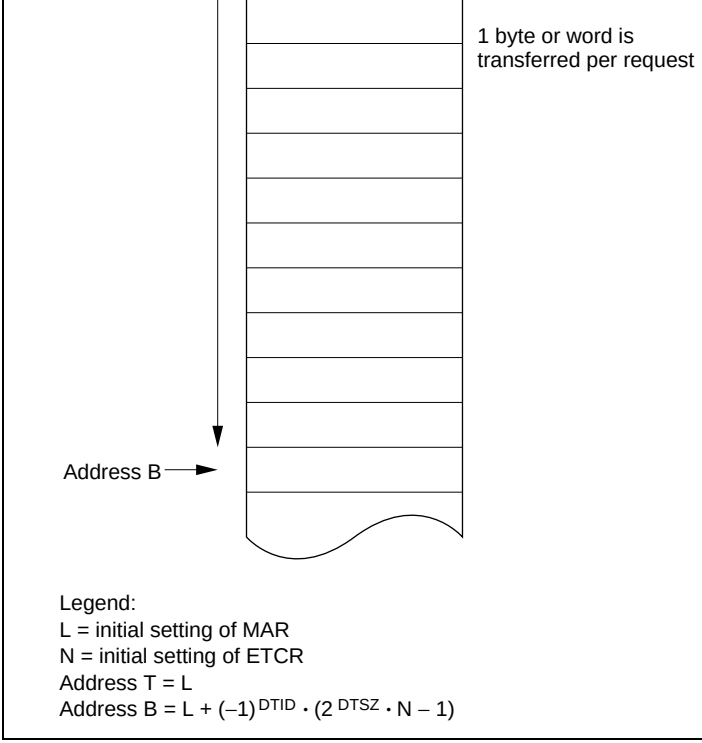
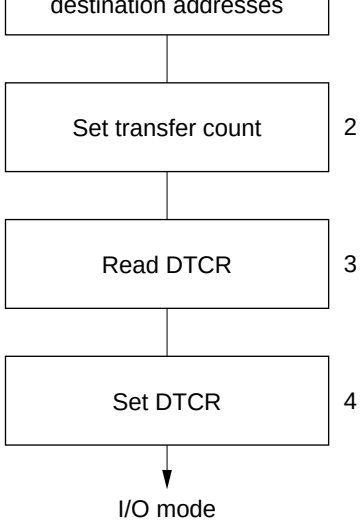


Figure 8.2 Operation in I/O Mode

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

Transfers can be requested (activated) by compare match/input capture A interrupts fr channels 0 to 3, transmit-data-empty and receive-data-full interrupts from SCI channe external request signals.



2. Set the transfer count in ETCR.
3. Read DTCR while the DTE bit is cleared t
4. Set the DTCR bits as follows.
   - Select the DMAC activation source with DTS2 to DTS0.
   - Set or clear the DTIE bit to enable or di the CPU interrupt at the end of the tran
   - Clear the RPE bit to 0 to select I/O mod
   - Select MAR increment or decrement wi DTID bit.
   - Select byte size or word size with the D
   - Set the DTE bit to 1 to enable the trans

Figure 8.3 I/O Mode Setup Procedure (Example)

8.4.3 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 num these transfers are executed. One address is specified in the memory address register (M other in the I/O address register (IOAR). The direction of transfer is determined automa from the activation source. The transfer is from the address specified in IOAR to the ad specified in MAR if activated by an SCI channel 0 receive-data-full interrupt, and from address specified in MAR to the address specified in IOAR otherwise.

Table 8.7 indicates the register functions in idle mode.

	register	register		
23 7 0 All 1s | IOAR	Source address register	Destination address register	Source or destination address	Hel
15 0 ETCR	Transfer counter	Transfer counter	Number of transfers	Dec onc tran H'0 rea tran

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. IOAR specifies the lower 8 bits of a fixed address. The upper 16 b 1s. MAR and IOAR are not incremented or decremented.

Figure 8.4 illustrates how idle mode operates.

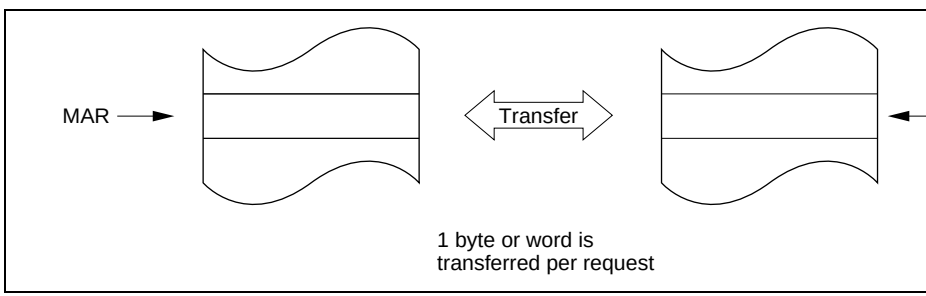


Figure 8.4 Operation in Idle Mode

For the detailed settings see section 8.2.4, Data Transfer Control Registers (DTCR).

Figure 8.5 shows a sample setup procedure for idle mode.

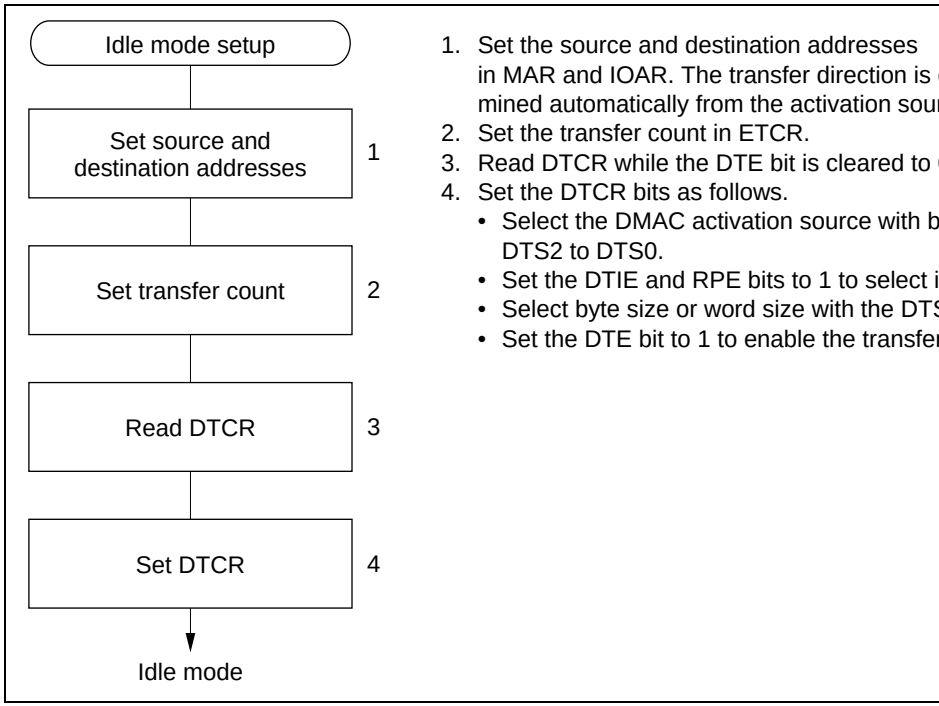
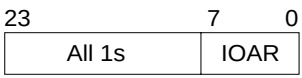
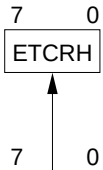
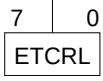


Figure 8.5 Idle Mode Setup Procedure (Example)

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

Table 8.8 indicates the register functions in repeat mode.

	register	register		each until then initia
	Source address register	Destination address register	Source or destination address	Held
	Transfer counter	Transfer counter	Number of transfers	Decr once trans H'00 reac reloa ETC
	Initial transfer count	Initial transfer count	Number of transfers	Held

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

In repeat mode ETCRH is used as the transfer counter while ETCRL holds the initial tr
count. ETCRH is decremented by 1 at each transfer until it reaches H'00, then is reload
ETCRL. MAR is also restored to its initial value, which is calculated from the DTSZ a
bits in DTCR. Specifically, MAR is restored as follows:

$$MAR \leftarrow MAR - (-1)^{DTID} \cdot 2^{DTSZ} \cdot ETCRL$$

ETCRH and ETCRL should be initially set to the same value.

In repeat mode transfers continue until the CPU clears the DTE bit to 0. After DTE is c
if the CPU sets DTE to 1 again, transfers resume from the state at which DTE was clea
CPU interrupt is requested.

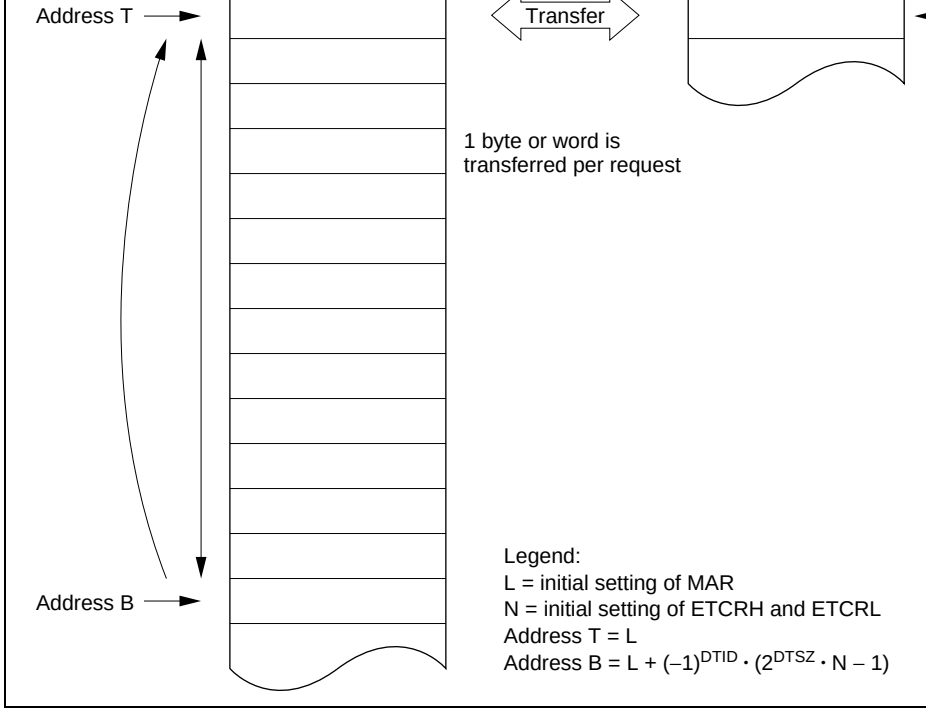
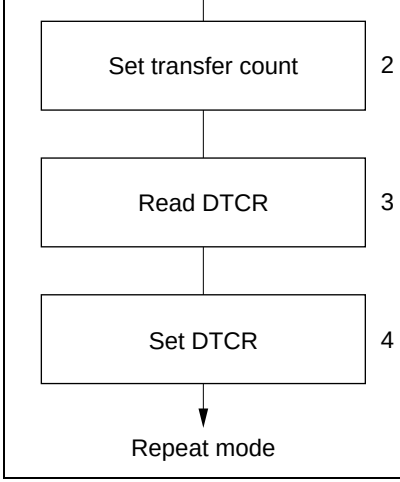


Figure 8.6 Operation in Repeat Mode

The transfer count is specified as an 8-bit value in ETCRH and ETCRL. The maximu count is 255, obtained by setting both ETCRH and ETCRL to H'FF.

Transfers can be requested (activated) by compare match/input capture A interrupts fr channels 0 to 3, transmit-data-empty and receive-data-full interrupts from SCI channe external request signals.

For the detailed settings see section 8.2.4, Data Transfer Control Registers (DTCR).



- DTS2 to DTS0.
- Clear the DTIE bit to 0 and set the RPE
to select repeat mode.
- Select MAR increment or decrement wit
DTID bit.
- Select byte size or word size with the D
- Set the DTE bit to 1 to enable the transf

Figure 8.7 Repeat Mode Setup Procedure (Example)

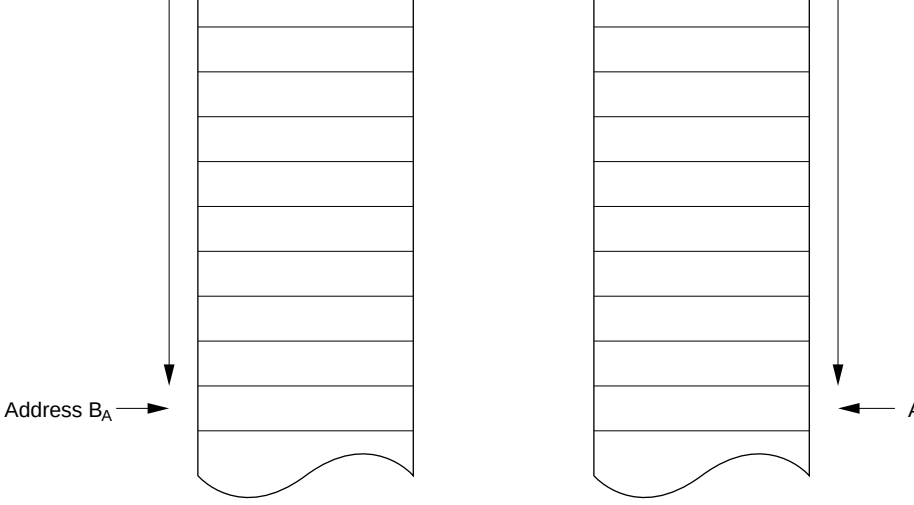
23 0 MARA	Source address register	Source address	Incremented decremented transfer, or he
23 0 MARB	Destination address register	Destination address	Incremented decremented transfer, or he
15 0 ETCRA	Transfer counter	Number of transfers	Decremented transfer

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, a CPU interrupt is requested at this time. The maximum tr is 65,536, obtained by setting ETCRA to H'0000.

Figure 8.8 illustrates how normal mode operates.

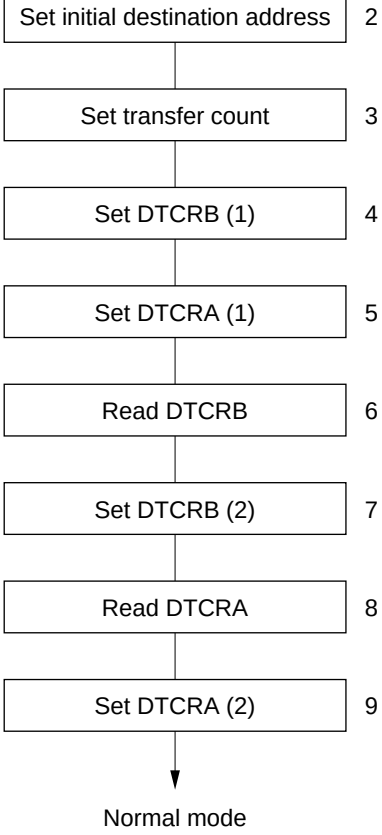


Legend:
L_A = initial setting of MARA
L_B = initial setting of MARB
N = initial setting of ETCRA
$T_A = L_A$
$B_A = L_A + SAIDE \cdot (-1)^{SAID} \cdot (2^{DTSZ} \cdot N - 1)$
$T_B = L_B$
$B_B = L_B + DAIDE \cdot (-1)^{DAID} \cdot (2^{DTSZ} \cdot N - 1)$

Figure 8.8 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 8.3.4, Data Transfer Control Registers (DTCR).

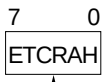
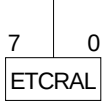


MARB is incremented, decremented, o
- Select the DMAC activation source with DTS2B to DTS0B.

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, o
   - Set or clear the DTIE bit to enable or di CPU interrupt at the end of the transfer
   - Clear the DTS0A bit to 0 and set the D and DTS1A bits to 1 to select normal m
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

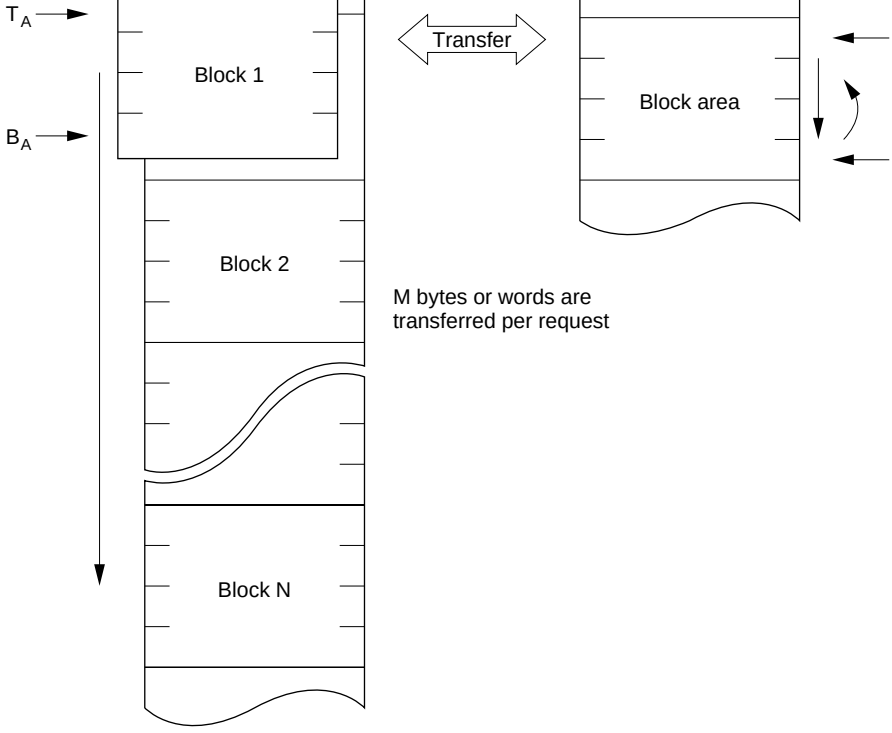
Note: Carry out settings 1 to 9 with the DEND interrupt masked in the CPU.
If an NMI interrupt occurs during the setup procedure, it may clear the DTME which case the transfer will not start.

Figure 8.9 Normal Mode Setup Procedure (Example)

Register	Function	Initial Setting	Operation
	Source address register	Source address	Incremented or decremented c transfer, or hel
	Destination address register	Destination address	Incremented or decremented c transfer, or hel
	Block size counter	Block size	Decremented c transfer until H reached, then r from ETCRAL
	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 8.10 Operation in Block Transfer Mode

cleared to 0 and the transfer ends. If the DTIE bit is set to 1, a CPU interrupt is request
time.

Figure 8.11 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 3, and by external request signals.

For the detailed settings see section 8.3.4, Data Transfer Control Registers (DTCR).

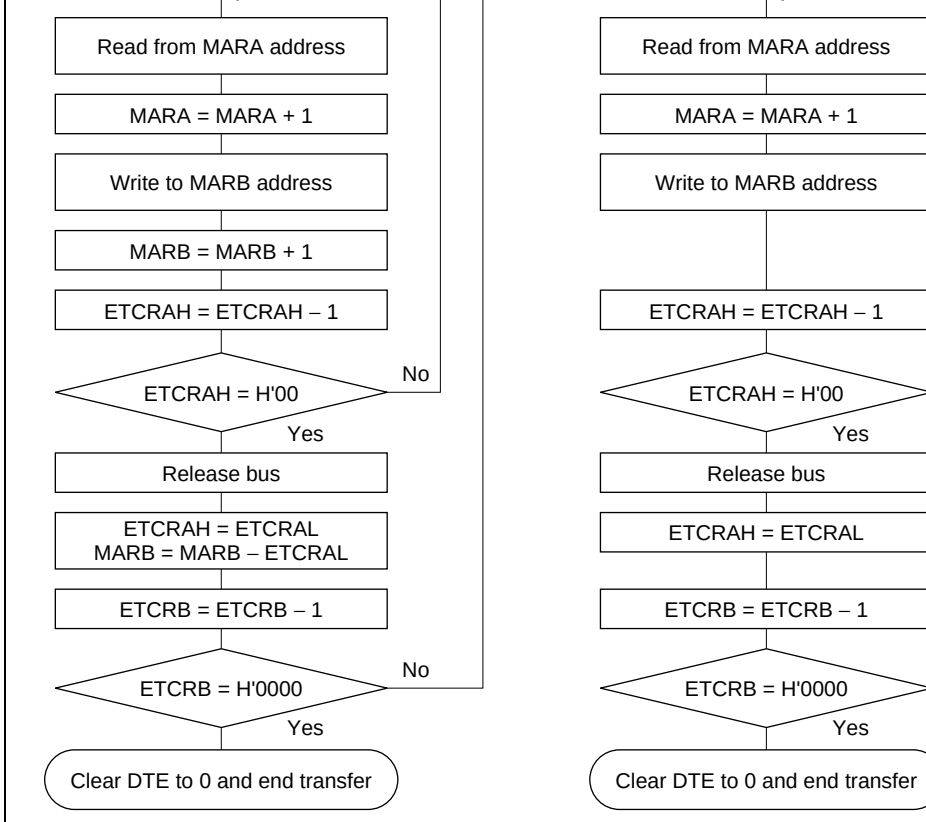
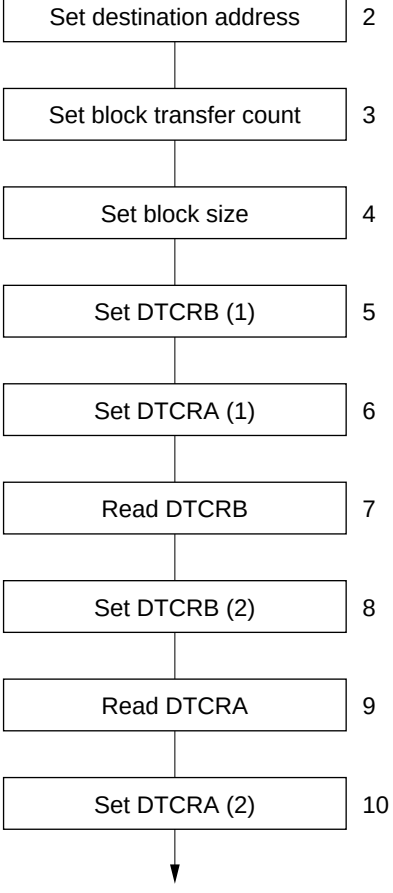


Figure 8.11 Block Transfer Mode Flowcharts (Examples)



- 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 8.12 Block Transfer Mode Setup Procedure (Example)

Activation Source		Channels 0A and 1A	Channels 0B and 1B	Normal	B
Internal interrupts	IMIA0	Yes	Yes	No	Y
	IMIA1	Yes	Yes	No	Y
	IMIA2	Yes	Yes	No	Y
	IMIA3	Yes	Yes	No	Y
	TXI0	Yes	Yes	No	N
	RXI0	Yes	Yes	No	N
External requests	Falling edge of $\overline{\text{DREQ}}$	No	Yes	Yes	Y
	Low input at $\overline{\text{DREQ}}$	No	Yes	Yes	N
Auto-request		No	No	Yes	N

Activation by Internal Interrupts

When an interrupt request is selected as a DMAC activation source and the DTE bit is interrupt request is not sent to the CPU. It is not possible for an interrupt request to ac DMAC and simultaneously generate a CPU interrupt.

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

executed. If level sensing is selected, the transfer continues while $\overline{\text{DREQ}}$ is low, until th
is completed. The bus is released temporarily after each byte or word has been transferr
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, and continues until completed.
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.

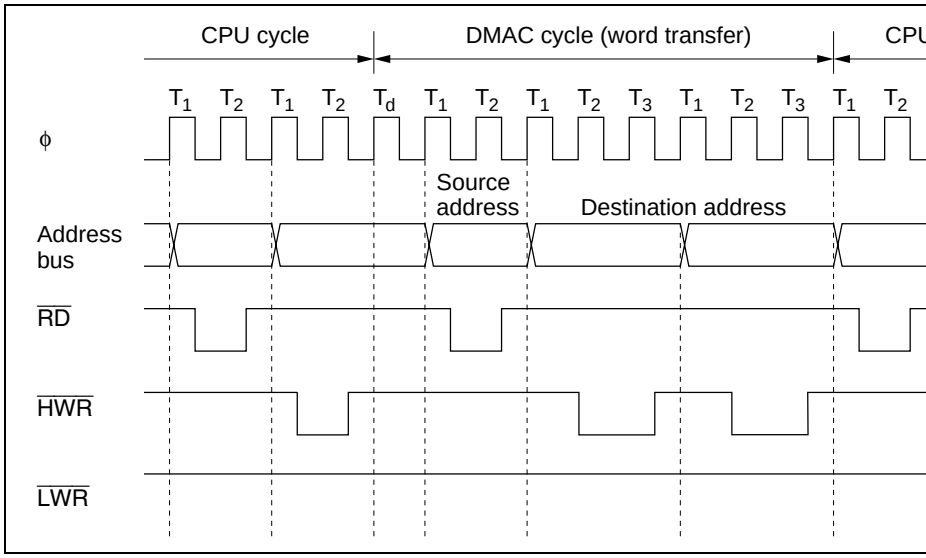


Figure 8.13 DMA Transfer Bus Timing (Example)

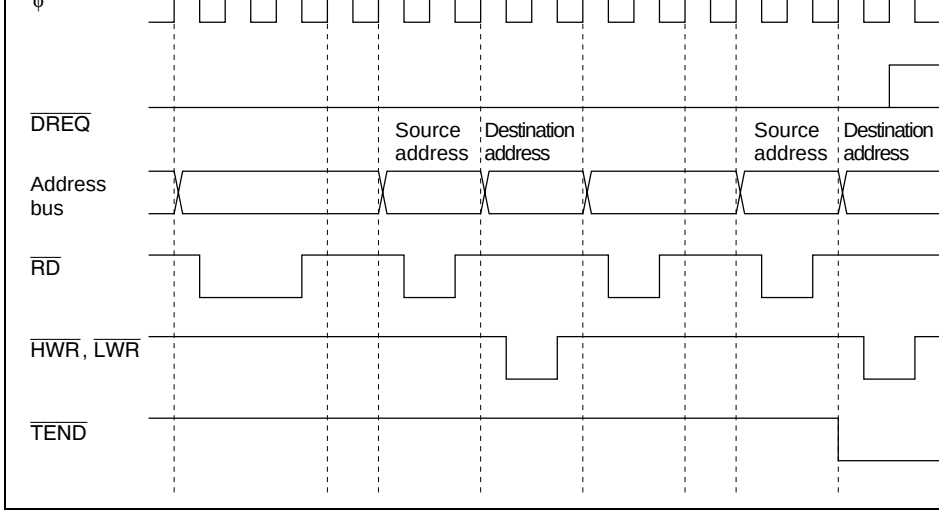


Figure 8.14 Bus Timing of DMA Transfer Requested by Low DREQ Inp

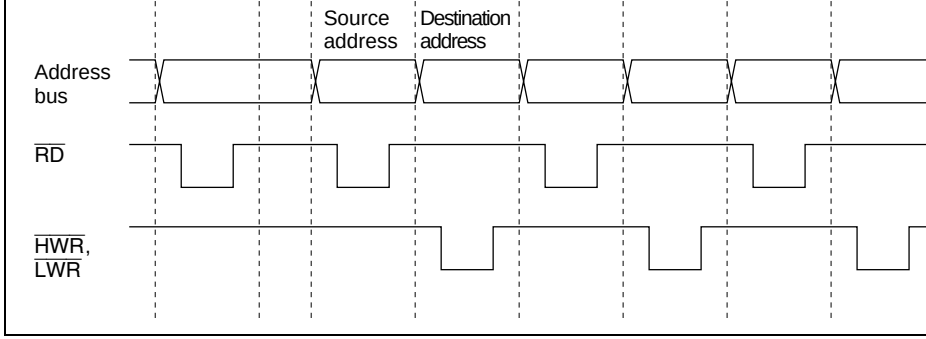


Figure 8.15 Bus Timing of Burst Mode DMA Transfer

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 addre
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.

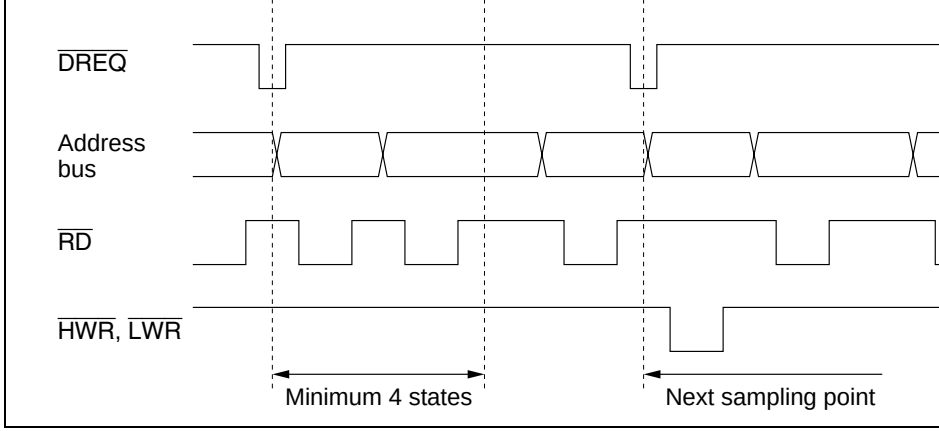


Figure 8.16 Timing of DMAC Activation by Falling Edge of DREQ in Norma

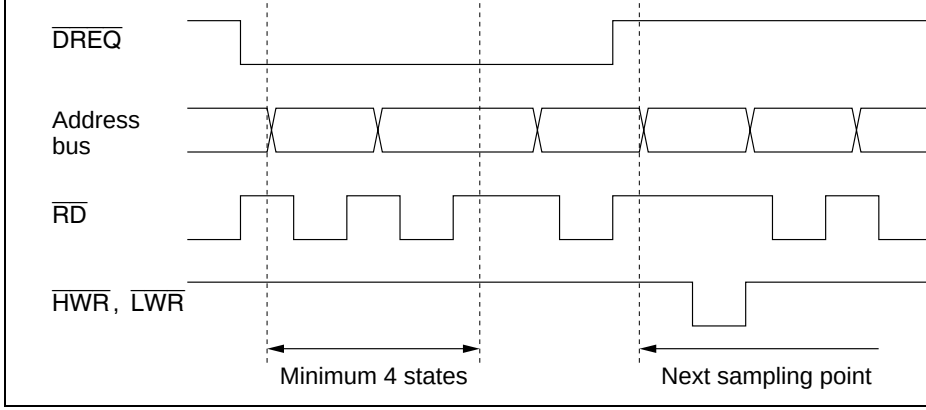


Figure 8.17 Timing of DMAC Activation by Low DREQ Level in Normal

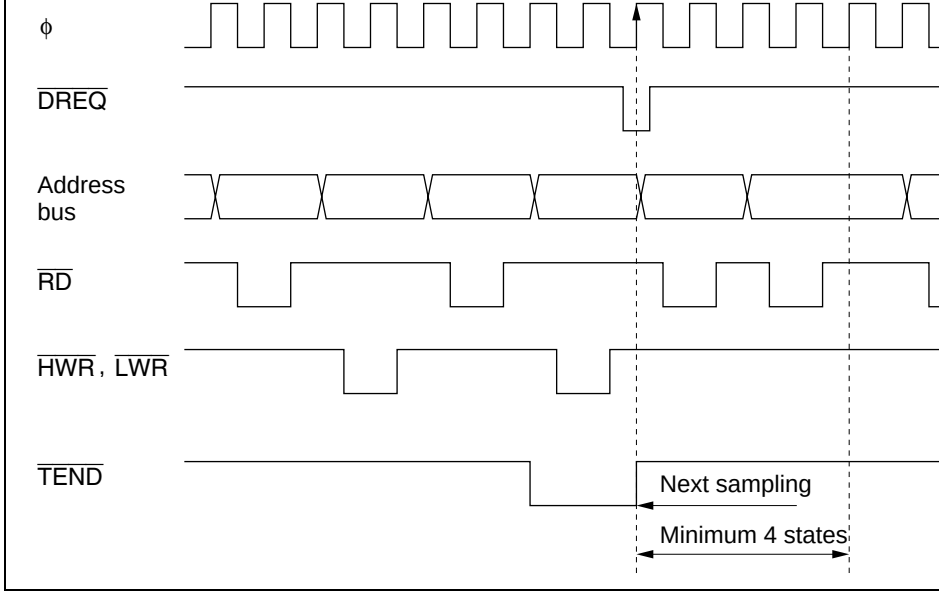


Figure 8.18 Timing of DMAC Activation by Falling Edge of DREQ in Block Tra

Channel 0A	Channel 0	High
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.

1. 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.
2. Once a transfer starts on one channel, requests to other channels are held pending u
channel releases the bus.
3. 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.
4. 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 8.19 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.

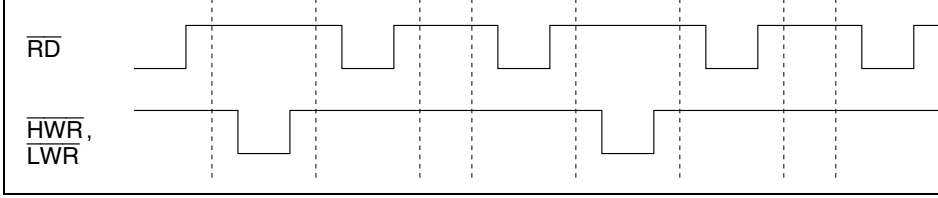


Figure 8.19 Timing of Multiple-Channel Operations

8.4.10 External Bus Requests, Refresh Controller, and DMAC

During a DMA transfer, if the bus right is requested by an external bus request signal (by the refresh controller, the DMAC releases the bus after completing the transfer of th byte or word. If there is a transfer request at this point, the DMAC requests the bus righ Figure 8.20 shows an example of the timing of insertion of a refresh cycle during a burs on channel 0.

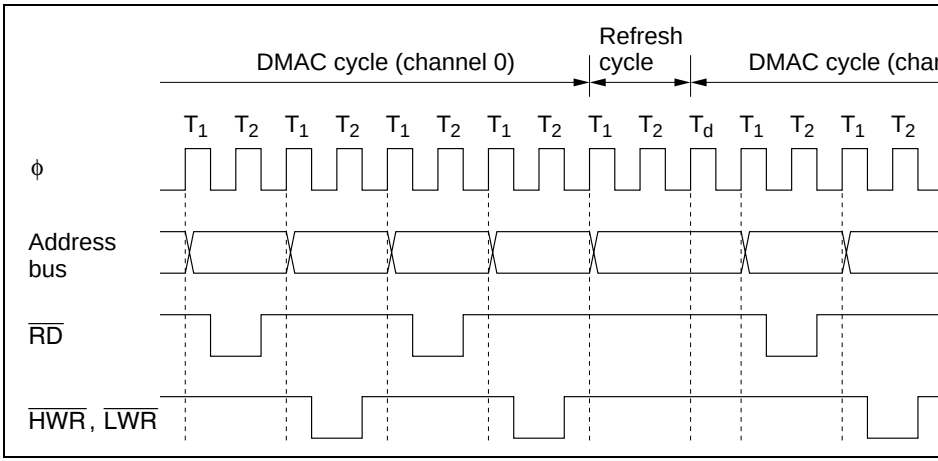


Figure 8.20 Bus Timing of Refresh Controller and DMAC

the DTME bit to 1.

Figure 8.21 shows the procedure for resuming a DMA transfer in normal mode on cha
the transfer was halted by NMI input.

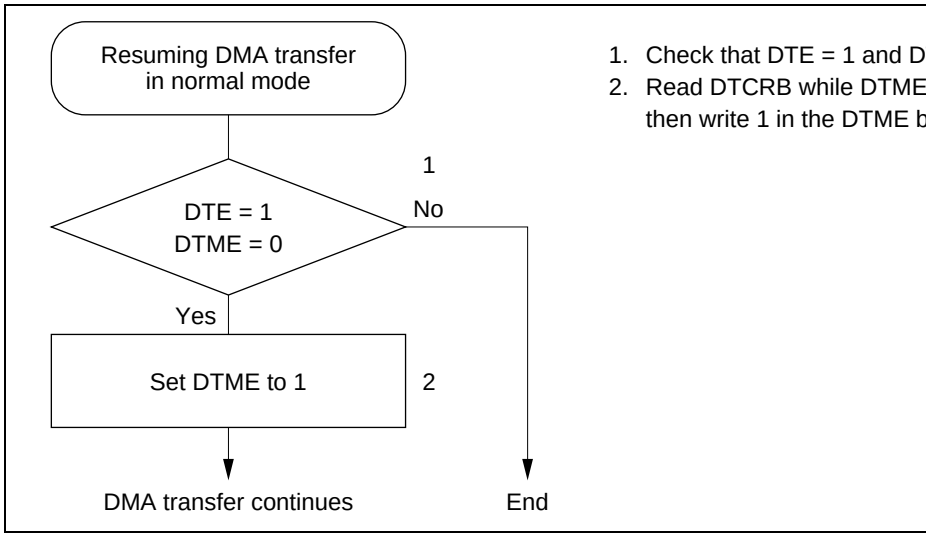
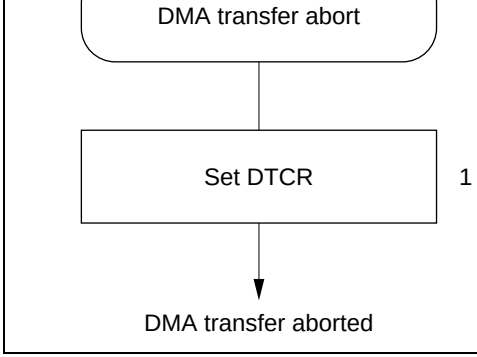


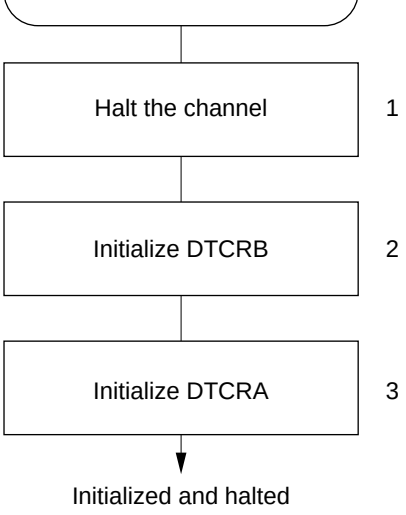
Figure 8.21 Procedure for Resuming a DMA Transfer Halted by NMI (Ex

For information about NMI interrupts in block transfer mode, see section 8.6.6, NMI I
and Block Transfer Mode.



1. Clear the DTE bit to 0 in DTCR.
   To avoid generating an interrupt wh
   aborting a DMA transfer, clear the
   bit to 0 simultaneously.

Figure 8.22 Procedure for Aborting a DMA Transfer



for the transfer to end and the DT
to be cleared to 0.
2. Clear all DTCRB bits to 0.
3. Clear all DTCRA bits to 0.

Figure 8.23 Procedure for Exiting Full Address Mode (Example)

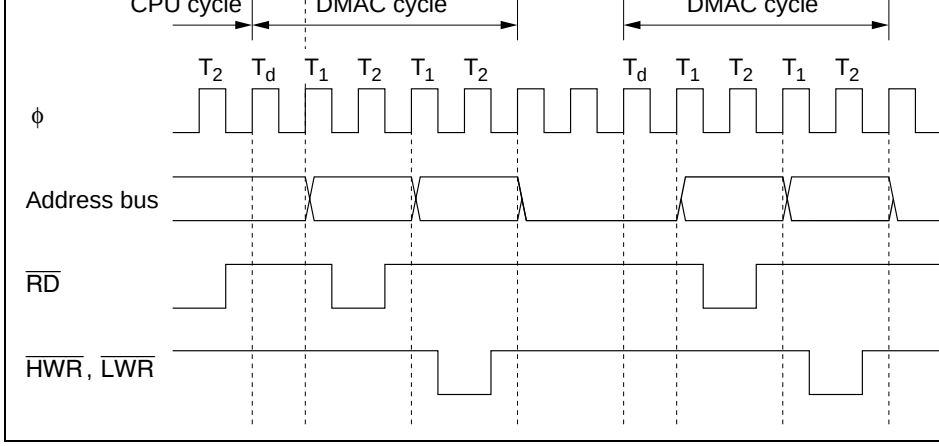


Figure 8.24 Timing of Cycle-Steal Transfer in Sleep Mode

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	—	Lo

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

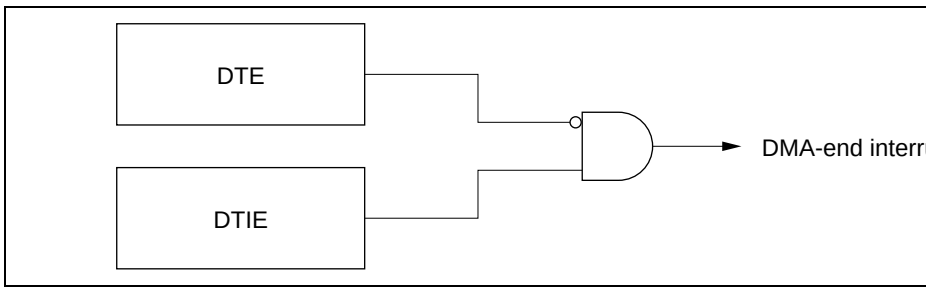


Figure 8.25 DMA-End Interrupt Logic

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.

8.6.3 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.

8.6.4 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.

8.6.5 Note on Activating DMAC by Internal Interrupts

When using an internal interrupt to activate the DMAC, make sure that the interrupt sel
the activating source does not occur during the interval after it has been selected but be
DMAC has been enabled. The on-chip supporting module that will generate the interru
not be activated until the DMAC has been enabled. If the DMAC must be enabled whil
chip supporting module is active, follow the procedure in figure 8.26.

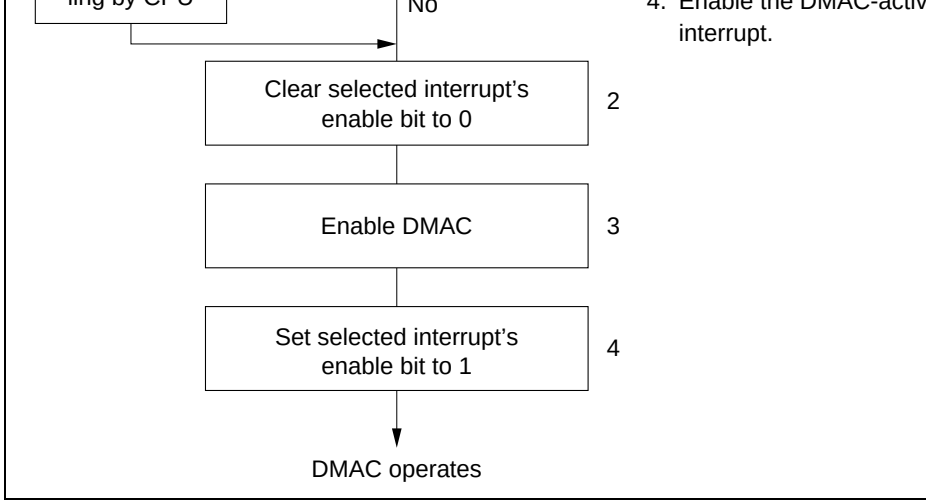


Figure 8.26 Procedure for Enabling DMAC while On-Chip Supporting Mo Operating (Example)

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 8.26 before and after setting the D 1.

When an ITU interrupt activates the DMAC, make sure the next interrupt does not oc the DMA transfer ends. If one ITU interrupt activates two or more channels, make sur interrupt does not occur before the DMA transfers end on all the activated channels. If interrupt occurs before a transfer ends, the channel or channels for which that interrup selected may fail to accept further activation requests.

halted in the middle of a block.

2. If the transfer is halted in the middle of a block, the activating interrupt flag is clear activation request is not held pending.

3. 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 8.6.5, Note on Activating DMAC by Internal Interrupts.

4. 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

8.6.7 Memory and I/O Address Register Values

Table 8.14 indicates the address ranges that can be specified in the memory and I/O add registers (MAR and IOAR).

Table 8.14 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.

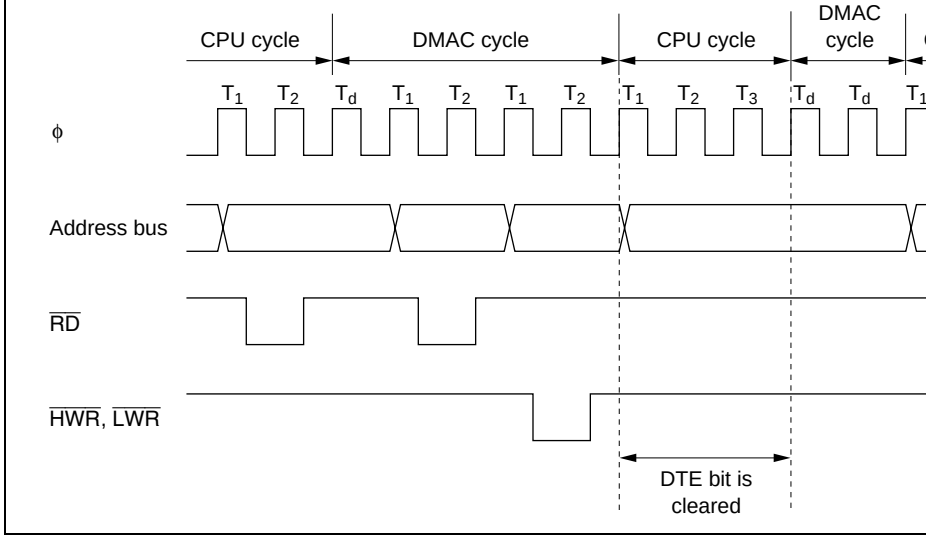


Figure 8.27 Bus Timing at Abort of DMA Transfer in Cycle-Steal Mo

Each port has a data direction register (DDR) for selecting input or output, and a data
(DR) for storing output data. In addition to these registers, ports 2, 4, and 5 have an in
MOS control register (PCR) for switching input pull-up MOS 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, 5, and B can drive LEDs (with 10-mA current sink). Pins $P8_2$ to $P8_0$, P
and PB_3 to PB_0 have Schmitt-trigger input circuits.

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

				output
Port 2	• 8-bit I/O port • Input pull-up MOS • Can drive LEDs	$P2_7$ to $P2_0$/ A_{15} to A_8	Address output pins (A_{15} to A_8)	Address output (A_{15} to A_8) and generic input 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)	
Port 4	• 8-bit I/O port • Input pull-up MOS	$P4_7$ 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	
Port 5	• 4-bit I/O port • Input pull-up MOS • 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
Port 6	• 7-bit I/O port	$P6_6$/$\overline{LWR}$, $P6_5$/$\overline{HWR}$, $P6_4$/$\overline{RD}$, $P6_3$/$\overline{AS}$	Bus control signal output ($\overline{LWR}$, $\overline{HWR}$, $\overline{RD}$, $\overline{AS}$)	
		$P6_2$/$\overline{BACK}$, $P6_1$/$\overline{BREQ}$, $P6_0$/$\overline{WAIT}$	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 D/A converter, and generic input	
		$P7_5$ to $P7_0$/ AN_5 to AN_0	Analog input (AN_5 to AN_0) to A/D converter, and generic inp	

Port 9	• 6-bit I/O port	$P9_5/SCK_1/\overline{IRQ_5}$, $P9_4/SCK_0/\overline{IRQ_4}$, $P9_3/RxD_1$, $P9_2/RxD_0$, $P9_1/TxD_1$, $P9_0/TxD_0$	Input and output (SCK_1, SCK_0, RxD_1, RxD_0, TxD_1, TxD_0) for communication interfaces 1 and 0 (SCI1/0), $\overline{IRQ_5}$ and $\overline{IRQ}$ bit generic input/output			
Port A	• 8-bit I/O port • Schmitt inputs	$PA_7/TP_7/TIOCB_2/A_{20}$	Output (TP_7) from programmable timing pattern controller (TPC), input or output ($TIOCB_2$) for 16-bit integrated timer unit (ITU), and generic input/output	Address output (A_{20})	TPC output (TP_7), ITU input or output ($TIOCB_2$), and generic input/ output	Addre outpu (A_{20})
		$PA_6/TP_6/TIOCA_2/A_{21}/\overline{CS_4}$ $PA_5/TP_5/TIOCB_1/A_{22}/\overline{CS_5}$ $PA_4/TP_4/TIOCA_1/A_{23}/\overline{CS_6}$	TPC output (TP_6 to TP_4), ITU input and output ($TIOCA_2$, $TIOCB_1$, $TIOCA_1$), $\overline{CS_4}$ to $\overline{CS_6}$ output, and generic input/ output	TPC output (TP_6 to TP_4), ITU input and output ($TIOCA_2$, $TIOCB_1$, $TIOCA_1$), address output (A_{23} to A_{21}), $\overline{CS_4}$ to $\overline{CS_6}$ output, and generic input/output	TPC output (TP_6 to TP_4), ITU input and output ($TIOCA_2$, $TIOCB_1$, $TIOCA_1$), $\overline{CS_4}$ to $\overline{CS_6}$ output, and generic input/ output	TPC outpu (TP_6 t TP_4), input outpu (TIOC TIOC TIOC addre outpu (A_{23} to A_{21}), to $\overline{CS}$ outpu and gener input/ put

		$\overline{TEND_0}$/TCLKA	
Port B	• 8-bit I/O port • Can drive LEDs • PB_3 to PB_0 have Schmitt inputs	PB_7/TP_{15}/ $\overline{DREQ_1}$/ADTRG	TPC output (TP_{15}), DMAC input ($\overline{DREQ_1}$), trigger input ($\overline{ADT}$ converter, and generic input/output
		PB_6/TP_{14}/ $\overline{DREQ_0}$,/$\overline{CS_7}$	TPC output (TP_{14}), DMAC input ($\overline{DREQ_0}$), $\overline{CS_7}$ output, and generic input/output
		PB_5/TP_{13}/ $TOCXB_4$, PB_4/TP_{12}/ $TOCXA_4$, PB_3/TP_{11}/ $TIOCB_4$, PB_2/TP_{10}/ $TIOCA_4$, PB_1/TP_9/ $TIOCB_3$, PB_0/TP_8/ $TIOCA_3$	TPC output (TP_{13} to TP_8), ITU input and output ($TOCXB_4$, T $TIOCB_4$, $TIOCA_4$, $TIOCB_3$, $TIOCA_3$), and generic input/outpu

In modes 5 and 6 (expanded modes with on-chip ROM enabled), settings in the port 1 direction register (P1DDR) can designate pins for address bus output (A_7 to A_0) or gen In mode 7 (single-chip mode), port 1 is a generic input/output port.

When DRAM is connected to area 3, A_7 to A_0 output row and column addresses in rea cycles. For details see section 7, Refresh Controller.

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

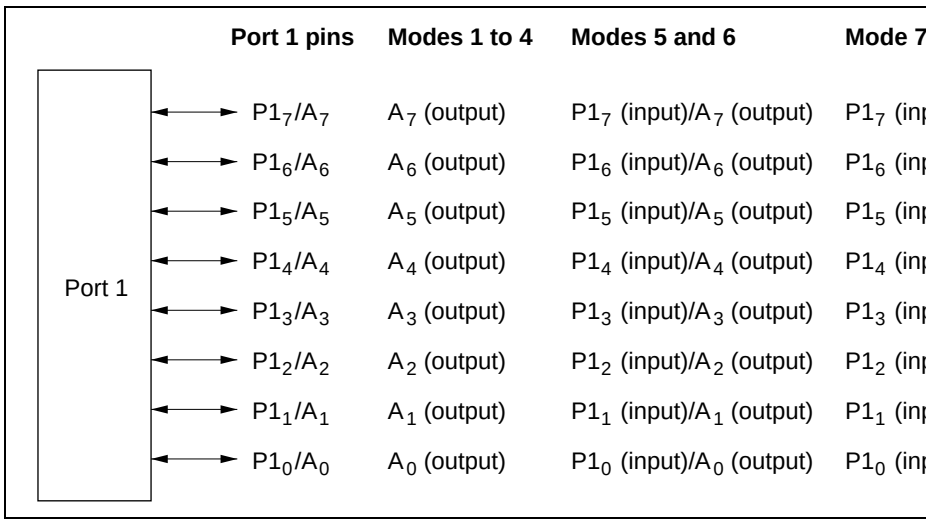


Figure 9.1 Port 1 Pin Configuration

	register				
H'FFC2	Port 1 data register	P1DR	R/W	H'00	H'00

Note: * Lower 16 bits of the address.

Port 1 Data Direction Register (P1DDR)

P1DDR is an 8-bit write-only register that can select input or output for each pin in por

Bit		7	6	5	4	3	2	1
		$P1_7DDR$	$P1_6DDR$	$P1_5DDR$	$P1_4DDR$	$P1_3DDR$	$P1_2DDR$	$P1_1DDF$
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 and cannot be modified. Port 1 functions as an address bus.

Modes 5 and 6 (Expanded Modes with On-Chip ROM Enabled): A pin in port 1 be address output pin if the corresponding P1DDR bit is set to 1, and a generic input pin if cleared to 0.

Mode 7 (Single-Chip Mode): Port 1 functions as an input/output port. A pin in port 1 output pin if the corresponding P1DDR bit is set to 1, and an input pin if this bit is clea

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

is read the value of the corresponding P1DR bit is returned. When a bit in P1DDR is c
if port 1 is read the corresponding pin level is 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 stan
retains its previous setting.

output pins (A_{15} to A_8). In modes 5 and 6 (expanded modes with on-chip ROM enabled) in the port 2 data direction register (P2DDR) can designate pins for address bus output or generic input. In mode 7 (single-chip mode), port 2 is a generic input/output port.

When DRAM is connected to area 3, A_9 and A_8 output row and column addresses in rea cycles. For details see section 7, Refresh Controller.

Port 2 has software-programmable built-in pull-up MOS. Pins in port 2 can drive one T and a 90-pF capacitive load. They can also drive a darlington transistor pair.

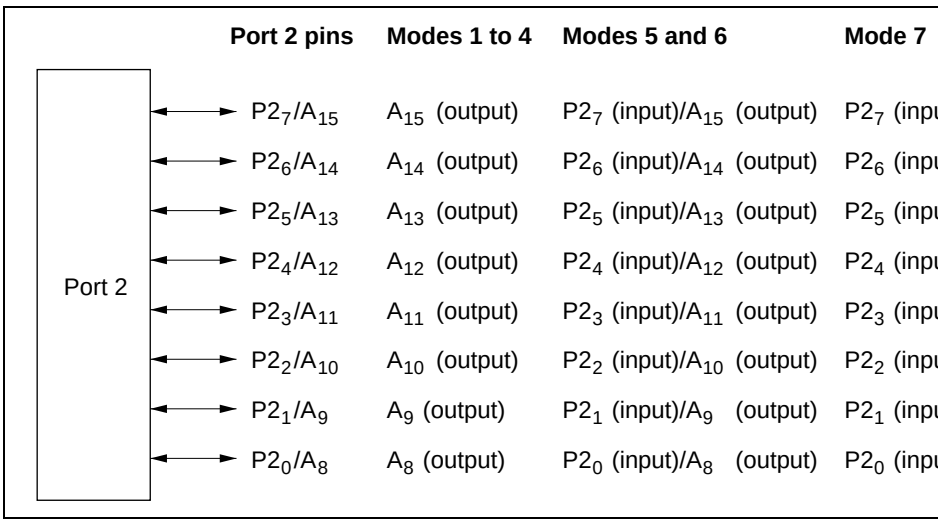


Figure 9.2 Port 2 Pin Configuration

	register				
H'FFC3	Port 2 data register	P2DR	R/W	H'00	H'0
H'FFD8	Port 2 input pull-up MOS control register	P2PCR	R/W	H'00	H'0

Note: * Lower 16 bits of the address.

Port 2 Data Direction Register (P2DDR)

P2DDR is an 8-bit write-only register that can select input or output for each pin in po

Bit		7	6	5	4	3	2	1
		$P2_7DDR$	$P2_6DDR$	$P2_5DDR$	$P2_4DDR$	$P2_3DDR$	$P2_2DDR$	$P2_1DD$
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 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 and cannot be modified. Port 2 functions as an address bus.

Modes 5 and 6 (Expanded Modes with On-Chip ROM Enabled): Following a rese an input port. A pin in port 2 becomes an address output pin if the corresponding P2D to 1, and a generic input port if this bit is cleared to 0.

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

In modes 1 to 4, P2DDR always returns 1 when read. No value can be written to.

P2DR is an 8-bit readable/writable register that stores output data for pins $P2_7$ to $P2_0$. W
acts as an output port, the value of this register is output. When a bit in P2DDR is set to
is read the value of the corresponding P2DR bit is returned. When a bit in P2DDR is cl
if port 2 is read the corresponding pin level is 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 stand
retains its previous setting.

Port 2 Input Pull-Up MOS Control Register (P2PCR)

P2PCR is an 8-bit readable/writable register that controls the MOS input pull-up transis
2.

Bit	7	6	5	4	3	2	1
	$P2_7PCR$	$P2_6PCR$	$P2_5PCR$	$P2_4PCR$	$P2_3PCR$	$P2_2PCR$	$P2_1PCR$
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 2 input pull-up MOS control 7 to 0
These bits control input pull-up
transistors built into port 2

Mode	Reset	Hardware Standby Mode	Software Standby Mode	Other M
1	Off	Off	Off	Off
2				
3				
4				
5	Off	Off	On/off	On/off
6				
7				

Legend:

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

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

Pins in port 3 can drive one TTL load and a 90-pF capacitive load. They can also drive
darlington transistor pair.

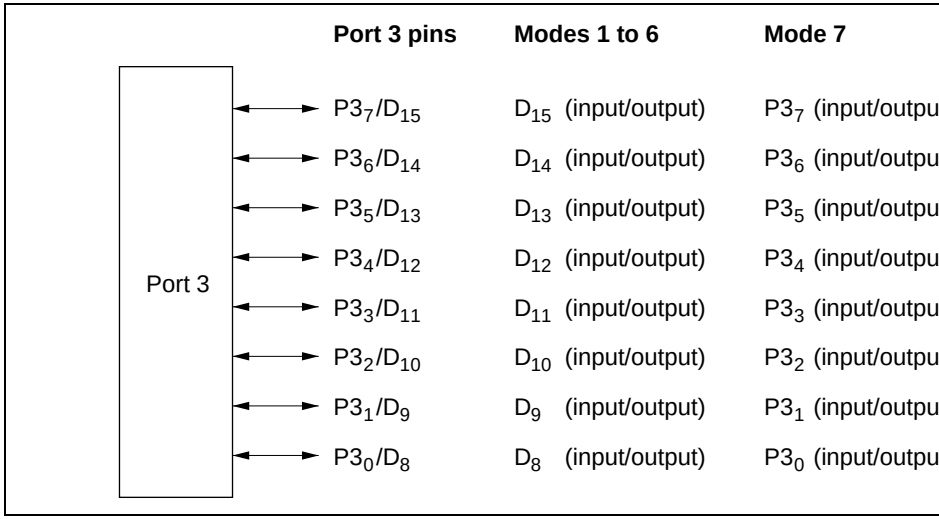


Figure 9.3 Port 3 Pin Configuration

9.4.2 Register Descriptions

Table 9.5 summarizes the registers of port 3.

Table 9.5 Port 3 Registers

Address*	Name	Abbreviation	R/W	Initial V
H'FFC4	Port 3 data direction register	P3DDR	W	H'00
H'FFC6	Port 3 data register	P3DR	R/W	H'00

Note: * Lower 16 bits of the address.

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

Modes 1 to 6 (Expanded Modes): Port 3 functions as a data bus. P3DDR is ignored.

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

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 sta it retains its previous setting. If a P3DDR bit is set to 1, the corresponding pin maintai state in software standby mode.

Port 3 Data Register (P3DR)

P3DR is an 8-bit readable/writable register that stores output data for pins $P3_7$ to $P3_0$. V acts as an output port, the value of this register is output. When a bit in P3DDR is set t is read the value of the corresponding P3DR bit is returned. When a bit in P3DDR is c if port 3 is read the corresponding pin 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 stan retains its previous setting.

areas 0 to 7 all as 8-bit-access areas, the chip operates in 8-bit bus mode and port 4 is a input/output port. When at least one of areas 0 to 7 is designated as a 16-bit-access area operates in 16-bit bus mode and port 4 becomes part of the data bus. In mode 7 (single- mode), port 4 is a generic input/output port.

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

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

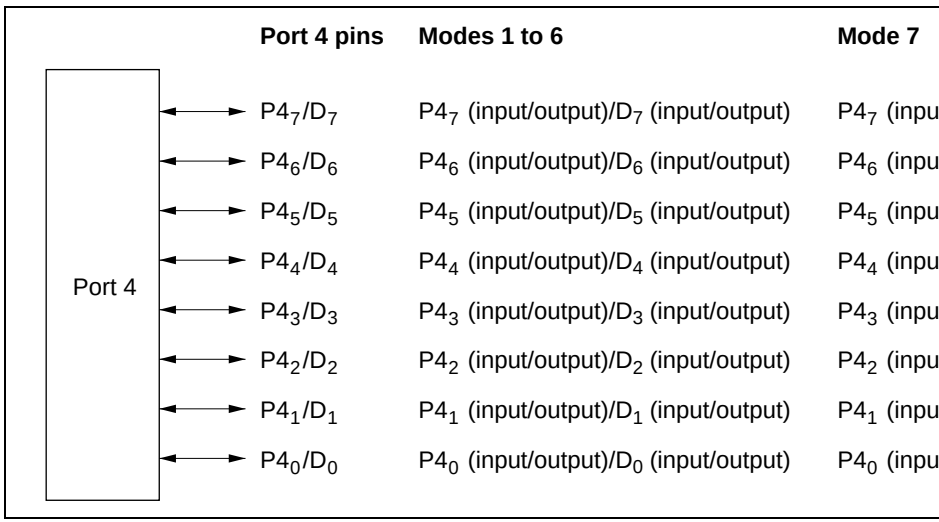


Figure 9.4 Port 4 Pin Configuration

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

Note: * Lower 16 bits of the address.

Port 4 Data Direction Register (P4DDR)

P4DDR is an 8-bit write-only register that can select input or output for each pin in po

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 pi

Modes 1 to 6 (Expanded Modes): When all areas are designated as 8-bit-access area bus width control register (ABWCR) of the bus controller, selecting 8-bit bus mode, p functions as a generic input/output port. A pin in port 4 becomes an output port if the corresponding P4DDR bit is set to 1, and an input port if this bit is cleared to 0.

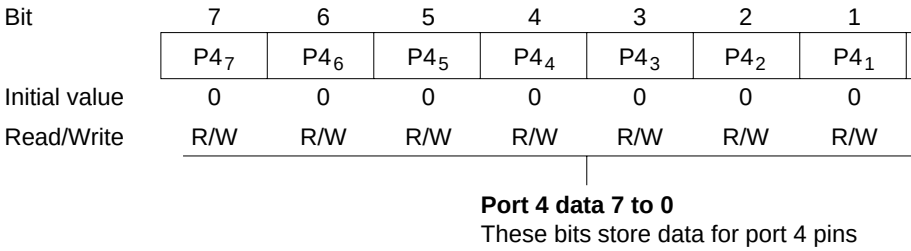
When at least one area is designated as a 16-bit-access area, selecting 16-bit bus mode functions as part of the data bus regardless of the value in P4DDR.

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

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 sta it retains its previous setting.

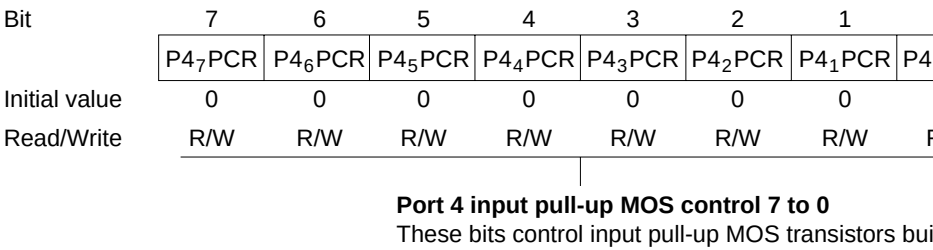
is read the value of the corresponding P4DR bit is returned. When a bit in P4DDR is cl
if port 4 is read the corresponding pin level is read.



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

Port 4 Input Pull-Up MOS Control Register (P4PCR)

P4PCR is an 8-bit readable/writable register that controls the MOS input pull-up transis
4.



In mode 7 (single-chip mode), and in 8-bit bus mode in modes 1 to 6 (expanded modes
P4DDR bit is cleared to 0 (selecting generic input), if the corresponding P4PCR bit is s
input pull-up MOS transistor is turned on.

1 to 6	8-bit bus mode	Off	Off	On/off	On/
	16-bit bus mode			Off	Off
7				On/off	On/

Legend:

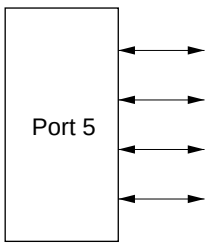
Off: The input pull-up MOS transistor is always off.

On/off: The input pull-up MOS transistor is on if P4PCR = 1 and P4DDR = 0. Otherwis

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

Port 5 has software-programmable built-in pull-up MOS 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	Modes 5 and 6	Mode 7
$P5_3/A_{19}$	A_{19} (output)	$P5_3$ (input)/A_{19} (output)	$P5_3$ (inp
$P5_2/A_{18}$	A_{18} (output)	$P5_2$ (input)/A_{18} (output)	$P5_2$ (inp
$P5_1/A_{17}$	A_{17} (output)	$P5_1$ (input)/A_{17} (output)	$P5_1$ (inp
$P5_0/A_{16}$	A_{16} (output)	$P5_0$ (input)/A_{16} (output)	$P5_0$ (inp

Figure 9.5 Port 5 Pin Configuration

	register				
H'FFCA	Port 5 data register	P5DR	R/W	H'F0	H'F
H'FFDB	Port 5 input pull-up MOS control register	P5PCR	R/W	H'F0	H'F

Note: * Lower 16 bits of the address.

Port 5 Data Direction Register (P5DDR)

P5DDR is an 8-bit write-only register that can select input or output for each pin in po

Bits 7 to 4 are reserved. They cannot be modified and are always read as 1.

Bit		7	6	5	4	3	2	1
		—	—	—	—	$P5_3DDR$	$P5_2DDR$	$P5_1DD$
Modes 1 to 4	Initial value	1	1	1	1	1	1	1
	Read/Write	—	—	—	—	—	—	—
Modes 5 to 7	Initial value	1	1	1	1	0	0	0
	Read/Write	—	—	—	—	W	W	W

Reserved bits

Port 5 data directio
These bits select inp
output for port 5 pins

Modes 1 to 4 (Expanded Modes with On-Chip ROM Disabled): P5DDR values are and cannot be modified. Port 5 functions as an address bus.

Modes 5 and 6 (Expanded Modes with On-Chip ROM Enabled): Following a rese an input port. A pin in port 5 becomes an address output pin if the corresponding P5D to 1, and an input port if this bit is cleared to 0.

standby mode. In software standby mode it retains its previous setting, so if a P5DDR b
1 while port 5 acts as an I/O port, the corresponding pin maintains its output state in so
standby mode.

Port 5 Data Register (P5DR)

P5DR is an 8-bit readable/writable register that stores output data for pins $P5_3$ to $P5_0$. W
acts as an output port, the value of this register is output. When a bit in P5DDR is set to
is read the value of the corresponding P5DR bit is returned. When a bit in P5DDR is cl
if port 5 is read the corresponding pin level is read.

Bits 7 to 4 are reserved. They cannot be modified and are always read as 1.

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	

Reserved bits

Port 5 data 3 to 0
These bits store da
for port 5 pins

P5DR is initialized to H'F0 by a reset and in hardware standby mode. In software stand
retains its previous setting.

Initial value	1	1	1	1	0	0	0
Read/Write	—	—	—	—	R/W	R/W	R/W

Reserved bits

Port 5 input pull-up MOS co
These bits control input pull-u
transistors built into port 5

In modes 5 to 7, when a P5DDR bit is cleared to 0 (selecting generic input), if the cor
bit from $P5_3$PCR to $P5_0$PCR is set to 1, the input pull-up MOS 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 9.9 summarizes the states of the input pull-ups MOS in each mode.

Table 9.9 Input Pull-Up MOS Transistor States (Port 5)

Mode	Reset	Hardware Standby Mode	Software Standby Mode	Other M
1	Off	Off	Off	Off
2				
3				
4				
5	Off	Off	On/off	On/off
6				
7				

Legend:
Off: The input pull-up MOS transistor is always off.
On/off: The input pull-up MOS transistor is on if P5PCR = 1 and P5DDR = 0. Otherwis

Figure 9.6 shows the pin configuration of port 6. In modes 1 to 6 (expanded modes) the functions are $\overline{LWR}$, $\overline{HWR}$, $\overline{RD}$, $\overline{AS}$, $P6_2/\overline{BACK}$, $P6_1/\overline{BREQ}$, and $P6_0/\overline{WAIT}$. See table 9 method of selecting the pin states. In mode 7 (single-chip mode) port 6 is a generic inpu port.

Pins in port 6 can drive one TTL load and a 30-pF capacitive load. They can also drive darlington transistor pair.

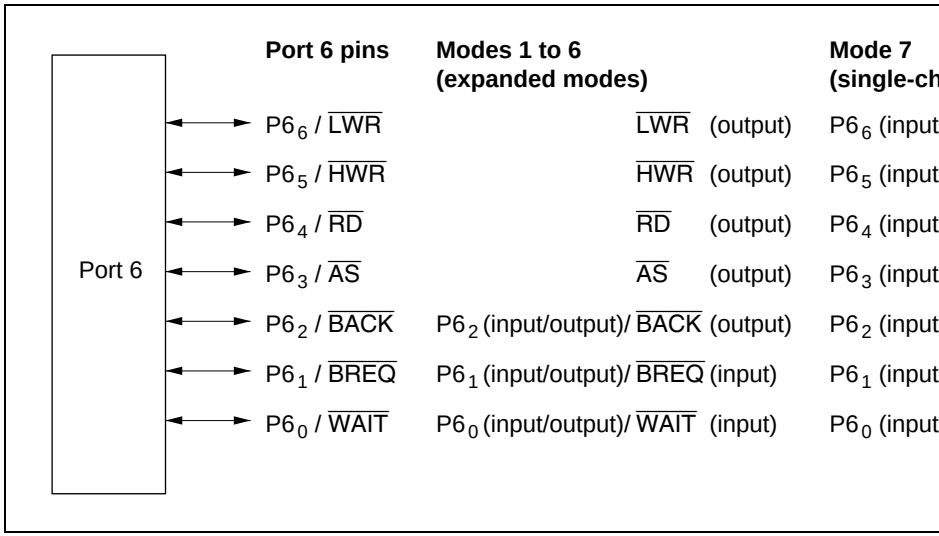


Figure 9.6 Port 6 Pin Configuration

	register				
H'FFCB	Port 6 data register	P6DR	R/W	H'80	H'8

Note: * Lower 16 bits of the address.

Port 6 Data Direction Register (P6DDR)

P6DDR is an 8-bit write-only register that can select input or output for each pin in po reserved. It cannot be modified and is always read as 1.

Bit	7	6	5	4	3	2	1
	—	$P6_6DDR$	$P6_5DDR$	$P6_4DDR$	$P6_3DDR$	$P6_2DDR$	$P6_1DDR$
Initial value	1	0	0	0	0	0	0
Read/Write	—	W	W	W	W	W	W

Reserved bit

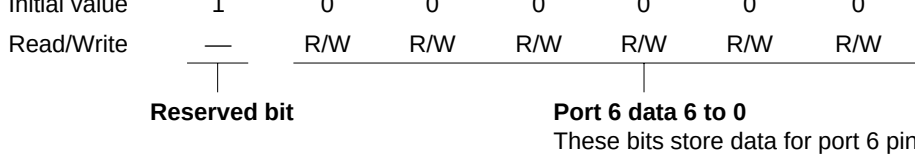
Port 6 data direction 6 to 0
These bits select input or output for po

Modes 1 to 6 (Expanded Modes): Ports $P6_6$ to $P6_3$ function as bus control output pins $\overline{HWR}$, $\overline{RD}$, $\overline{AS}$), regardless of the settings of $P6_6DDR$ to $P6_3DDR$. Ports $P6_2$ to $P6_0$ fun bus control pins ($\overline{BACK}$, $\overline{BREQ}$, $\overline{WAIT}$) or I/O ports. For selecting the pin function, s 9.11. When ports $P6_2$ to $P6_0$ function as I/O ports and if P6DDR is set to 1, the corresp of port 6 functions as an output port. If P6DDR is cleared to 0, the corresponding pin f an input port.

Mode 7 (Single-Chip Mode): Port 6 is a generic input/output port. A pin in port 6 be output port if the corresponding P6DDR bit is set to 1, and an input port if this bit is cl

P6DDR is a write-only register. Its value cannot be read. All bits return 1 when read.

P6DDR is initialized to H'80 by a reset and in hardware standby mode. In software sta it retains its previous setting. If a P6DDR bit is set to 1 while port 6 acts as an I/O por corresponding pin maintains its output state in software standby mode.



Bit 7 is reserved, cannot be modified, and always read as 1.

P6DR is initialized to H'80 by a reset and in hardware standby mode. In software stand retains its previous setting.

Table 9.11 Port 6 Pin Functions in Modes 1 to 6

Pin	Pin Functions and Selection Method		
$P6_6$/$\overline{LWR}$	Functions as follows regardless of $P6_6$DDR		
	$P6_6$DDR	0	1
	Pin function	$\overline{LWR}$ output	
$P6_5$/$\overline{HWR}$	Functions as follows regardless of $P6_5$DDR		
	$P6_5$DDR	0	1
	Pin function	$\overline{HWR}$ output	
$P6_4$/$\overline{RD}$	Functions as follows regardless of $P6_4$DDR		
	$P6_4$DDR	0	1
	Pin function	$\overline{RD}$ output	

$P6_2$DDR	0	1	
Pin function	$P6_2$ input	$P6_2$ output	$\overline{BA}$

$P6_1/\overline{BREQ}$

Bit BRLE in BRCR and bit $P6_1$DDR select the pin function as follow

BRLE	0		
$P6_1$DDR	0	1	
Pin function	$P6_1$ input	$P6_1$ output	$\overline{BR}$

$P6_0/\overline{WAIT}$

Bits WCE7 to WCE0 in WCER, bit WMS1 in WCR, and bit $P6_0$DDR pin function as follows

WCER	All 1s			
WMS1	0		1	
$P6_0$DDR	0	1	0*	
Pin function	$P6_0$ input	$P6_0$ output	$\overline{WAIT}$ in	

Note: * Do not set bit $P6_0$DDR to 1.

For the analog input pins of the A/D converter, see section 15, A/D Converter.

For the analog input pins of the D/A converter, see section 16, D/A Converter.

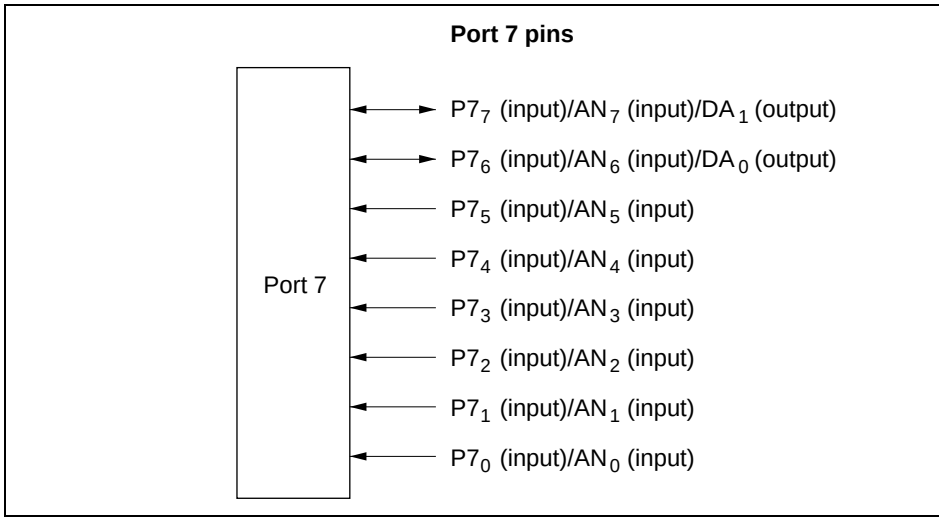


Figure 9.7 Port 7 Pin Configuration

Note: * Lower 16 bits of the address.

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 P7DR is read, the logic level of the pin is always read. No data can be written t

to $\overline{IRQ}_0$ input. See table 9.14 for the selection of pin functions in expanded modes.

In mode 7 (single-chip mode), port 8 can provide $\overline{IRQ}_3$ to $\overline{IRQ}_0$ input. See table 9.15 for selection of pin functions in single-chip mode.

The $\overline{IRQ}_3$ to $\overline{IRQ}_0$ functions are selected by IER settings, regardless of whether the pin is input or output. For details see section 5, Interrupt Controller.

Pins in port 8 can drive one TTL load and a 90-pF capacitive load. They can also drive a darlington transistor pair.

Pins $P8_2$ to $P8_0$ have Schmitt-trigger inputs.

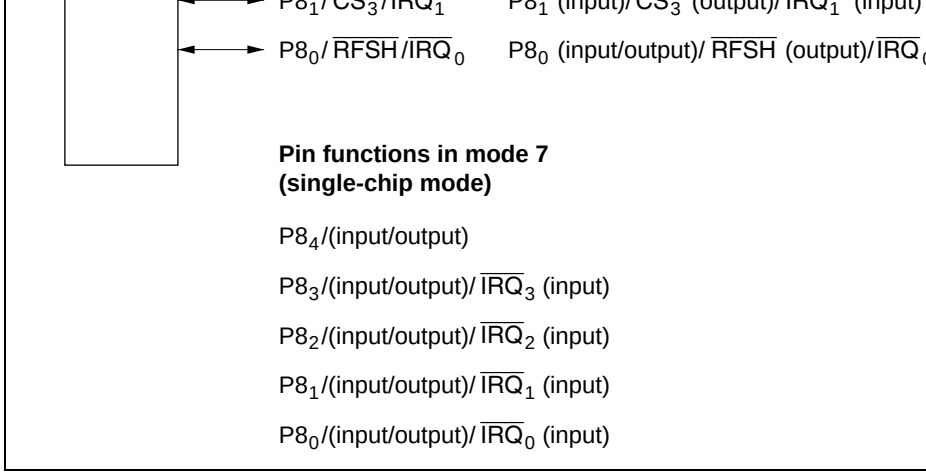


Figure 9.8 Port 8 Pin Configuration

9.9.2 Register Descriptions

Table 9.13 summarizes the registers of port 8.

Table 9.13 Port 8 Registers

				Initial Va	
Address*	**Name**	**Abbreviation**	**R/W**	**Modes 1 to 4**	**Mo**
H'FFCD	Port 8 data direction register	P8DDR	W	H'F0	H'E
H'FFCF	Port 8 data register	P8DR	R/W	H'E0	H'E

Note: * Lower 16 bits of the address.

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 0
These bits select input or output for port 8 pins

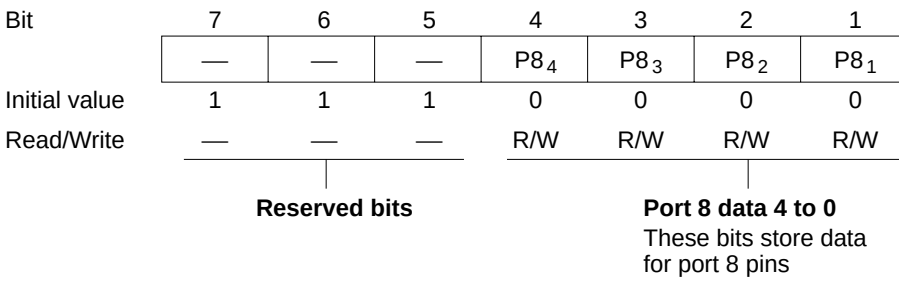
Modes 1 to 6 (Expanded Modes): When bits in P8DDR bit are set to 1, $P8_4$ to $P8_1$ bec $\overline{CS}_3$ output pins. When bits in P8DDR are cleared to 0, the corresponding pins become In modes 1 to 4 (expanded modes with on-chip ROM disabled), following a reset only output. The other three pins are input ports. In modes 5 and 6 (expanded modes with or ROM enabled), following a reset all four pins are input ports.

When the refresh controller is enabled, $P8_0$ is used unconditionally for $\overline{RFSH}$ output. W refresh controller is disabled, $P8_0$ becomes a generic input/output port according to the setting. For details see table 9.15.

Mode 7 (Single-Chip Mode): Port 8 is a generic input/output port. A pin in port 8 bec output port if the corresponding P8DDR bit is set to 1, and an input port if this bit is cle

P8DDR is a write-only register. Its value cannot be read. All bits return 1 when read.

P8DDR is initialized to H'F0 in modes 1 to 4 and H'E0 in modes 5 to 7 by a reset and i standby mode. In software standby mode it retains its previous setting, so if a P8DDR b 1 while port 8 acts as an I/O port, the corresponding pin maintains its output state in so standby mode.



P8DR is initialized to H'E0 by a reset and in hardware standby mode. In software stan
retains its previous setting.

$P8_3$DDR	0	1
Pin function	$P8_3$ input	$\overline{CS}_1$ outp
	$\overline{IRQ}_3$ input	

$P8_2/\overline{CS}_2/\overline{IRQ}_2$

Bit $P8_2$DDR selects the pin function as follows

$P8_2$DDR	0	1
Pin function	$P8_2$ input	$\overline{CS}_2$ outp
	$\overline{IRQ}_2$ input	

$P8_1/\overline{CS}_3/\overline{IRQ}_1$

Bit $P8_1$DDR selects the pin function as follows

$P8_1$DDR	0	1
Pin function	$P8_1$ input	$\overline{CS}_3$ outp
	$\overline{IRQ}_1$ input	

$P8_0/\overline{RFSH}/\overline{IRQ}_0$

Bit RFSHE in RFSHCR and bit $P8_0$DDR select the pin function as fo

RFSHE	0		
$P8_0$DDR	0	1	
Pin function	$P8_0$ input	$P8_0$ output	$\overline{RFS}$
	$\overline{IRQ}_0$ input		

$P8_3$DDR	0	1
Pin function	$P8_3$ input	$P8_3$ out
	$\overline{IRQ}_3$ 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	$P8_2$ out
	$\overline{IRQ}_2$ 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	$P8_1$ out
	$\overline{IRQ}_1$ 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	$P8_0$ out
	$\overline{IRQ}_0$ input	

The $\overline{IRQ}_5$ and $\overline{IRQ}_4$ functions are selected by IER settings, regardless of whether the pi
input or output. For details see section 5, Interrupt Controller.

Port 9 has the same set of pin functions in all operating modes. Figure 9.9 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.

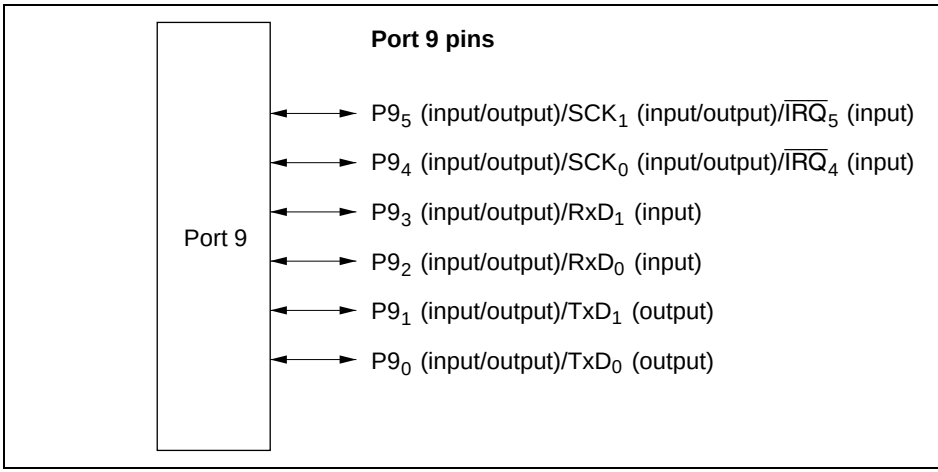


Figure 9.9 Port 9 Pin Configuration

Note: * Lower 16 bits of the address.

Port 9 Data Direction Register (P9DDR)

P9DDR is an 8-bit write-only register that can select input or output for each pin in po

Bits 7 and 6 are reserved. They cannot be modified and are always read as 1.

Bit	7	6	5	4	3	2	1
	—	—	$P9_5DDR$	$P9_4DDR$	$P9_3DDR$	$P9_2DDR$	$P9_1DDR$
Initial value	1	1	0	0	0	0	0
Read/Write	—	—	W	W	W	W	W

Reserved bits

Port 9 data direction 5 to 0
These bits select input or output for port 9 pins

While port 9 acts as an I/O port, a pin in port 9 becomes an output port if the correspo P9DDR bit is set to 1, and an input port if this bit is cleared to 0. For selecting the pin table 9.17.

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. If a P9DDR bit is set to 1 while port 9 acts as an I/O por corresponding pin maintains its output state in software standby mode.

	—	—	$P9_5$	$P9_4$	$P9_3$	$P9_2$	$P9_1$
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

Bits 7 and 6 are reserved. They cannot be modified and are always read as 1.

P9DR is initialized to H'C0 by a reset and in hardware standby mode. In software stand
retains its previous setting.

$P9_5$DDR	0	1	—	—
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 of SCI0, and select the 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 and bit $P9_3$DDR select the pin function as fo

RE	0		
$P9_3$DDR	0	1	
Pin function	$P9_3$ input	$P9_3$ output	Rx

$P9_2/RxD_0$

Bit RE in SCR of SCI0, bit SMIF in SCMR, and bit $P9_2$DDR select t function as follows

SMIF	0			
RE	0		1	
$P9_2$DDR	0	1	—	
Pin function	$P9_2$ input	$P9_2$ output	RxD_0 input	

RENESAS

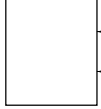
function as follows

SMIF	0			
TE	0		1	
$P9_0$DDR	0	1	—	
Pin function	$P9_0$ input	$P9_0$ output	TxD_0 output	Tx

Note: * Functions as the TxD_0 output pin, but there are two state which the pin is driven, and another in which the pin is a impedance.

(TEND$_1$, TEND$_0$) from the DMA controller (DMAC), $\overline{CS}_4$ to $\overline{CS}_6$ output, and address
to A_{20}). A reset or hardware standby leaves port A as an input port, except that in mod
6, one pin is always used for A_{20} output. For selecting the pin function, see table 9.19.
pins for TPC, ITU, and DMAC input and output is described in the sections on those
output of address bits A_{23} to A_{21} in modes 3, 4, and 6, see section 6.2.5, Bus Release C
Register (BRCR). For output of $\overline{CS}_4$ to $\overline{CS}_6$ in modes 1 to 6, see section 6.3.2, Chip Se
Pins not assigned to any of these functions are available for generic input/output. Figu
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.



$PA_1/TP_1/\overline{TEND}_1$/TCLKB

$PA_0/TP_0/\overline{TEND}_0$/TCLKA

Pin functions in modes 1, 2, and 5

PA_7 (input/output)/TP_7 (output)/$TIOCB_2$ (input/output)

PA_6 (input/output)/TP_6 (output)/$TIOCA_2$ (input/output)/$\overline{CS}_4$ (output)

PA_5 (input/output)/TP_5 (output)/$TIOCB_1$ (input/output)/$\overline{CS}_5$(output)

PA_4 (input/output)/TP_4 (output)/$TIOCA_1$ (input/output)/$\overline{CS}_6$(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)

Pin functions in modes 3, 4, and 6

A_{20} (output)

PA_6 (input/output)/TP_6 (output)/$TIOCA_2$ (input/output)/A_{21}(output)/$\overline{CS}_4$ (outp

PA_5 (input/output)/TP_5 (output)/$TIOCB_1$ (input/output)/A_{22}(output)/$\overline{CS}_5$ (outp

PA_4 (input/output)/TP_4 (output)/$TIOCA_1$ (input/output)/A_{23}(output)/$\overline{CS}_6$ (outp

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)

Pin functions in mode 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)/$\overline{TEND}_1$ (output)/TCLKB (input)

PA_0 (input/output)/TP_0 (output)/$\overline{TEND}_0$ (output)/TCLKA (input)

Figure 9.10 Port A Pin Configuration

H'FFD1	Port A data direction register	PADDR	W	H'00	H'8
H'FFD3	Port A data register	PADR	R/W	H'00	H'0

Note: * Lower 16 bits of the address.

Port A Data Direction Register (PADDR)

PADDR is an 8-bit write-only register that can select input or output for each pin in po pins are used for TPC output, the corresponding 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, 4, and 6	Initial value	1	0	0	0	0	0	0
	Read/Write	—	W	W	W	W	W	W
Modes 1, 2, 5, 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

While port A acts as an I/O port, a pin in port A becomes an output pin if the correspo PADDR bit is set to 1, and an input pin if this bit is cleared to 0. In modes 3, 4, and 6, fixed at 1 and PA_7 functions as an address output pin.

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

PADDR is initialized to H'00 by a reset and in hardware standby mode in modes 1, 2,

It is initialized to H'80 by a reset and in hardware standby mode in modes 3, 4, and 6. standby mode it retains its previous setting. If a PADDR bit is set to 1, the correspond maintains its output state in software standby mode.

	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.

Mode	1, 2, 5, 7			
ITU 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_7 output
		$TIOCB_2$ input*		

Note: * $TIOCB_2$ input when IOB2 = 1 and PWM2 = 0.

ITU channel 2 settings	(2)	(1)	
IOB2	0		
IOB1	0	0	1
IOB0	0	1	—

ITU channel 2 settings	table below	below				table below	below					table below	
PA_6DDR	—	0	1	1	—	—	0	1	1	—	—	—	0
NDER6	—	—	0	1	—	—	—	0	1	—	—	—	—
Pin function	$TIOCA_2$ output	PA_6 input	PA_6 output	TP_6 output	$\overline{CS}_4$ output	$TIOCA_2$ output	PA_6 input	PA_6 output	TP_6 output	A_{21} output	$\overline{CS}_4$ output	$TIOCA_2$ output	PA inp
		$TIOCA_2$ input*					$TIOCA_2$ input*						T

Note: * $TIOCA_2$ input when IOA2 = 1.

ITU channel 2 settings	(2)	(1)		(2)	
PWM2	0				
IOA2	0			1	
IOA1	0	0	1	—	
IOA0	0	1	—	—	

channel 1 settings	table below	below				table below	below					table below	
PA_5DDR	—	0	1	1	—	—	0	1	1	—	—	—	
NDER5	—	—	0	1	—	—	—	0	1	—	—	—	
Pin function	$TIOCB_1$ output	PA_5 input	PA_5 output	TP_5 output	$\overline{CS}_5$ output	$TIOCB_1$ output	PA_5 input	PA_5 output	TP_5 output	A_{22} output	$\overline{CS}_5$ output	$TIOCB_1$ output	I/
		$TIOCB_1$ input*					$TIOCB_1$ input*						

Note: * $TIOCB_1$ input when IOB2 = 1 and PWM1 = 0.

ITU channel 1 settings	(2)	(1)		
IOB2	0			
IOB1	0	0	1	
IOB0	0	1	—	

channel 2 settings	table below	below				table below	below					table below	
PA_4DDR	—	0	1	1	—	—	0	1	1	—	—	—	0
$NDER_4$	—	—	0	1	—	—	—	0	1	—	—	—	—
Pin function	$TIOCA_1$ output	PA_4 input	PA_4 output	TP_4 output	$\overline{CS}_6$ output	$TIOCA_1$ output	PA_4 input	PA_4 output	TP_4 output	A_{23} output	$\overline{CS}_6$ output	$TIOCA_1$ output	PA inp
		$TIOCA_1$ input*					$TIOCA_1$ input*						T

Note: * TIOCA1 input when IOA2 = 1.

ITU channel 1 settings	(2)	(1)		(2)	
PWM1	0				
IOA2	0			1	
IOA1	0	0	1	—	
IOA0	0	1	—	—	

Pin function	$TIOCB_0$ output	PA_3 input	PA_3 output	T
		$TIOCB_0$ input*1		
	TCLKD input*2			

Notes: 1. $TIOCB_0$ input when IOB2 = 1 and PWM0 = 0.
2. TCLKD input when TPSC2 = TPSC1 = TPSC0 = 1 in any of TC

ITU channel 0 settings	(2)	(1)		
IOB2	0			
IOB1	0	0	1	
IOB0	0	1	—	

Pin function	$TIOCA_0$ output	PA_2 input	PA_2 output	TF
		$TIOCA_0$ input*1		
	TCLKC input*2			

Notes: 1. $TIOCA_0$ input when IOA2 = 1.
2. TCLKC input when TPSC2 = TPSC1 = 1 and TPSC0 = 0 in any c
TCR0.

ITU channel 0 settings	(2)	(1)		(2)	
PWM0	0				
IOA2	0			1	
IOA1	0	0	1	—	
IOA0	0	1	—	—	

NDER1	—	—	0	
Pin function	$\overline{TEND}_1$ output	PA_1 input	PA_1 output	T
	TCLKB input*			

Note: * TCLKB input when MDF = 1 in TMDR, or when TPSC2 = 1, TPS
TPSC0 = 1 in any of TCR4 to TCR0.

DMAC channel 1 settings	(2)		(1)	(2)	(1)	(2)	
DTS2A, DTS1A	Not both 1			Both 1			
DTS0A	—			0	0	1	1
DTS2B	0	1	1	0	1	0	1
DTS1B	—	0	1	—	—	—	0

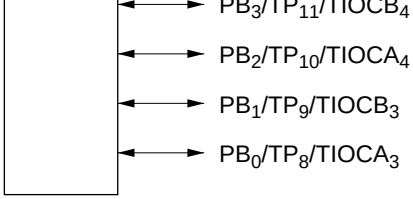
NDER0	—	—	0	
Pin function	$\overline{TEND}_0$ output	PA_0 input	PA_0 output	TF
	TCLKA input*			

Note: * TCLKA input when MDF = 1 in TMDR, or when TPSC2 = 1 and T in any of TCR4 to TCR0.

DMAC channel 0 settings	(2)		(1)	(2)	(1)	(2)	
DTS2A, DTS1A	Not both 1			Both 1			
DTS0A	—			0	0	1	1
DTS2B	0	1	1	0	1	0	1
DTS1B	—	0	1	—	—	—	0

to the DMAC controller (DMAC), ADTRG input to the A/D converter, and $\overline{CS}_7$ output.
hardware standby leaves port B as an input port. For selecting the pin function, see tab
Usage of pins for TPC, ITU, DMAC, and A/D converter input and output is described
sections on those modules. For output of $\overline{CS}_7$ in modes 1 to 6, see section 6.3.2, Chip S
Signals. Pins not assigned to any of these functions are available for generic input/outp
9.11 shows the pin configuration of port B.

Pins in port B can drive one TTL load and a 30-pF capacitive load. They can also driv
darlington transistor pair. Pins PB_3 to PB_0 have Schmitt-trigger inputs.



Pin functions in modes 1 to 6

PB_7 (input/output)/TP_{15} (output)/$\overline{DREQ}_1$ (input)/$\overline{ADTRG}$ (inp

PB_6 (input/output)/TP_{14} (output)/$\overline{DREQ}_0$ (input)/$\overline{CS}_7$ (output

PB_5 (input/output)/TP_{13} (output)/$TOCXB_4$ (output)

PB_4 (input/output)/TP_{12} (output)/$TOCXA_4$ (output)

PB_3 (input/output)/TP_{11} (output)/$TIOCB_4$ (input/output)

PB_2 (input/output)/TP_{10} (output)/$TIOCA_4$ (input/output)

PB_1 (input/output)/TP_9 (output)/$TIOCB_3$ (input/output)

PB_0 (input/output)/TP_8 (output)/$TIOCA_3$ (input/output)

Pin functions in mode 7

PB_7 (input/output)/TP_{15} (output)/$\overline{DREQ}_1$ (input)/$\overline{ADTRG}$ (inp

PB_6 (input/output)/TP_{14} (output)/$\overline{DREQ}_0$ (input)

PB_5 (input/output)/TP_{13} (output)/$TOCXB_4$ (output)

PB_4 (input/output)/TP_{12} (output)/$TOCXA_4$ (output)

PB_3 (input/output)/TP_{11} (output)/$TIOCB_4$ (input/output)

PB_2 (input/output)/TP_{10} (output)/$TIOCA_4$ (input/output)

PB_1 (input/output)/TP_9 (output)/$TIOCB_3$ (input/output)

PB_0 (input/output)/TP_8 (output)/$TIOCA_3$ (input/output)

Figure 9.11 Port B Pin Configuration

Note: * Lower 16 bits of the address.

Port B Data Direction Register (PBDDR)

PBDDR is an 8-bit write-only register that can select input or output for each pin in po
pins are used for TPC output, the corresponding 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

While port B acts as an I/O port, a pin in port B becomes an output pin if the correspo
PBDDR bit is set to 1, and an input pin 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. If a PBDDR bit is set to 1 while port B acts as an I/O po
corresponding pin maintains its output state in software standby mode.

	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 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.

ADTRG select the pin function as follows

PB_7DDR	0	1	
NDER15	—	0	
Pin function	PB_7 input	PB_7 output	TP_1
	$\overline{DREQ}_1$ input*1		
	$\overline{ADTRG}$ input*2		

Notes: 1. $\overline{DREQ}_1$ input under DMAC channel 1 settings (1) in the table b
2. $\overline{ADTRG}$ input when TRGE = 1.

DMAC channel 1 settings	(2)		(1)	(2)	(1)	(2)	
DTS2A, DTS1A	Not both 1			Both 1			
DTS0A	—			0	0	1	
DTS2B	0	1	1	0	1	0	
DTS1B	—	0	1	—	—	—	

	$\overline{DREQ}_0$ input*						$\overline{C}$

Note: * $\overline{DREQ}_0$ input under DMAC channel 0 settings (1) in the table be

DMAC channel 0 settings	(2)		(1)	(2)	(1)	(2)	
DTS2A, DTS1A	Not both 1			Both 1			
DTS0A	—			0	0	1	1
DTS2B	0	1	1	0	1	0	1
DTS1B	—	0	1	—	—	—	0

$PB_5/TP_{13}/TOCXB_4$

ITU channel 4 settings (bit CMD1 in TFCR and bit EXB4 in TOER), bit NDE NDERB, and bit PB_5DDR in PBDDR select the pin function as follows

EXB4, CMD1	Not both 1			
PB_5DDR	0	1	1	
NDER13	—	0	1	
Pin function	PB_5 input	PB_5 output	TP_{13} output	TOC

$PB_4/TP_{12}/TOCXA_4$

ITU channel 4 settings (bit CMD1 in TFCR and bit EXA4 in TOER), bit NDE NDERB, and bit PB_4DDR in PBDDR select the pin function as follows

EXA4, CMD1	Not both 1			
PB_4DDR	0	1	1	
NDER12	—	0	1	
Pin function	PB_4 input	PB_4 output	TP_{12} output	TOC

Pin function	$TIOCB_4$ output	PB_3 input	PB_3 output	T
		$TIOCB_4$ input*		

Note: * $TIOCB_4$ input when CMD1 = PWM4 = 0 and IOB2 = 1.

ITU channel 4 settings	(2)	(2)	(1)		(2)
EB4	0	1			
CMD1	—	0			
IOB2	—	0	0	0	1
IOB1	—	0	0	1	—
IOB0	—	0	1	—	—

Pin function	$TIOCA_4$ output	PB_2 input	PB_2 output	TI
		$TIOCA_4$ input*		

Note: * $TIOCA_4$ input when CMD1 = PWM4 = 0 and IOA2 = 1.

ITU channel 4 settings	(2)	(2)	(1)		(2)	
EA4	0	1				
CMD1	—	0				
PWM4	—	0				1
IOA2	—	0	0	0	1	—
IOA1	—	0	0	1	—	—
IOA0	—	0	1	—	—	—

Pin function	$TIOCB_3$ output	PB_1 input	PB_1 output	T
		$TIOCB_3$ input*		

Note: * $TIOCB_3$ input when CMD1 = PWM3 = 0 and IOB2 = 1.

ITU channel 3 settings	(2)	(2)	(1)		(2)
EB3	0	1			
CMD1	—	0			
IOB2	—	0	0	0	1
IOB1	—	0	0	1	—
IOB0	—	0	1	—	—

Pin function	$TIOCA_3$ output	PB_0 input	PB_0 output	T
		$TIOCA_3$ input*		

Note: * $TIOCA_3$ input when CMD1 = PWM3 = 0 and IOA2 = 1.

ITU channel 3 settings	(2)	(2)	(1)		(2)	
EA3	0	1				
CMD1	—	0				
PWM3	—	0				1
IOA2	—	0	0	0	1	—
IOA1	—	0	0	1	—	—
IOA0	—	0	1	—	—	—

When the ITU is not used, it can be independently halted to conserve power. For detai
section 20.6, Module Standby Function.

10.1.1 Features

ITU features are listed below.

- Capability to process up to 12 pulse outputs or 10 pulse inputs
- Ten general registers (GRs, two per channel) with independently-assignable outpu
  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 (TCNTs) can be preset simultaneously, or cleared
    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 synchronizatio
    five-phase PWM output is possible
- Phase counting mode selectable in channel 2
  Two-phase encoder output can be counted automatically.

- Buffering

  Input capture registers can be double-buffered. Output compare registers can be automatically.

- High-speed access via internal 16-bit bus

  The 16-bit timer counters, general registers, and buffer registers can be accessed at via a 16-bit bus.

- Fifteen interrupt sources

  Each channel has two compare match/input capture interrupts and an overflow inter interrupts can be requested independently.

- Activation of DMA controller (DMAC)

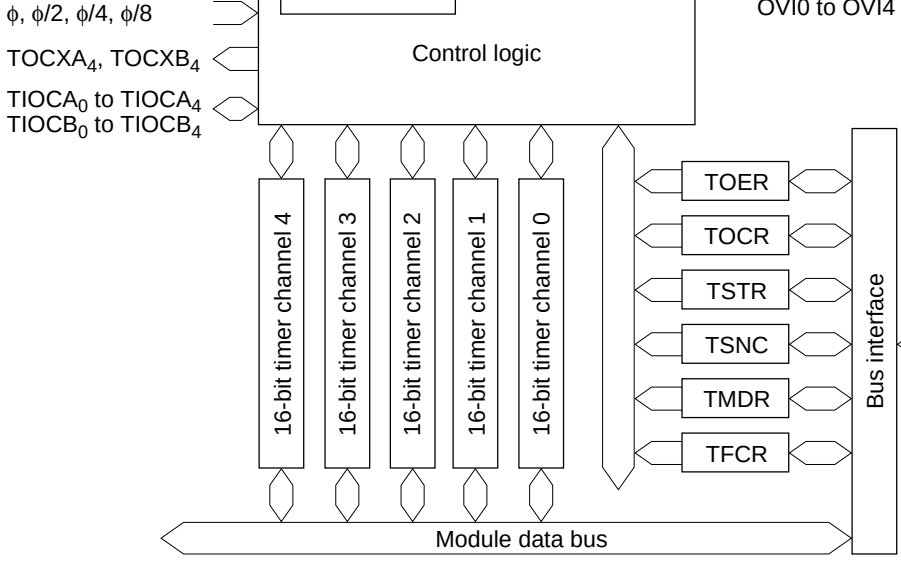
  Four of the compare match/input capture interrupts from channels 0 to 3 can start th

- Output triggering of programmable timing pattern controller (TPC)

  Compare match/input capture signals from channels 0 to 3 can be used as TPC outp

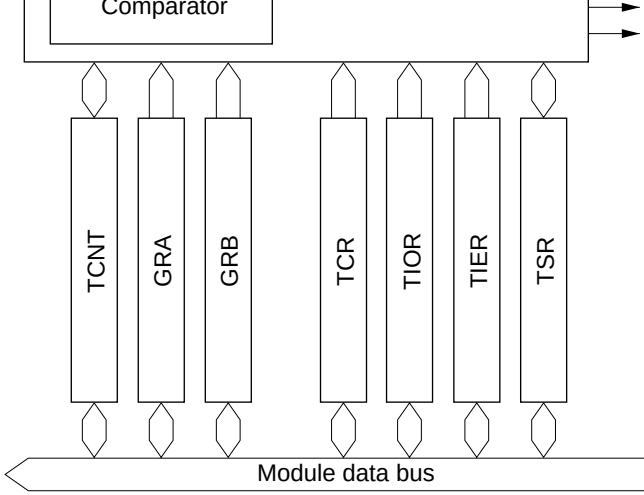
capture registers)						
Buffer registers		—	—	—	BRA3, BRB3	B
Input/output pins		$TIOCA_0$, $TIOCB_0$	$TIOCA_1$, $TIOCB_1$	$TIOCA_2$, $TIOCB_2$	$TIOCA_3$, $TIOCB_3$	T T
Output pins		—	—	—	—	T T
Counter clearing function		GRA0/GRB0 compare match or input capture	GRA1/GRB1 compare match or input capture	GRA2/GRB2 compare match or input capture	GRA3/GRB3 compare match or input capture	G c n in
Compare match output	0	O	O	O	O	O
	1	O	O	O	O	O
	Toggle	O	O	—	O	O
Input capture function		O	O	O	O	O
Synchronization		O	O	O	O	O
PWM mode		O	O	O	O	O
Reset-synchronized PWM mode		—	—	—	O	O
Complementary PWM mode		—	—	—	O	O
Phase counting mode		—	—	O	—	–
Buffering		—	—	—	O	O
DMAC activation		GRA0 compare match or input capture	GRA1 compare match or input capture	GRA2 compare match or input capture	GRA3 compare match or input capture	–
Interrupt sources		Three sources • tch/input capture A0 • Compare match/input capture B0 • Overflow	Three sources • Compare match/input capture A1 • Compare match/input capture B1 • Overflow	Three sources • Compare match/input capture A2 • Compare match/input capture B2 • Overflow	Three sources • Compare match/input capture A3 • Compare match/input capture B3 • Overflow	T • • •

Legend: O: Available
—: Not available



Legend:
TOER: Timer output master enable register (8 bits)
TOCR: Timer output control register (8 bits)
TSTR: Timer start register (8 bits)
TSNC: Timer synchro register (8 bits)
TMDR: Timer mode register (8 bits)
TFCR: Timer function control register (8 bits)

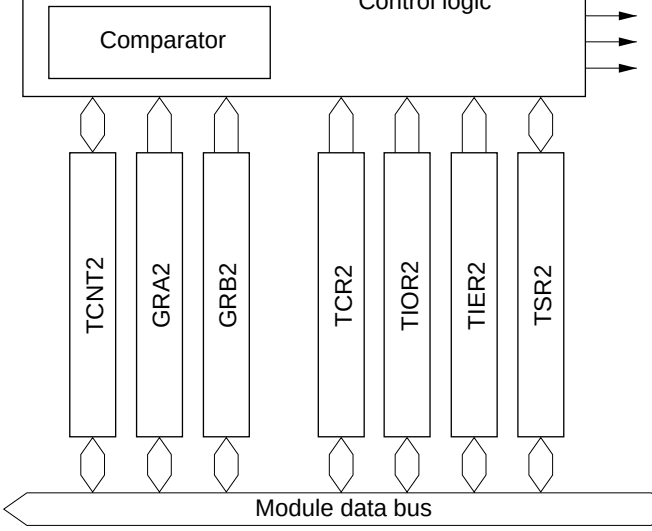
Figure 10.1 ITU Block Diagram (Overall)



Legend:

TCNT:	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)
TIER:	Timer interrupt enable register (8 bits)
TSR:	Timer status register (8 bits)

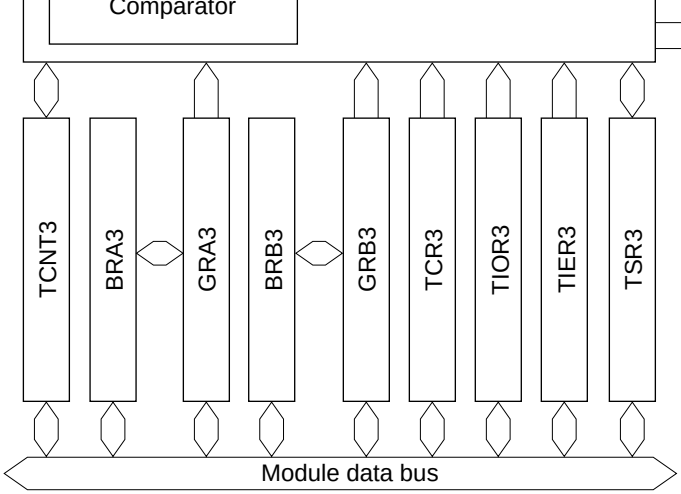
Figure 10.2 Block Diagram of Channels 0 and 1 (for Channel 0)



Legend:

TCNT2:	Timer counter 2 (16 bits)
GRA2, GRB2:	General registers A2 and B2 (input capture/output compare registers (16 bits × 2)
TCR2:	Timer control register 2 (8 bits)
TIOR2:	Timer I/O control register 2 (8 bits)
TIER2:	Timer interrupt enable register 2 (8 bits)
TSR2:	Timer status register 2 (8 bits)

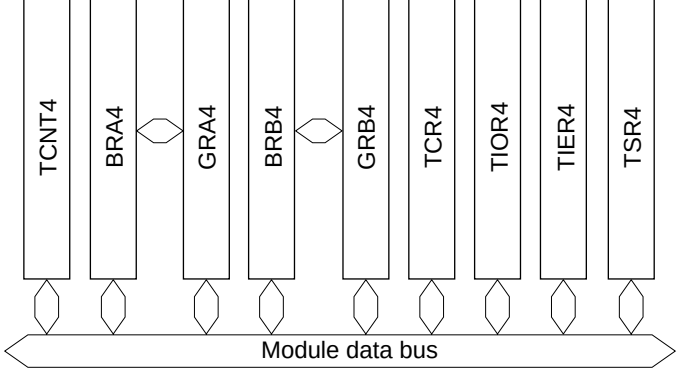
Figure 10.3 Block Diagram of Channel 2



Legend:

TCNT3:	Timer counter 3 (16 bits)
GRA3, GRB3:	General registers A3 and B3 (input capture/output compare register (16 bits × 2)
BRA3, BRB3:	Buffer registers A3 and B3 (input capture/output compare buffer reg (16 bits × 2)
TCR3:	Timer control register 3 (8 bits)
TIOR3:	Timer I/O control register 3 (8 bits)
TIER3:	Timer interrupt enable register 3 (8 bits)
TSR3:	Timer status register 3 (8 bits)

Figure 10.4 Block Diagram of Channel 3



Legend:

TCNT4:	Timer counter 4 (16 bits)
GRA4, GRB4:	General registers A4 and B4 (input capture/output compare registers (16 bits × 2)
BRA4, BRB4:	Buffer registers A4 and B4 (input capture/output compare buffer regis (16 bits × 2)
TCR4:	Timer control register 4 (8 bits)
TIOR4:	Timer I/O control register 4 (8 bits)
TIER4:	Timer interrupt enable register 4 (8 bits)
TSR4:	Timer status register 4 (8 bits)

Figure 10.5 Block Diagram of Channel 4

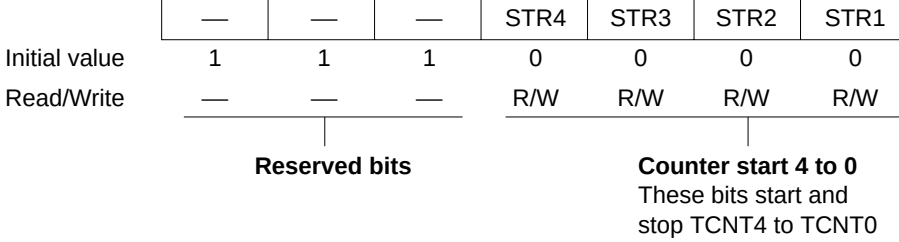
				(phase-A input pin in phase counti
	Clock input B	TCLKB	Input	External clock B input pin (phase-B input pin in phase counti
	Clock input C	TCLKC	Input	External clock C input pin
	Clock input D	TCLKD	Input	External clock D input pin
0	Input capture/output compare A0	$TIOCA_0$	Input/ output	GRA0 output compare or input cap output pin in PWM mode
	Input capture/output compare B0	$TIOCB_0$	Input/ output	GRB0 output compare or input cap
1	Input capture/output compare A1	$TIOCA_1$	Input/ output	GRA1 output compare or input cap output pin in PWM mode
	Input capture/output compare B1	$TIOCB_1$	Input/ output	GRB1 output compare or input cap
2	Input capture/output compare A2	$TIOCA_2$	Input/ output	GRA2 output compare or input cap output pin in PWM mode
	Input capture/output compare B2	$TIOCB_2$	Input/ output	GRB2 output compare or input cap
3	Input capture/output compare A3	$TIOCA_3$	Input/ output	GRA3 output compare or input cap output pin in PWM mode, complem mode, or reset-synchronized PWM
	Input capture/output compare B3	$TIOCB_3$	Input/ output	GRB3 output compare or input cap output pin in complementary PWM reset-synchronized PWM mode
4	Input capture/output compare A4	$TIOCA_4$	Input/ output	GRA4 output compare or input cap output pin in PWM mode, complem mode, or reset-synchronized PWM
	Input capture/output compare B4	$TIOCB_4$	Input/ output	GRB4 output compare or input cap output pin in complementary PWM reset-synchronized PWM mode
	Output compare XA4	$TOCXA_4$	Output	PWM output pin in complementary or reset-synchronized PWM mode
	Output compare XB4	$TOCXB_4$	Output	PWM output pin in complementary or reset-synchronized PWM mode

	H'FF61	Timer synchro register	TSNC	R/W
	H'FF62	Timer mode register	TMDR	R/W
	H'FF63	Timer function control register	TFCR	R/W
	H'FF90	Timer output master enable register	TOER	R/W
	H'FF91	Timer output control register	TOCR	R/W
0	H'FF64	Timer control register 0	TCR0	R/W
	H'FF65	Timer I/O control register 0	TIOR0	R/W
	H'FF66	Timer interrupt enable register 0	TIER0	R/W
	H'FF67	Timer status register 0	TSR0	R/(W)*2
	H'FF68	Timer counter 0 (high)	TCNT0H	R/W
	H'FF69	Timer counter 0 (low)	TCNT0L	R/W
	H'FF6A	General register A0 (high)	GRA0H	R/W
	H'FF6B	General register A0 (low)	GRA0L	R/W
	H'FF6C	General register B0 (high)	GRB0H	R/W
	H'FF6D	General register B0 (low)	GRB0L	R/W
1	H'FF6E	Timer control register 1	TCR1	R/W
	H'FF6F	Timer I/O control register 1	TIOR1	R/W
	H'FF70	Timer interrupt enable register 1	TIER1	R/W
	H'FF71	Timer status register 1	TSR1	R/(W)*2
	H'FF72	Timer counter 1 (high)	TCNT1H	R/W
	H'FF73	Timer counter 1 (low)	TCNT1L	R/W
	H'FF74	General register A1 (high)	GRA1H	R/W
	H'FF75	General register A1 (low)	GRA1L	R/W
	H'FF76	General register B1 (high)	GRB1H	R/W
	H'FF77	General register B1 (low)	GRB1L	R/W

	H'FF7D	Timer counter 2 (low)	TCNT2L	R/W
	H'FF7E	General register A2 (high)	GRA2H	R/W
	H'FF7F	General register A2 (low)	GRA2L	R/W
	H'FF80	General register B2 (high)	GRB2H	R/W
	H'FF81	General register B2 (low)	GRB2L	R/W
3	H'FF82	Timer control register 3	TCR3	R/W
	H'FF83	Timer I/O control register 3	TIOR3	R/W
	H'FF84	Timer interrupt enable register 3	TIER3	R/W
	H'FF85	Timer status register 3	TSR3	R/(W)*2
	H'FF86	Timer counter 3 (high)	TCNT3H	R/W
	H'FF87	Timer counter 3 (low)	TCNT3L	R/W
	H'FF88	General register A3 (high)	GRA3H	R/W
	H'FF89	General register A3 (low)	GRA3L	R/W
	H'FF8A	General register B3 (high)	GRB3H	R/W
	H'FF8B	General register B3 (low)	GRB3L	R/W
	H'FF8C	Buffer register A3 (high)	BRA3H	R/W
	H'FF8D	Buffer register A3 (low)	BRA3L	R/W
	H'FF8E	Buffer register B3 (high)	BRB3H	R/W
	H'FF8F	Buffer register B3 (low)	BRB3L	R/W

H'FF97	Timer counter 4 (low)	TCNT4L	R/W
H'FF98	General register A4 (high)	GRA4H	R/W
H'FF99	General register A4 (low)	GRA4L	R/W
H'FF9A	General register B4 (high)	GRB4H	R/W
H'FF9B	General register B4 (low)	GRB4L	R/W
H'FF9C	Buffer register A4 (high)	BRA4H	R/W
H'FF9D	Buffer register A4 (low)	BRA4L	R/W
H'FF9E	Buffer register B4 (high)	BRB4H	R/W
H'FF9F	Buffer register B4 (low)	BRB4L	R/W

Notes: 1. Lower 16 bits of the address.
2. Only 0 can be written, to clear flags.



TSTR is initialized to H'E0 by a reset and in standby mode.

Bits 7 to 5—Reserved: Read-only bits, always read as 1.

Bit 4—Counter Start 4 (STR4): Starts and stops timer counter 4 (TCNT4).

Bit 4: STR4	Description
0	TCNT4 is halted
1	TCNT4 is counting

Bit 3—Counter Start 3 (STR3): Starts and stops timer counter 3 (TCNT3).

Bit 3: STR3	Description
0	TCNT3 is halted
1	TCNT3 is counting

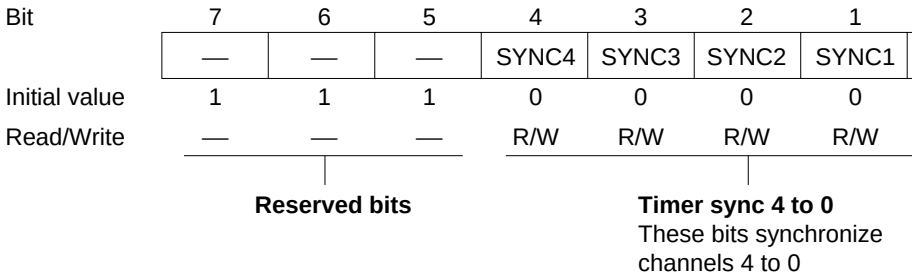
Bit 2—Counter Start 2 (STR2): Starts and stops timer counter 2 (TCNT2).

Bit 2: STR2	Description
0	TCNT2 is halted
1	TCNT2 is counting

Bit 0: STR0	Description	
0	TCNT0 is halted	(I
1	TCNT0 is counting	

10.2.2 Timer Synchro Register (TSNC)

TSNC is an 8-bit readable/writable register that selects whether channels 0 to 4 operate independently or synchronously. Channels are synchronized by setting the correspondi



TSNC is initialized to H'E0 by a reset and in standby mode.

Bits 7 to 5—Reserved: Read-only bits, always read as 1.

Bit 4—Timer Sync 4 (SYNC4): Selects whether channel 4 operates independently or synchronously.

Bit 4: SYNC4	Description	
0	Channel 4's timer counter (TCNT4) operates independently TCNT4 is preset and cleared independently of other channels	(I
1	Channel 4 operates synchronously TCNT4 can be synchronously preset and cleared	

Bit 2—Timer Sync 2 (SYNC2): Selects whether channel 2 operates independently or synchronously.

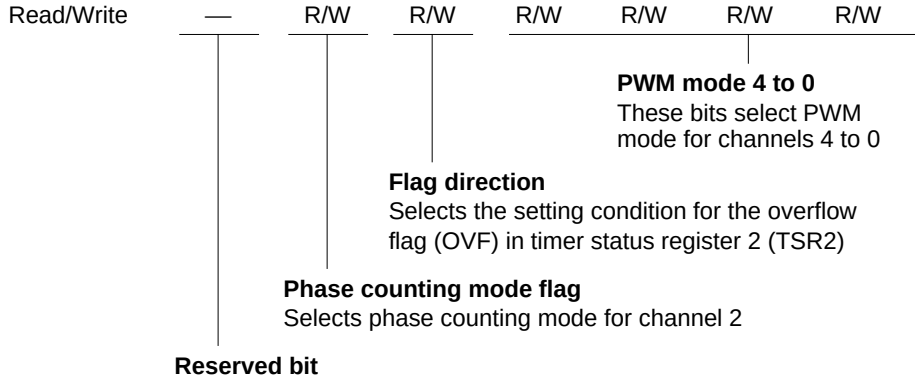
Bit 2: SYNC2	Description
0	Channel 2's timer counter (TCNT2) operates independently TCNT2 is preset and cleared independently of other channels
1	Channel 2 operates synchronously TCNT2 can be synchronously preset and cleared

Bit 1—Timer Sync 1 (SYNC1): Selects whether channel 1 operates independently or synchronously.

Bit 1: SYNC1	Description
0	Channel 1's timer counter (TCNT1) operates independently TCNT1 is preset and cleared independently of other channels
1	Channel 1 operates synchronously TCNT1 can be synchronously preset and cleared

Bit 0—Timer Sync 0 (SYNC0): Selects whether channel 0 operates independently or synchronously.

Bit 0: Bit 0	Description
0	Channel 0's timer counter (TCNT0) operates independently TCNT0 is preset and cleared independently of other channels
1	Channel 0 operates synchronously TCNT0 can be synchronously preset and cleared



TMDR is initialized to H'80 by a reset and in standby mode.

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

Bit 6—Phase Counting Mode Flag (MDF): Selects whether channel 2 operates norma
phase counting mode.

Bit 6: MDF	Description	
0	Channel 2 operates normally	(I
1	Channel 2 operates in phase counting mode	

In phase counting mode channel 2 operates as above regardless of the external clock e
selected by bits CKEG1 and CKEG0 and the clock source selected by bits TPSC2 to T
TCR2. Phase counting mode takes precedence over these settings.

The counter clearing condition selected by the CCLR1 and CCLR0 bits in TCR2 and t
match/input capture settings and interrupt functions of TIOR2, TIER2, and TSR2 rem
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 TSR2 when TCNT2 overflows or underflows
1	OVF is set to 1 in TSR2 when TCNT2 overflows

Bit 4—PWM Mode 4 (PWM4): Selects whether channel 4 operates normally or in P

Bit 4: PWM4	Description
0	Channel 4 operates normally
1	Channel 4 operates in PWM mode

When bit PWM4 is set to 1 to select PWM mode, pin $TIOCA_4$ becomes a PWM outpu
output goes to 1 at compare match with GRA4, and to 0 at compare match with GRB4

If complementary PWM mode or reset-synchronized PWM mode is selected by bits C
CMD0 in TFCR, the CMD1 and CMD0 setting takes precedence and the PWM4 setti
ignored.

If complementary PWM mode or reset-synchronized PWM mode is selected by bits CM
CMD0 in TFCR, the CMD1 and CMD0 setting takes precedence and the PWM3 setting
ignored.

Bit 2—PWM Mode 2 (PWM2): Selects whether channel 2 operates normally or in PW

Bit 2: PWM2	Description	
0	Channel 2 operates normally	(I
1	Channel 2 operates in PWM mode	

When bit PWM2 is set to 1 to select PWM mode, pin $TIOCA_2$ becomes a PWM output
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

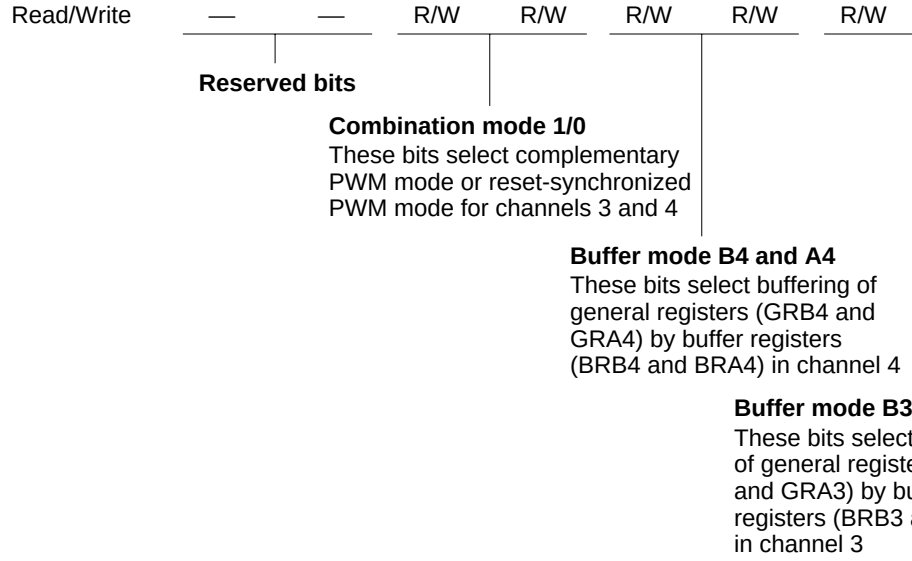
Bit 1: PWM1	Description	
0	Channel 1 operates normally	(I
1	Channel 1 operates in PWM mode	

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	(I
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.



TFCR is initialized to H'C0 by a reset and in standby mode.

Bits 7 and 6—Reserved: Read-only bits, always read as 1.

Bits 5 and 4—Combination Mode 1 and 0 (CMD1, CMD0): These bits select whet
3 and 4 operate in normal mode, complementary PWM mode, or reset-synchronized P

Bit 5: CMD1	Bit 4: CMD0	Description
0	0	Channels 3 and 4 operate normally
	1	
1	0	Channels 3 and 4 operate together in compler PWM mode
	1	Channels 3 and 4 operate together in reset-sy PWM mode

bit is 1 in TSNC).

Bit 3—Buffer Mode B4 (BFB4): Selects whether GRB4 operates normally in channel whether GRB4 is buffered by BRB4.

Bit 3: BFB4	Description
0	GRB4 operates normally (I
1	GRB4 is buffered by BRB4

Bit 2—Buffer Mode A4 (BFA4): Selects whether GRA4 operates normally in channel whether GRA4 is buffered by BRA4.

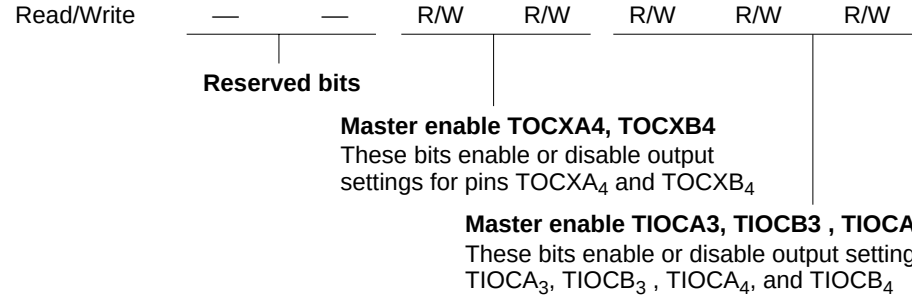
Bit 2: BFA4	Description
0	GRA4 operates normally (I
1	GRA4 is buffered by BRA4

Bit 1—Buffer Mode B3 (BFB3): Selects whether GRB3 operates normally in channel whether GRB3 is buffered by BRB3.

Bit 1: BFB3	Description
0	GRB3 operates normally (I
1	GRB3 is buffered by BRB3

Bit 0—Buffer Mode A3 (BFA3): Selects whether GRA3 operates normally in channel whether GRA3 is buffered by BRA3.

Bit 0: BFA3	Description
0	GRA3 operates normally (I
1	GRA3 is buffered by BRA3



TOER is initialized to H'FF by a reset and in standby mode.

Bits 7 and 6—Reserved: Read-only bits, always read as 1.

Bit 5—Master Enable TOCXB4 (EXB4): Enables or disables ITU output at pin TO

Bit 5: EXB4	Description
0	$TOCXB_4$ output is disabled regardless of TFCR settings ($TOCXB_4$ a generic input/output pin). If XTGD = 0, EXB4 is cleared to 0 when input capture A occurs in
1	$TOCXB_4$ is enabled for output according to TFCR settings

Bit 4—Master Enable TOCXA4 (EXA4): Enables or disables ITU output at pin TO

Bit 4: EXA4	Description
0	$TOCXA_4$ output is disabled regardless of TFCR settings ($TOCXA_4$ a generic input/output pin). If XTGD = 0, EXA4 is cleared to 0 when input capture A occurs in
1	$TOCXA_4$ is enabled for output according to TFCR settings

Bit 2—Master Enable TIOCB4 (EB4): Enables or disables ITU output at pin TIOCB

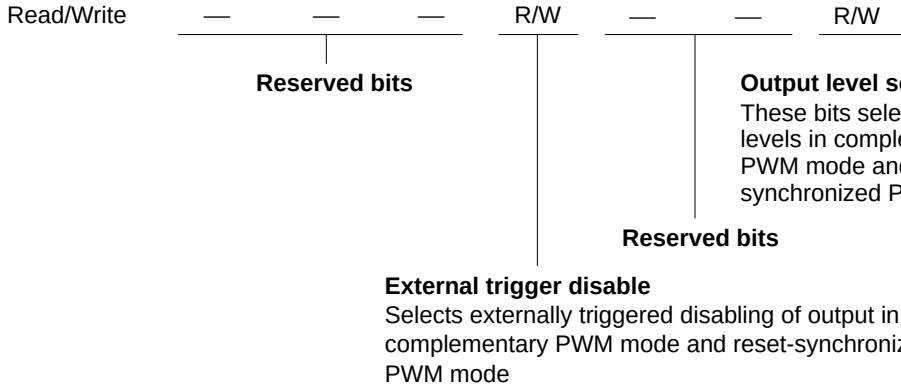
Bit 2: EB4	Description
0	$TIOCB_4$ output is disabled regardless of TIOR4 and TFCR settings (operates as a generic input/output pin). If XTGD = 0, EB4 is cleared to 0 when input capture A occurs in cha
1	$TIOCB_4$ is enabled for output according to TIOR4 and TFCR setting (In

Bit 1—Master Enable TIOCA4 (EA4): Enables or disables ITU output at pin TIOCA

Bit 1: EA4	Description
0	$TIOCA_4$ output is disabled regardless of TIOR4, TMDR, and TFCR s ($TIOCA_4$ operates as a generic input/output pin). If XTGD = 0, EA4 is cleared to 0 when input capture A occurs in cha
1	$TIOCA_4$ is enabled for output according to TIOR4, TMDR, and TFCR (In

Bit 0—Master Enable TIOCA3 (EA3): Enables or disables ITU output at pin TIOCA

Bit 0: EA3	Description
0	$TIOCA_3$ output is disabled regardless of TIOR3, TMDR, and TFCR s ($TIOCA_3$ operates as a generic input/output pin). If XTGD = 0, EA3 is cleared to 0 when input capture A occurs in cha
1	$TIOCA_3$ is enabled for output according to TIOR3, TMDR, and TFCR (In



The settings of the XTGD, OLS4, and OLS3 bits are valid only in complementary PW and reset-synchronized PWM mode. These settings do not affect other modes.

TOCR is initialized to H'FF by a reset and in standby mode.

Bits 7 to 5—Reserved: Read-only bits, always read as 1.

Bit 4—External Trigger Disable (XTGD): Selects externally triggered disabling of in complementary PWM mode and reset-synchronized PWM mode.

Bit 4: XTGD	Description
0	Input capture A in channel 1 is used as an external trigger signal i complementary PWM mode and reset-synchronized PWM mode. When an external trigger occurs, bits 5 to 0 in TOER are cleared t ITU output.
1	External triggering is disabled (

Bits 3 and 2—Reserved: Read-only bits, always read as 1.

reset-synchronized PWM mode.

Bit 0: OLS3	Description
0	$TIOCB_3$, $TOCXA_4$, and $TOCXB_4$ outputs are inverted
1	$TIOCB_3$, $TOCXA_4$, and $TOCXB_4$ outputs are not inverted (In

10.2.7 Timer Counters (TCNT)

TCNT is a 16-bit counter. The ITU has five TCNTs, one for each channel.

Channel	Abbreviation	Function
0	TCNT0	Up-counter
1	TCNT1	
2	TCNT2	Phase counting mode: up/down-counter Other modes: up-counter
3	TCNT3	Complementary PWM mode: up/down-counter
4	TCNT4	Other modes: up-counter

Bit	15	14	13	12	11	10	9	8	7	6	5	4	3	2
Initial value	0	0	0	0	0	0	0	0	0	0	0	0	0	0
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 TCNT is a 16-bit readable/writable register that counts pulse inputs from a clock s
clock source is selected by bits TPSC2 to TPSC0 in TCR.

TCNT0 and TCNT1 are up-counters. TCNT2 is an up/down-counter in phase counting
an up-counter in other modes. TCNT3 and TCNT4 are up/down-counters in compleme
mode and up-counters in other modes.

The TCNTs are linked to the CPU by an internal 16-bit bus and can be written or read
word access or byte access.

Each TCNT is initialized to H'0000 by a reset and in standby mode.

10.2.8 General Registers A, B (GRA, GRB)

The general registers are 16-bit registers. The ITU has 10 general registers, two in eac

Channel	Abbreviation	Function
0	GRA0, GRB0	Output compare/input capture register
1	GRA1, GRB1	
2	GRA2, GRB2	
3	GRA3, GRB3	Output compare/input capture register; can be buffe buffer registers BRA and BRB
4	GRA4, GRB4	

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 o
compare register or an input capture register. The function is selected by settings in TI

When a general register is used as an output compare register, its value is constantly c
with the TCNT value. When the two values match (compare match), the IMFA or IM
to 1 in TSR. Compare match output can be selected in TIOR.

When a general register is used as an input capture register, rising edges, falling edges
edges of an external input capture signal are detected and the current TCNT value is s

General registers are initialized to the output compare function (with no output signal) l
and in standby mode. The initial value is H'FFFF.

10.2.9 Buffer Registers A, B (BRA, BRB)

The buffer registers are 16-bit registers. The ITU has four buffer registers, two each in
and 4.

Channel	Abbreviation	Function
3	BRA3, BRB3	Used for buffering
4	BRA4, BRB4	• When the corresponding GRA or GRB functions output compare register, BRA or BRB can functio output compare buffer register: the BRA or BRB automatically transferred to GRA or GRB at comp • When the corresponding GRA or GRB functions capture register, BRA or BRB can function as an capture buffer register: the GRA or GRB value is automatically transferred to BRA or BRB at input

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

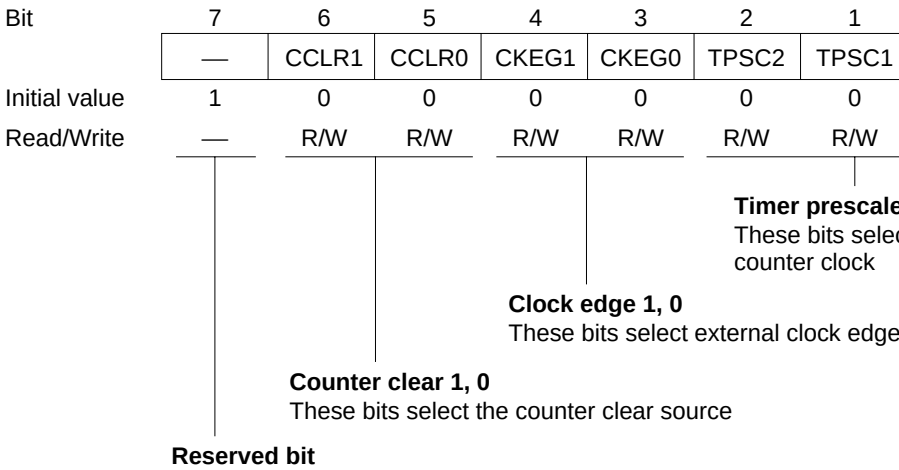
A buffer register is a 16-bit readable/writable register that is used when buffering is sel
Buffering can be selected independently by bits BFB4, BFA4, BFB3, and BFA3 in TF

The buffer register and general register operate as a pair. When the general register fun
output compare register, the buffer register functions as an output compare buffer regis

10.2.10 Timer Control Registers (TCR)

TCR is an 8-bit register. The ITU has five TCRs, one in each channel.

Channel	Abbreviation	Function
0	TCR0	TCR controls the timer counter. The TCRs in all cha
1	TCR1	functionally identical. When phase counting mode is
2	TCR2	channel 2, the settings of bits CKEG1 and CKEG0 a
3	TCR3	to TPSC0 in TCR2 are ignored.
4	TCR4	



Each TCR is an 8-bit readable/writable register that selects the timer counter clock sou
the edge or edges of external clock sources, and selects how the counter is cleared.

TCR is initialized to H'80 by a reset and in standby mode.

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

		capture
	1	Synchronous clear: TCNT is cleared in synchro with other synchronized timers*2

Notes: 1. TCNT is cleared by compare match when the general register functions as a 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 clo edges when an external clock source is used.

Bit 4: CKEG1	Bit 3: CKEG0	Description
0	0	Count rising edges (I
	1	Count falling edges
1	—	Count both edges

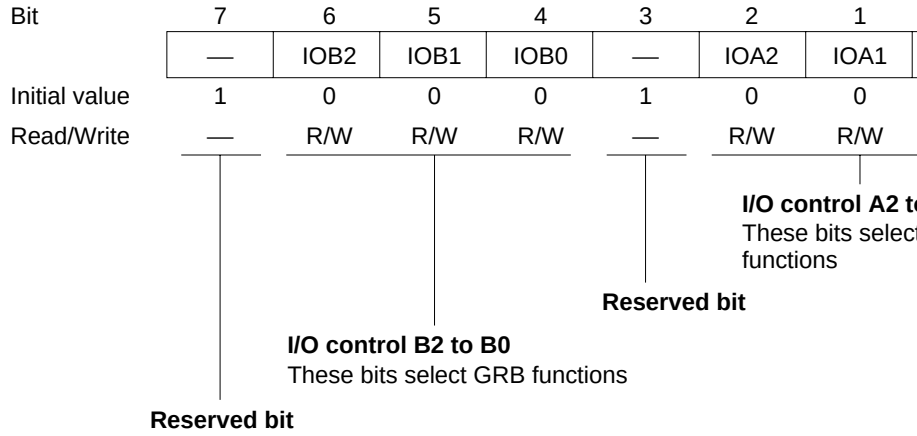
When channel 2 is set to phase counting mode, bits CKEG1 and CKEG0 in TCR2 are i Phase counting takes precedence.

1	0	0	External clock A: TCLKA i
		1	External clock B: TCLKB i
	1	0	External clock C: TCLKC i
		1	External clock D: TCLKD i

When bit TPSC2 is cleared to 0 an internal clock source is selected, and the timer cou
falling edges. When bit TPSC2 is set to 1 an external clock source is selected, and the
the edge or edges selected by bits CKEG1 and CKEG0.

When channel 2 is set to phase counting mode (MDF = 1 in TMDR), the settings of bi
TPSC0 in TCR2 are ignored. Phase counting takes precedence.

3	TIOR3
4	TIOR4



Each TIOR is an 8-bit readable/writable register that selects the output compare or inpu function for GRA and GRB, and specifies the functions of the TIOCA and TIOCB pins output compare function is selected, TIOR also selects the type of output. If input captu selected, TIOR also selects the edge or edges of the input capture signal.

TIOR is initialized to H'88 by a reset and in standby mode.

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

				(1 output in channel 2)*1*2
1	0	0	GRB is an input capture register	GRB captures rising edge of inp
		1		GRB captures falling edge of inp
	1	0		GRB captures both edges of inp
		1		

Notes: 1. After a reset, the output is 0 until the first compare match.
2. Channel 2 output cannot be toggled by compare match. This setting selects instead.

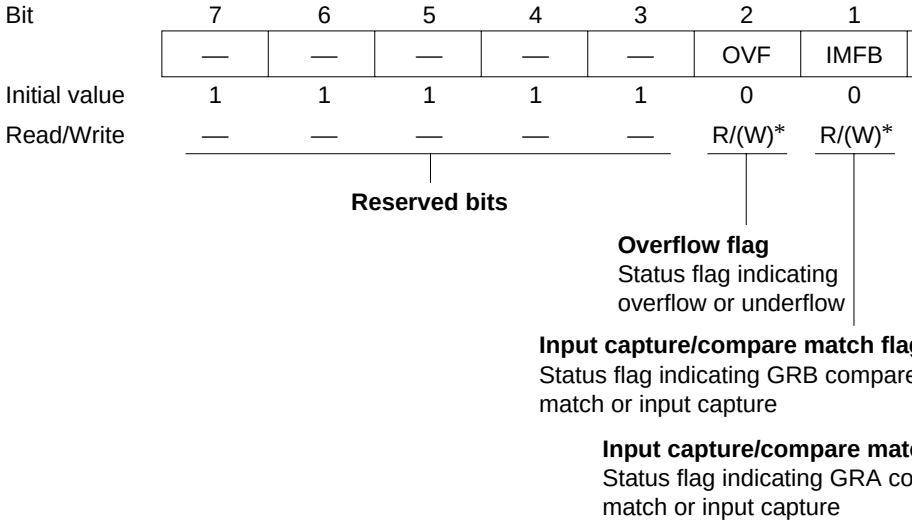
Bit 3—Reserved: Read-only bit, always read as 1.

Bits 2 to 0—I/O Control A2 to A0 (IOA2 to IOA0): These bits select the GRA func

Bit 2: IOA2	Bit 1: IOA1	Bit 0: IOA0	Description	
0	0	0	GRA is an output compare register	No output at compare match (I
		1		0 output at GRA compare match
	1	0		1 output at GRA compare match
		1		Output toggles at GRA compare (1 output in channel 2)*1*2
1	0	0	GRA is an input capture register	GRA captures rising edge of inp
		1		GRA captures falling edge of inp
	1	0		GRA captures both edges of inp
		1		

Notes: 1. After a reset, the output is 0 until the first compare match.
2. Channel 2 output cannot be toggled by compare match. This setting selects instead.

3	TSR3
4	TSR4



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

Each TSR is an 8-bit readable/writable register containing flags that indicate TCNT ov underflow and GRA or GRB compare match or input capture. These flags are interrupt and generate CPU interrupts if enabled by corresponding bits in TIER.

TSR is initialized to H'F8 by a reset and in standby mode.

Bits 7 to 3—Reserved: Read-only bits, always read as 1.

Note: * TCNT underflow occurs when TCNT operates as an up/down-counter. Und
occurs only under the following conditions:
(1) Channel 2 operates in phase counting mode (MDF = 1 in TMDR)
(2) Channels 3 and 4 operate in complementary PWM mode (CMD1 = 1 a
in TFCR)

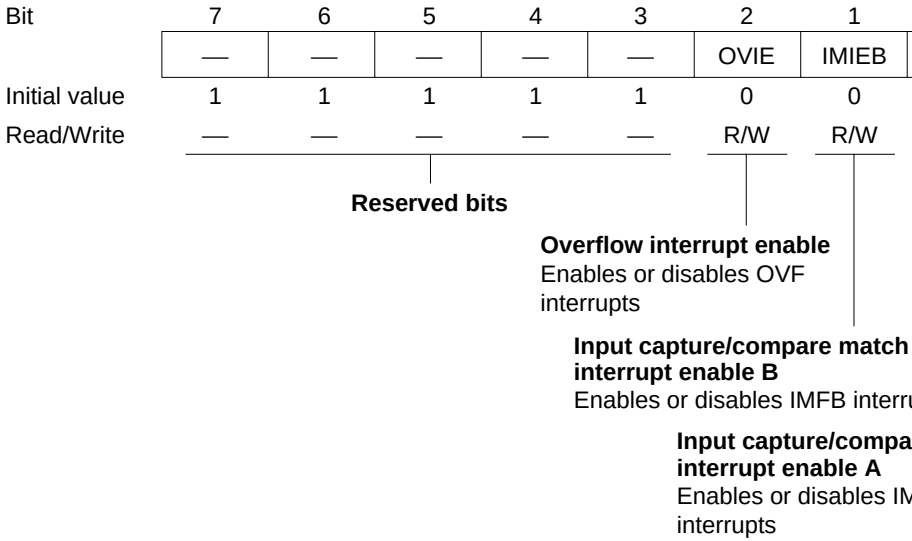
Bit 1—Input Capture/Compare Match Flag B (IMFB): This status flag indicates G
compare match or input capture events.

Bit 1: IMFB	Description
0	[Clearing condition]
	Read IMFB when IMFB = 1, then write 0 in IMFB
1	[Setting conditions]
	TCNT = GRB when GRB functions as an output compare register.
	TCNT value is transferred to GRB by an input capture signal, when functions as an input capture register.

Bit 0—Input Capture/Compare Match Flag A (IMFA): This status flag indicates G
compare match or input capture events.

Bit 0: IMFA	Description
0	[Clearing conditions]
	Read IMFA when IMFA = 1, then write 0 in IMFA.
	DMAC activated by IMIA interrupt (channels 0 to 3 only).
1	[Setting conditions]
	TCNT = GRA when GRA functions as an output compare register.
	TCNT value is transferred to GRA by an input capture signal, when functions as an input capture register.

3	TIER3
4	TIER4



Each TIER is an 8-bit readable/writable register that enables and disables overflow inte requests and general register compare match and input capture interrupt requests.

TIER is initialized to H'F8 by a reset and in standby mode.

Bits 7 to 3—Reserved: Read-only bits, always read as 1.

interrupt requested by the IMFB flag in TSR when IMFB is set to 1.

Bit 1: IMIEB	Description	
0	IMIB interrupt requested by IMFB is disabled	(
1	IMIB interrupt requested by IMFB is enabled	

Bit 0—Input Capture/Compare Match Interrupt Enable A (IMIEA): Enables or d
interrupt requested by the IMFA flag in TSR when IMFA is set to 1.

Bit 0: IMIEA	Description	
0	IMIA interrupt requested by IMFA is disabled	(
1	IMIA interrupt requested by IMFA is enabled	

10.3 CPU Interface

10.3.1 16-Bit Accessible Registers

The timer counters (TCNTs), general registers A and B (GRAs and GRBs), and buffe
and B (BRAs and BRBs) are 16-bit registers, and are linked to the CPU by an internal
bus. These registers can be written or read a word at a time, or a byte at a time.

Figures 10.6 and 10.7 show examples of word access to a timer counter (TCNT). Figu
10.11 show examples of byte access to TCNTH and TCNTL.

Figure 10.6 Access to Timer Counter (CPU Writes to TCNT, Word)

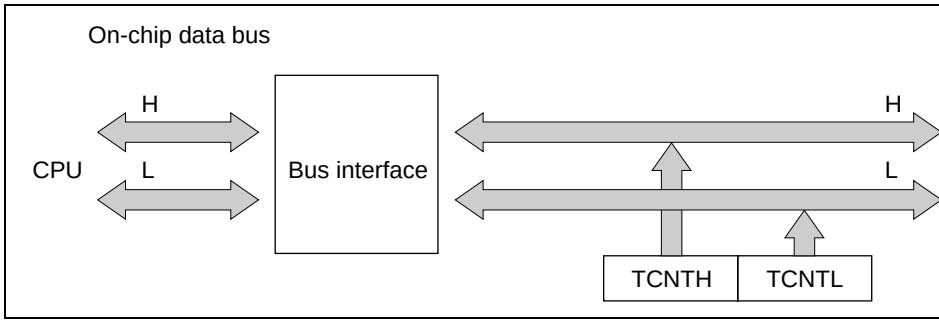


Figure 10.7 Access to Timer Counter (CPU Reads TCNT, Word)

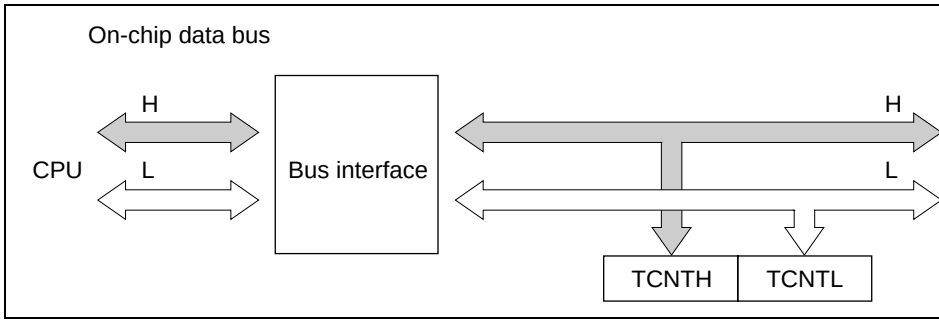


Figure 10.8 Access to Timer Counter (CPU Writes to TCNT, Upper Byt

Figure 10.9 Access to Timer Counter (CPU Writes to TCNT, Lower By

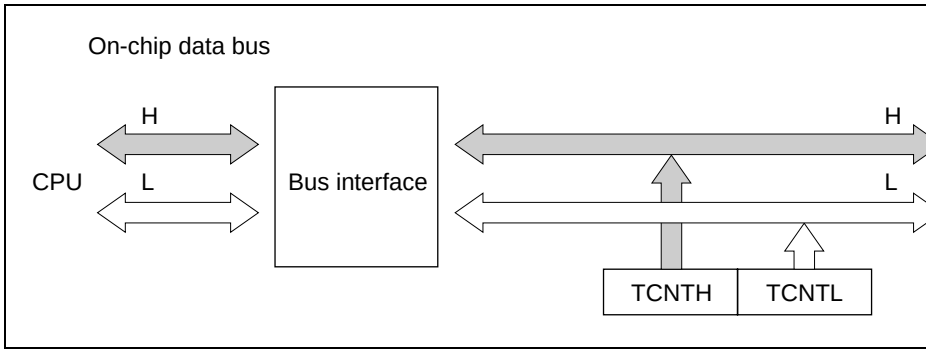


Figure 10.10 Access to Timer Counter (CPU Reads TCNT, Upper Byt

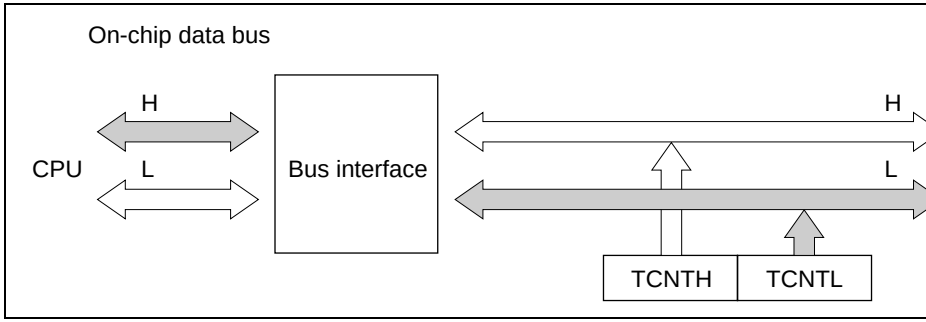


Figure 10.11 Access to Timer Counter (CPU Reads TCNT, Lower Byt

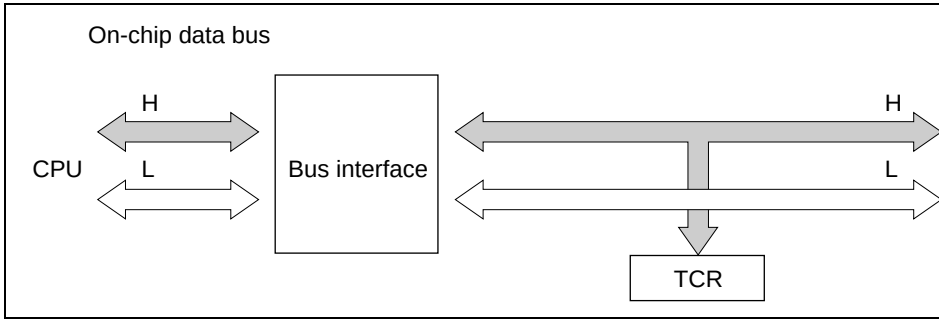


Figure 10.12 Access to Timer Counter (CPU Writes to TCR)

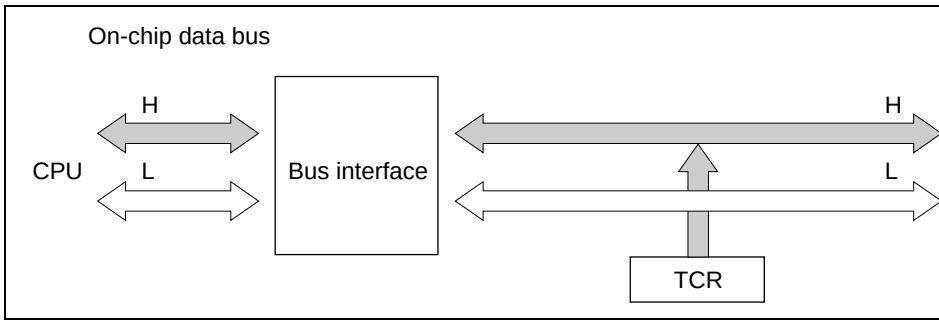


Figure 10.13 Access to Timer Counter (CPU Reads TCR)

Each channel has a timer counter and general registers. The timer counter counts up, a
operate as a free-running counter, periodic counter, or external event counter. General
and B can be used for input capture or output compare.

Synchronous Operation

The timer counters in designated channels are preset synchronously. Data written to th
counter in any one of these channels is simultaneously written to the timer counters in
channels as well. The timer counters can also be cleared synchronously if so designate
CCLR1 and CCLR0 bits in the TCRs.

PWM Mode

A PWM waveform is output from the TIOCA pin. The output goes to 1 at compare ma
to 0 at compare match B. The duty cycle can be varied from 0% to 100% depending o
settings of GRA and GRB. When a channel is set to PWM mode, its GRA and GRB a
become output compare registers.

Reset-Synchronized PWM Mode

Channels 3 and 4 are paired for three-phase PWM output with complementary wavefo
three phases are related by having a common transition point.) When reset-synchroniz
mode is selected GRA3, GRB3, GRA4, and GRB4 automatically function as output c
registers, $TIOCA_3$, $TIOCB_3$, $TIOCA_4$, $TOCXA_4$, $TIOCB_4$, and $TOCXB_4$ function as PW
pins, and TCNT3 operates as an up-counter. TCNT4 operates independently, and is no
with GRA4 or GRB4.

The phase relationship between two clock signals input at TCLKA and TCLKB is dete
TCNT2 counts up or down accordingly. When phase counting mode is selected TCLK
TCLKB become clock input pins and TCNT2 operates as an up/down-counter.

Buffering

- If the general register is an output compare register

  When compare match occurs the buffer register value is transferred to the general re
- If the general register is an input capture register

  When input capture occurs the TCNT value is transferred to the general register, an
  previous general register value is transferred to the buffer register.
- Complementary PWM mode

  The buffer register value is transferred to the general register when TCNT3 and TC
  change counting direction.
- Reset-synchronized PWM mode

  The buffer register value is transferred to the general register at GRA3 compare ma

counter.

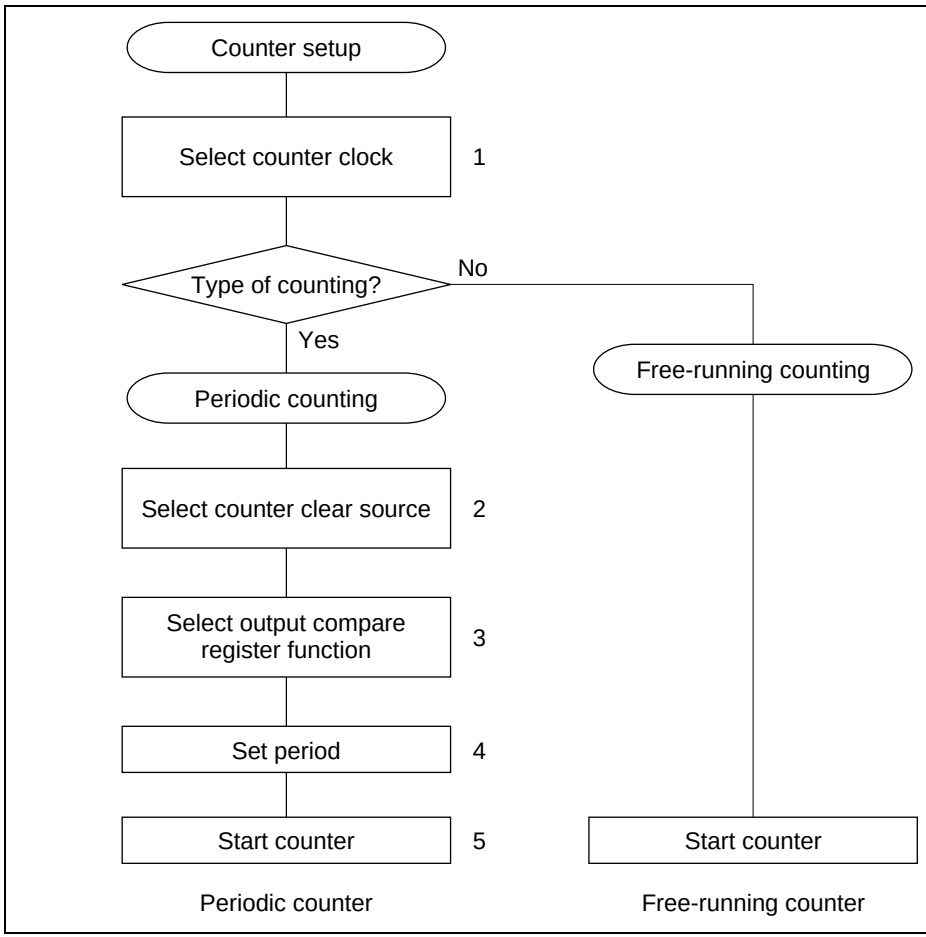


Figure 10.14 Counter Setup Procedure (Example)

5. Set the STR bit to 1 in TSTR to start the timer counter.

Free-running and periodic counter operation: A reset leaves the counters (TCNTs) i
channels 0 to 4 all set as free-running counters. A free-running counter starts counting u
corresponding bit in TSTR is set to 1. When the count overflows from H'FFFF to H'000
OVF flag is set to 1 in TSR. If the corresponding OVIE bit is set to 1 in TIER, a CPU i
requested. After the overflow, the counter continues counting up from H'0000. Figure 1
illustrates free-running counting.

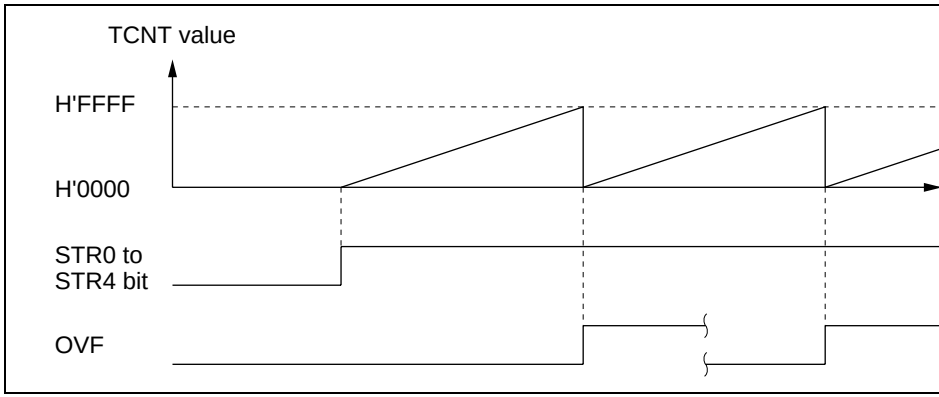


Figure 10.15 Free-Running Counter Operation

When a channel is set to have its counter cleared by compare match, in that channel TC
operates as a periodic counter. Select the output compare function of GRA or GRB, set
or CCLR0 in TCR to have the counter cleared by compare match, and set the count per
or GRB. After these settings, the counter starts counting up as a periodic counter when
corresponding bit is set to 1 in TSTR. When the count matches GRA or GRB, the IMF
flag is set to 1 in TSR and the counter is cleared to H'0000. If the corresponding IMIEA
bit is set to 1 in TIER, a CPU interrupt is requested at this time. After the compare mat
continues counting up from H'0000. Figure 10.16 illustrates periodic counting.

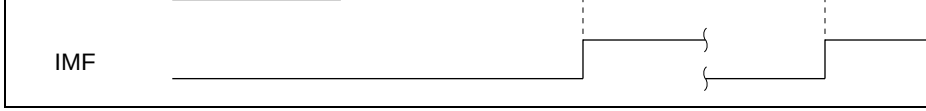


Figure 10.16 Periodic Counter Operation

TCNT count timing:

- Internal clock source

  Bits TPSC2 to TPSC0 in TCR select the system clock (ϕ) or one of three internal c
  obtained by prescaling the system clock (ϕ/2, ϕ/4, ϕ/8).

  Figure 10.17 shows the timing.

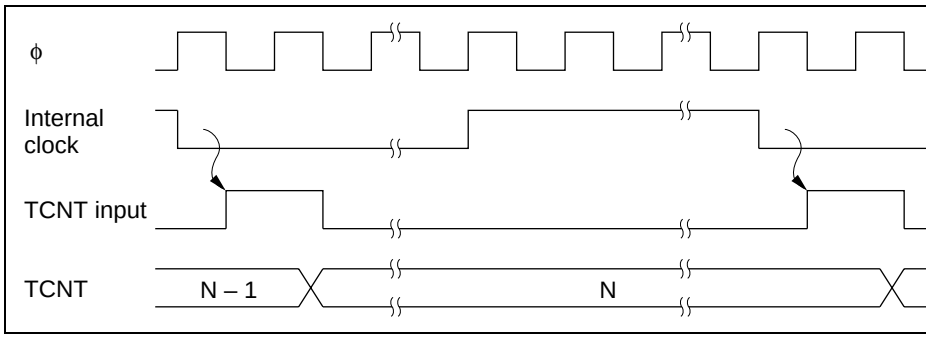


Figure 10.17 Count Timing for Internal Clock Sources

- External clock source

  Bits TPSC2 to TPSC0 in TCR select an external clock input pin (TCLKA to TCLK
  valid edge or edges are selected by bits CKEG1 and CKEG0. The rising edge, falli
  both edges can be selected.

  The pulse width of the external clock signal must be at least 1.5 system clocks whe
  edge is selected, and at least 2.5 system clocks when both edges are selected. Shor
  will not be counted correctly.

  Figure 10.18 shows the timing when both edges are detected.

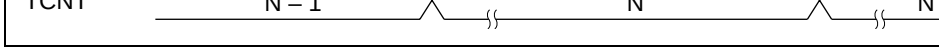


Figure 10.18 Count Timing for External Clock Sources (when Both Edges Are

Waveform Output by Compare Match

In ITU channels 0, 1, 3, and 4, compare match A or B can cause the output at the TIOC TIOCB pin to go to 0, go to 1, or toggle. In channel 2 the output can only go to 0 or go

Sample setup procedure for waveform output by compare match: Figure 10.19 sho sample procedure for setting up waveform output by compare match.

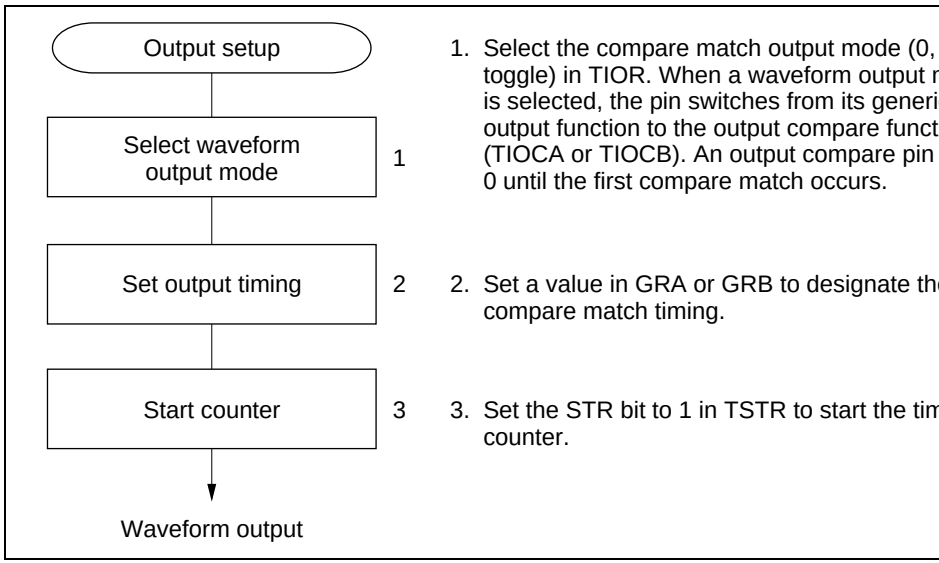


Figure 10.19 Setup Procedure for Waveform Output by Compare Match (Ex

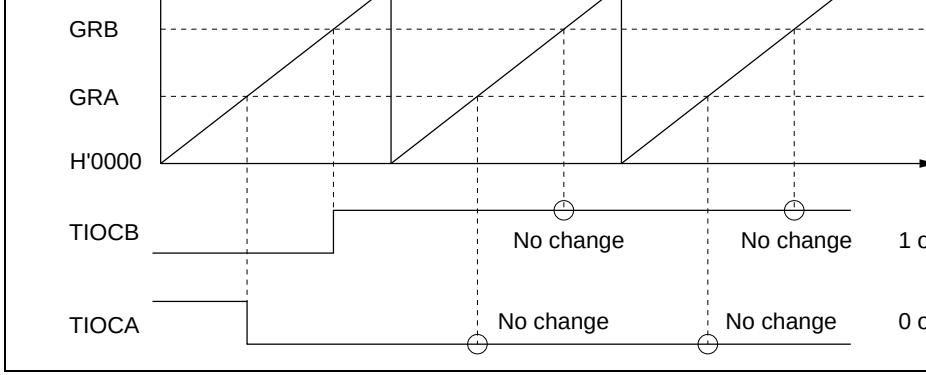


Figure 10.20 0 and 1 Output (Examples)

Figure 10.21 shows examples of toggle output. TCNT operates as a periodic counter, compare match B. Toggle output is selected for both compare match A and B.

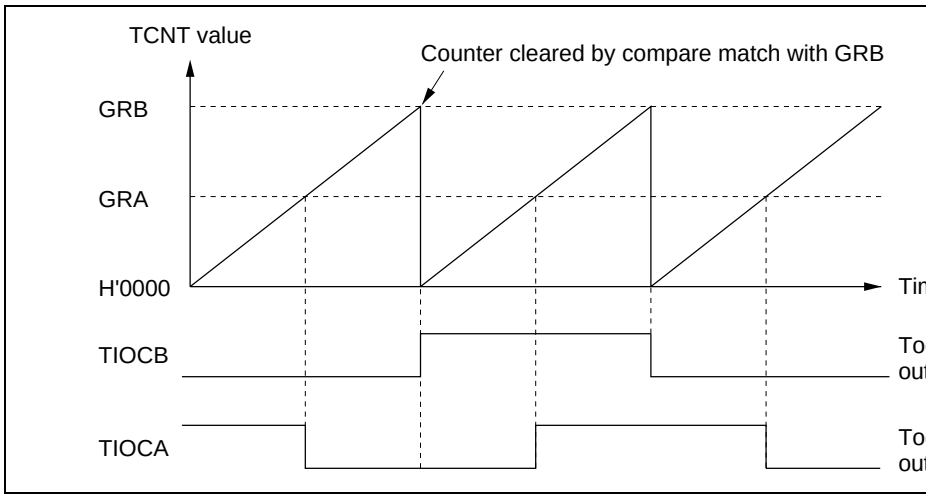


Figure 10.21 Toggle Output (Example)

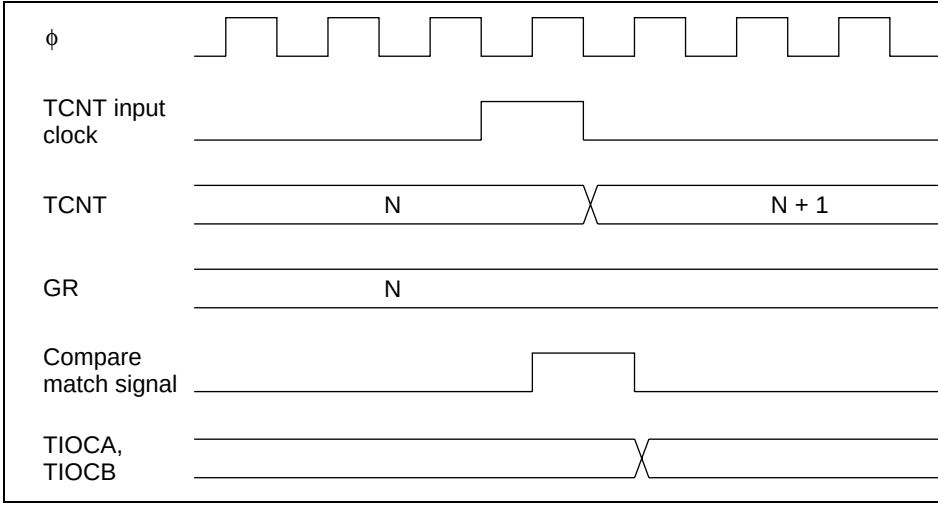
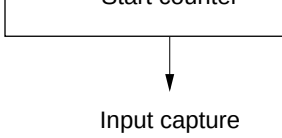


Figure 10.22 Output Compare Timing

Input Capture Function

The TCNT value can be captured into a general register when a transition occurs at an i
capture/output compare pin (TIOCA or TIOCB). Capture can take place on the rising e
edge, or both edges. The input capture function can be used to measure pulse width or p

Sample setup procedure for input capture: Figure 10.23 shows a sample procedure f
up input capture.



2. Set the STR bit to 1 in TSTR to start the t
counter.

Figure 10.23 Setup Procedure for Input Capture (Example)

Examples of input capture: Figure 10.24 illustrates input capture when the falling ed
TIOCB and both edges of TIOCA are selected as capture edges. TCNT is cleared by i
into GRB.

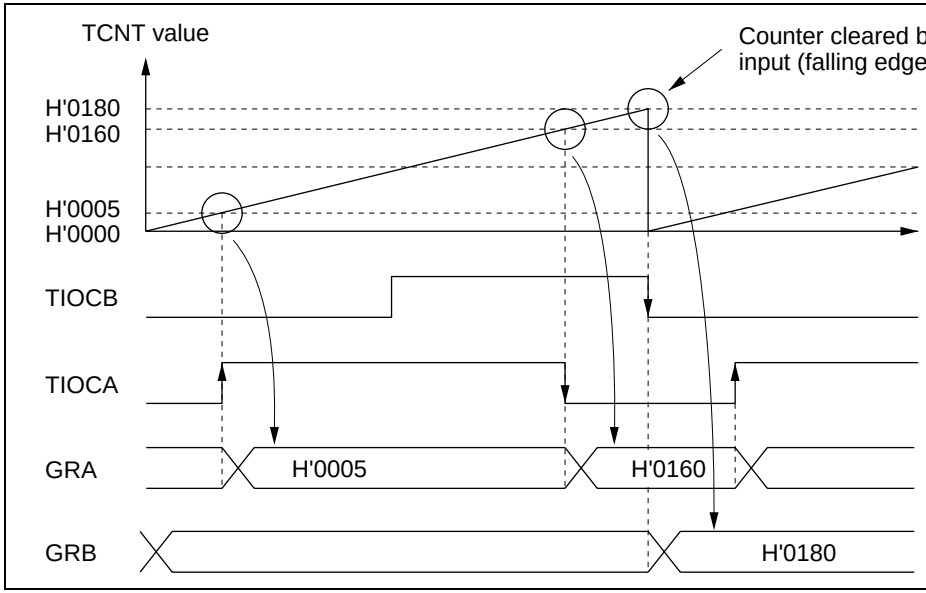


Figure 10.24 Input Capture (Example)

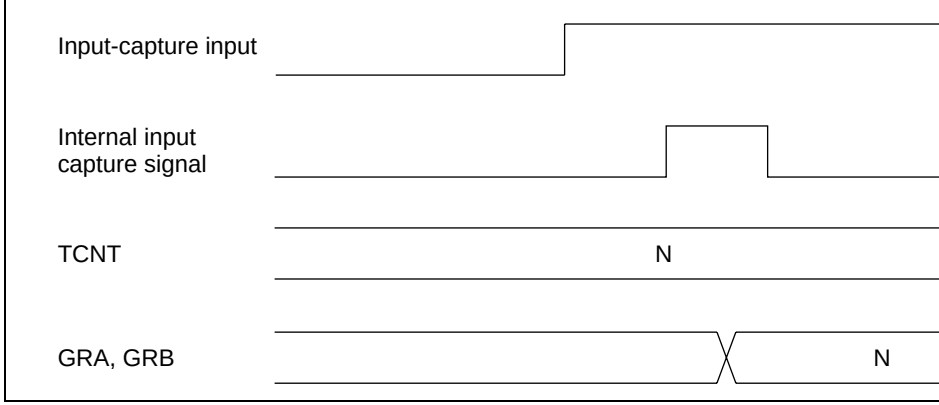
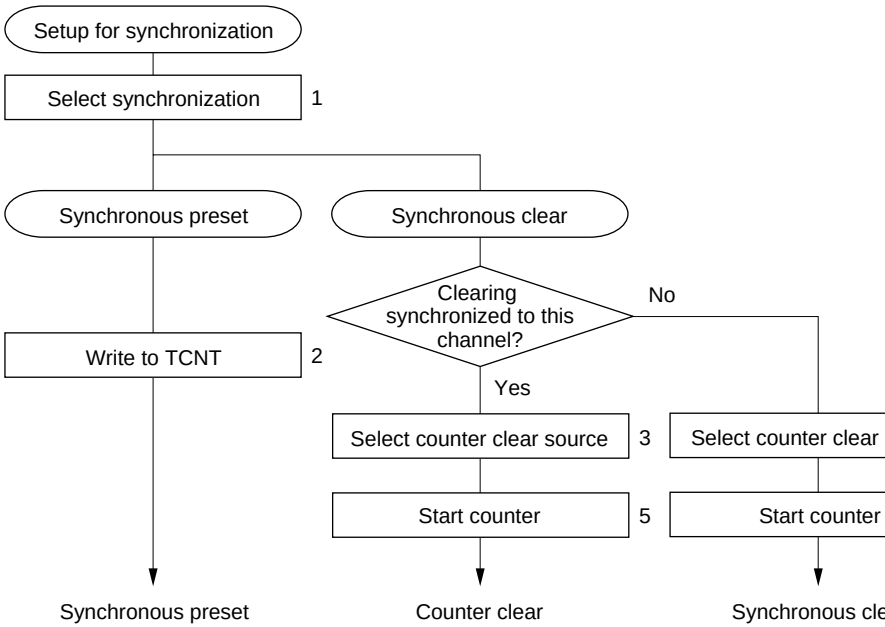


Figure 10.25 Input Capture Signal Timing

Sample Setup Procedure for Synchronization

Figure 10.26 shows a sample procedure for 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 TCNT in one of the synchronized channels, the same value is simultaneously written in TCNT in the other channels (synchronized preset).
3. Set the CCLR1 or CCLR0 bit in TCR to have the counter cleared by compare match or inpu
4. Set the CCLR1 and CCLR0 bits in TCR to have the counter cleared synchronously.
5. Set the STR bits in TSTR to 1 to start the synchronized counters.

Figure 10.26 Setup Procedure for Synchronization (Example)

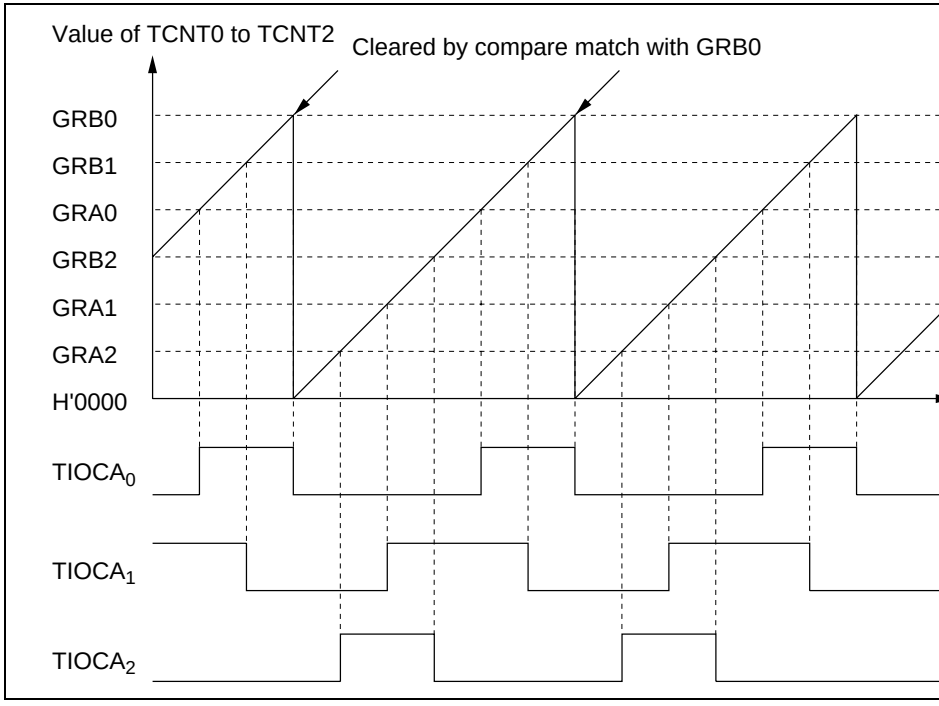
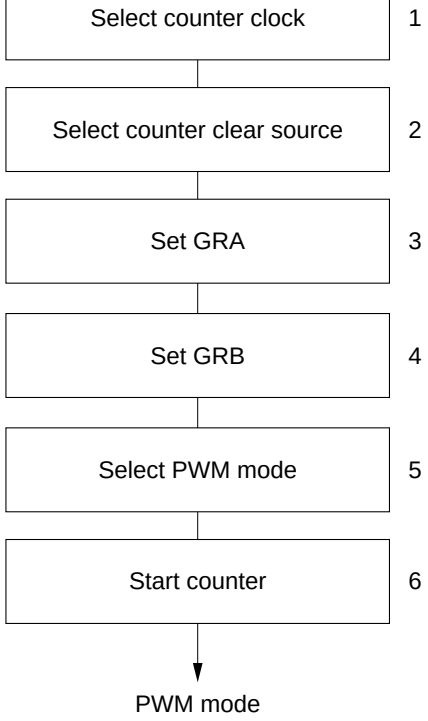


Figure 10.27 Synchronization (Example)

in GRA and GRB, the output does not change when compare match occurs.

Table 10.4 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
3	$TIOCA_3$	GRA3	GRB3
4	$TIOCA_4$	GRA4	GRB4



select the desired edge(s) of the external clock signal.
2. Set bits CCLR1 and CCLR0 in TC 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 selec PWM mode. When PWM mode is selected, regardless of the TIOR contents, GRA and GRB become output compare registers specifyi the times at which the PWM outpu goes to 1 and 0. The TIOCA pin automatically becomes the PWM output pin. The TIOCB pin conforn to the settings of bits IOB1 and IO in TIOR. If TIOCB output is not desired, clear both IOB1 and IOB
6. Set the STR bit to 1 in TSTR to st the timer counter.

Figure 10.28 Setup Procedure for PWM Mode (Example)

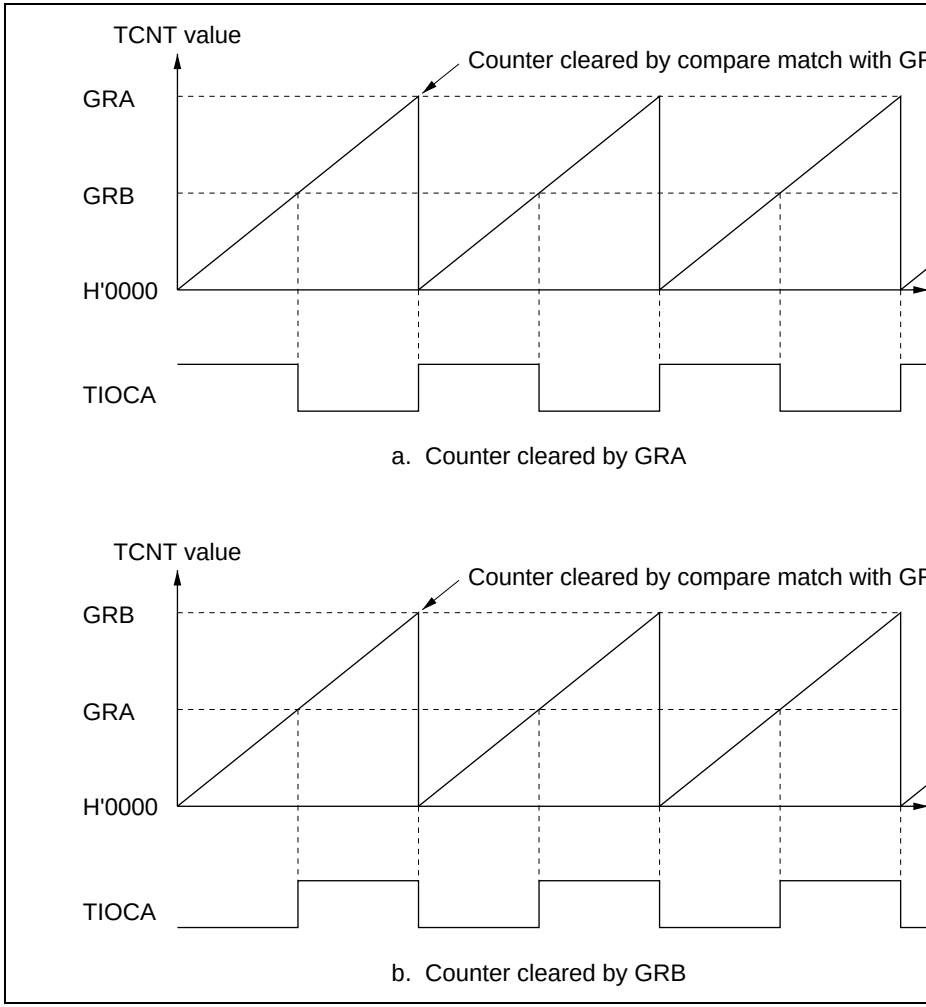


Figure 10.29 PWM Mode (Example 1)

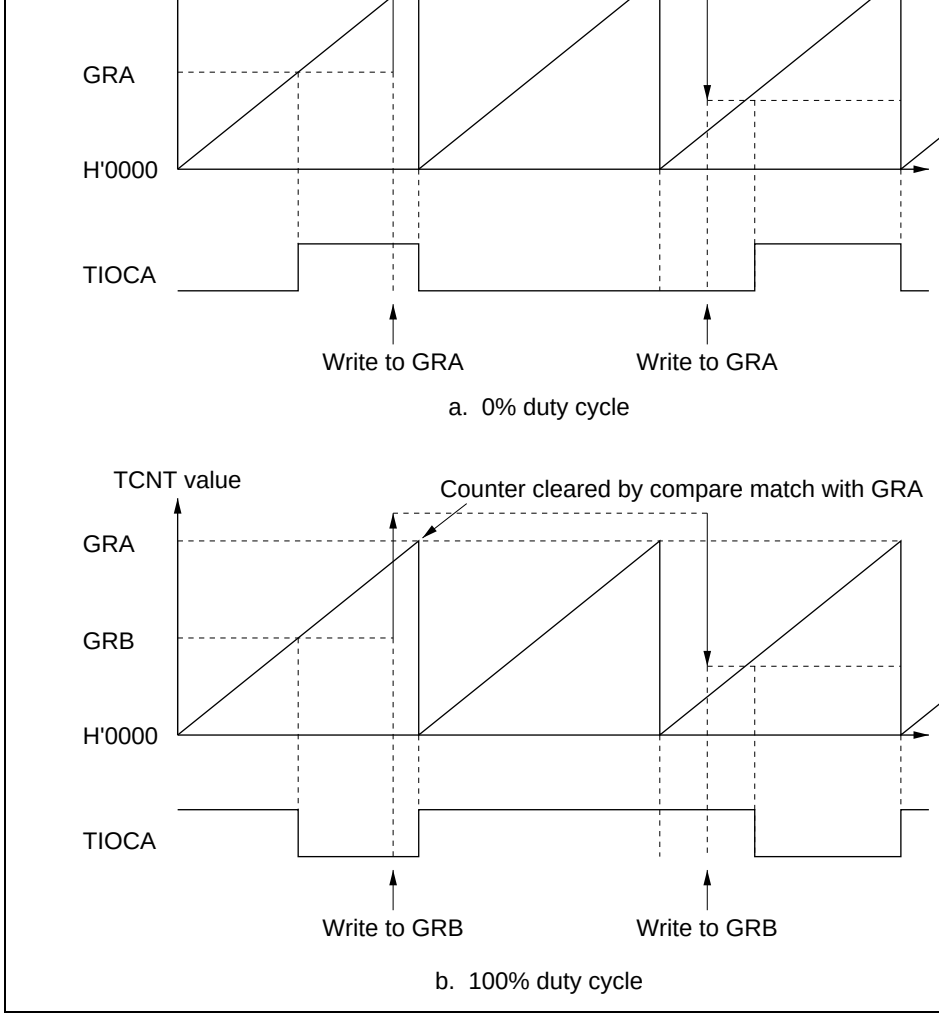


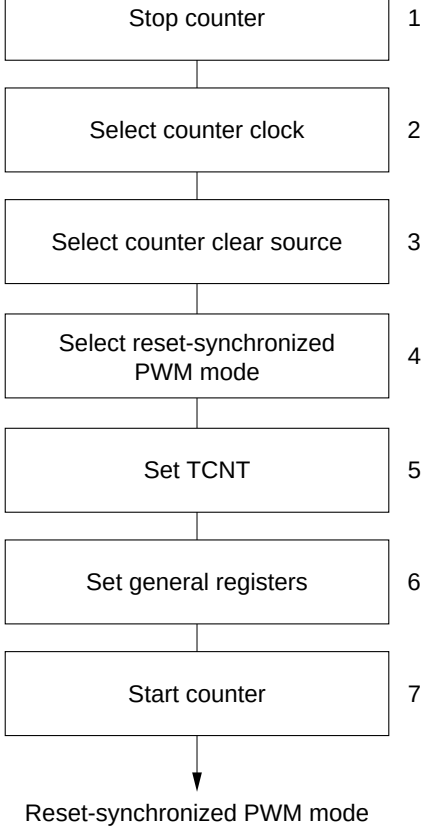
Figure 10.30 PWM Mode (Example 2)

Table 10.5 Output Pins in Reset-Synchronized PWM Mode

Channel	Output Pin	Description
3	$TIOCA_3$	PWM output 1
	$TIOCB_3$	PWM output 1' (complementary waveform to PWM ou
4	$TIOCA_4$	PWM output 2
	$TOCXA_4$	PWM output 2' (complementary waveform to PWM ou
	$TIOCB_4$	PWM output 3
	$TOCXB_4$	PWM output 3' (complementary waveform to PWM ou

Table 10.6 Register Settings in Reset-Synchronized PWM Mode

Register	Setting
TCNT3	Initially set to H'0000
TCNT4	Not used (operates independently)
GRA3	Specifies the count period of TCNT3
GRB3	Specifies a transition point of PWM waveforms output from $TIOCA_3$ and
GRA4	Specifies a transition point of PWM waveforms output from $TIOCA_4$ and
GRB4	Specifies a transition point of PWM waveforms output from $TIOCB_4$ and



TCNT3 is halted.

2. Set bits TPSC2 to TPSC0 in TCR select the counter clock source fo channel 3. If an external clock sou is selected, select the external clo edge(s) with bits CKEG1 and CKE in TCR.
3. Set bits CCLR1 and CCLR0 in TC to select GRA3 compare match as the counter clear source.
4. Set bits CMD1 and CMD0 in TFC select reset-synchronized PWM m $TIOCA_3$, $TIOCB_3$, $TIOCA_4$, TIOCB $TOCXA_4$, and $TOCXB_4$ automatic become PWM output pins.
5. Preset TCNT3 to H'0000. TCNT4 need not be preset.
6. GRA3 is the waveform period reg Set the waveform period value in GRA3. Set transition times of the PWM output waveforms in GRB3, GRA4, and GRB4. Set times withi the compare match range of TCN $X \leq GRA3$ (X: setting value)
7. Set the STR3 bit in TSTR to 1 to s TCNT3.

Figure 10.31 Setup Procedure for Reset-Synchronized PWM Mode (Exam

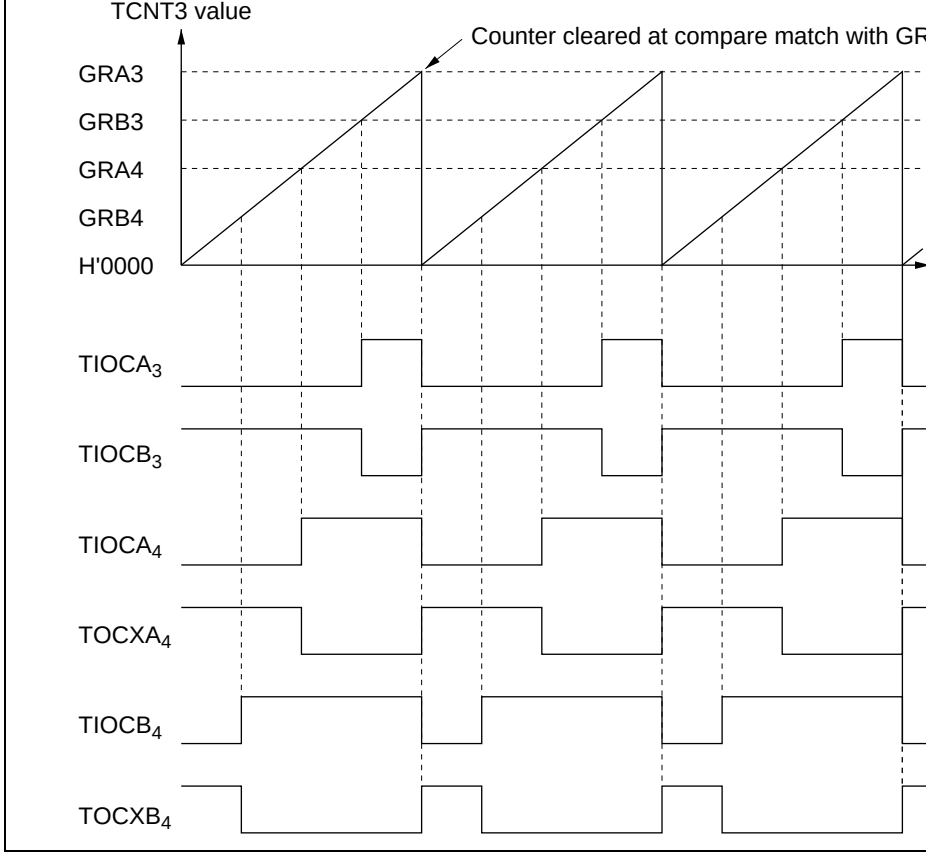


Figure 10.32 Operation in Reset-Synchronized PWM Mode (Example (when OLS3 = OLS4 = 1)

For the settings and operation when reset-synchronized PWM mode and buffer mode selected, see section 10.4.8, Buffering.

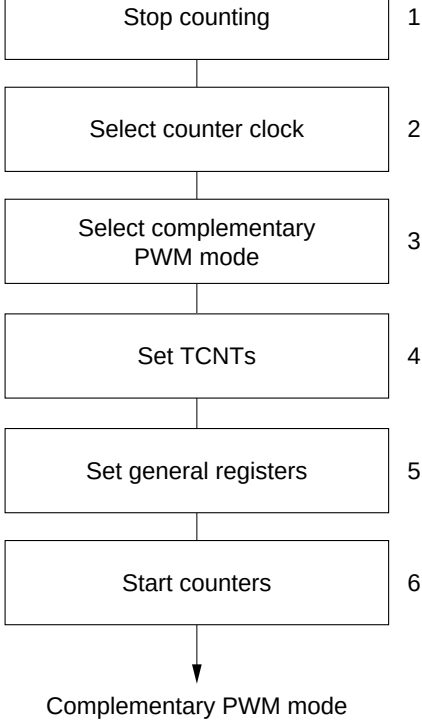
Table 10.7 lists the PWM output pins. Table 10.8 summarizes the register settings.

Table 10.7 Output Pins in Complementary PWM Mode

Channel	Output Pin	Description
3	$TIOCA_3$	PWM output 1
	$TIOCB_3$	PWM output 1' (non-overlapping complementary wavef to PWM output 1)
4	$TIOCA_4$	PWM output 2
	$TOCXA_4$	PWM output 2' (non-overlapping complementary wavef to PWM output 2)
	$TIOCB_4$	PWM output 3
	$TOCXB_4$	PWM output 3' (non-overlapping complementary wavef to PWM output 3)

Table 10.8 Register Settings in Complementary PWM Mode

Register	Setting
TCNT3	Initially specifies the non-overlap margin (difference to TCNT4)
TCNT4	Initially set to H'0000
GRA3	Specifies the upper limit value of TCNT3 minus 1
GRB3	Specifies a transition point of PWM waveforms output from $TIOCA_3$ and T
GRA4	Specifies a transition point of PWM waveforms output from $TIOCA_4$ and T
GRB4	Specifies a transition point of PWM waveforms output from $TIOCB_4$ and T



halted.

2. Set bits TPSC2 to TPSC0 in T
select the same counter clock
for channels 3 and 4. If an exte
clock source is selected, selec
external clock edge(s) with bits
CKEG1 and CKEG0 in TCR. D
select any counter clear sourc
with bits CCLR1 and CCLR0 i

3. Set bits CMD1 and CMD0 in T
to select complementary PWM
$TIOCA_3$, $TIOCB_3$, $TIOCA_4$, TI
$TOCXA_4$, and $TOCXB_4$ autom
become PWM output pins.

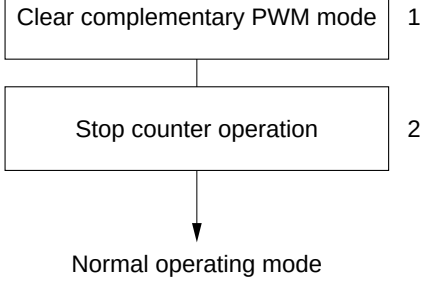
4. Clear TCNT4 to H'0000. Set th
non-overlap margin in TCNT3.
set TCNT3 and TCNT4 to the
value.

5. GRA3 is the waveform period
register. Set the upper limit va
TCNT3 minus 1 in GRA3. Set
transition times of the PWM ou
waveforms in GRB3, GRA4, a
GRB4. Set times within the co
match range of TCNT3 and TC
$T \leq X$ (X: initial setting of GRB
GRA4, or GRB4. T: initial setti
TCNT3)

6. Set bits STR3 and STR4 in TS
1 to start TCNT3 and TCNT4.

Note: After exiting complementary PWM mode, to resume operating in compleme
PWM mode, follow the entire setup procedure from step 1 again.

Figure 10.33 Setup Procedure for Complementary PWM Mode (Examp



2. After setting channels 3 and 4 to
normal operating mode, wait at le
one counter clock period, then cl
bits STR3 and STR4 of TSTR to
stop counter operation of TCNT3
TCNT4.

Figure 10.34 Clearing Procedure for Complementary PWM Mode

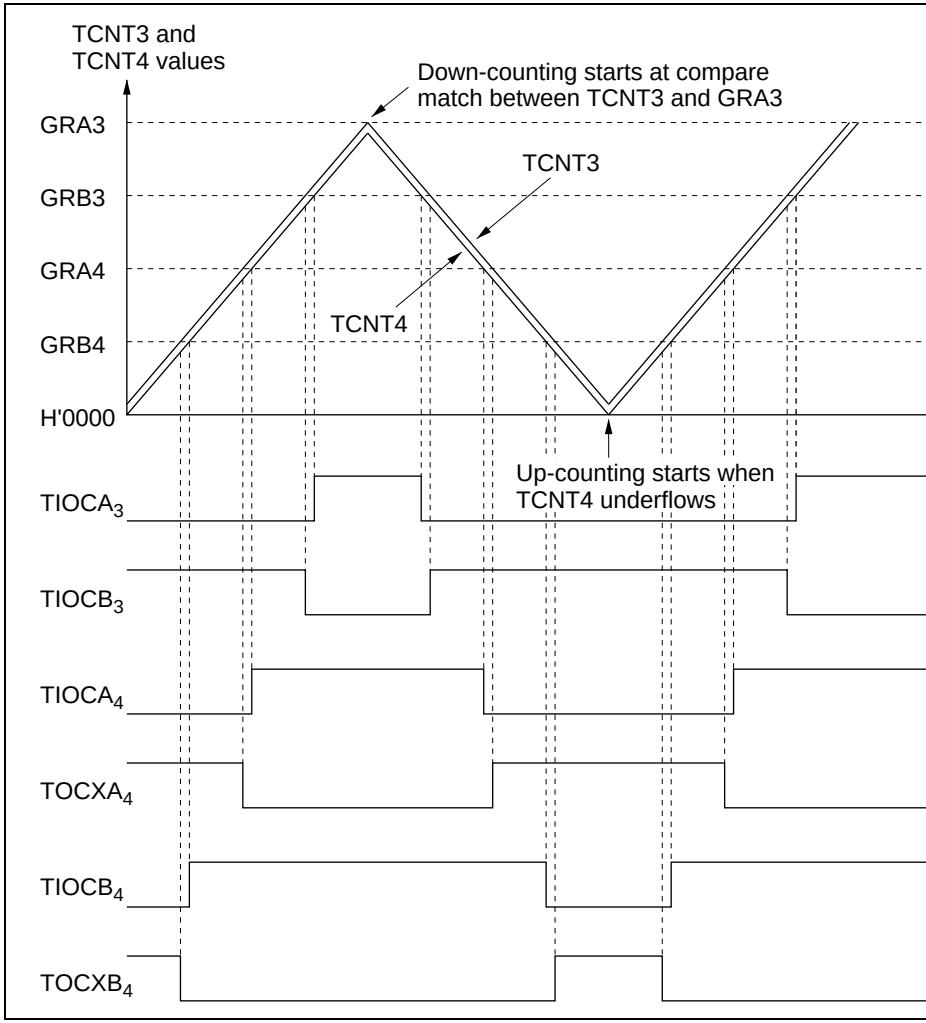


Figure 10.35 Operation in Complementary PWM Mode (Example 1, OLS3 =

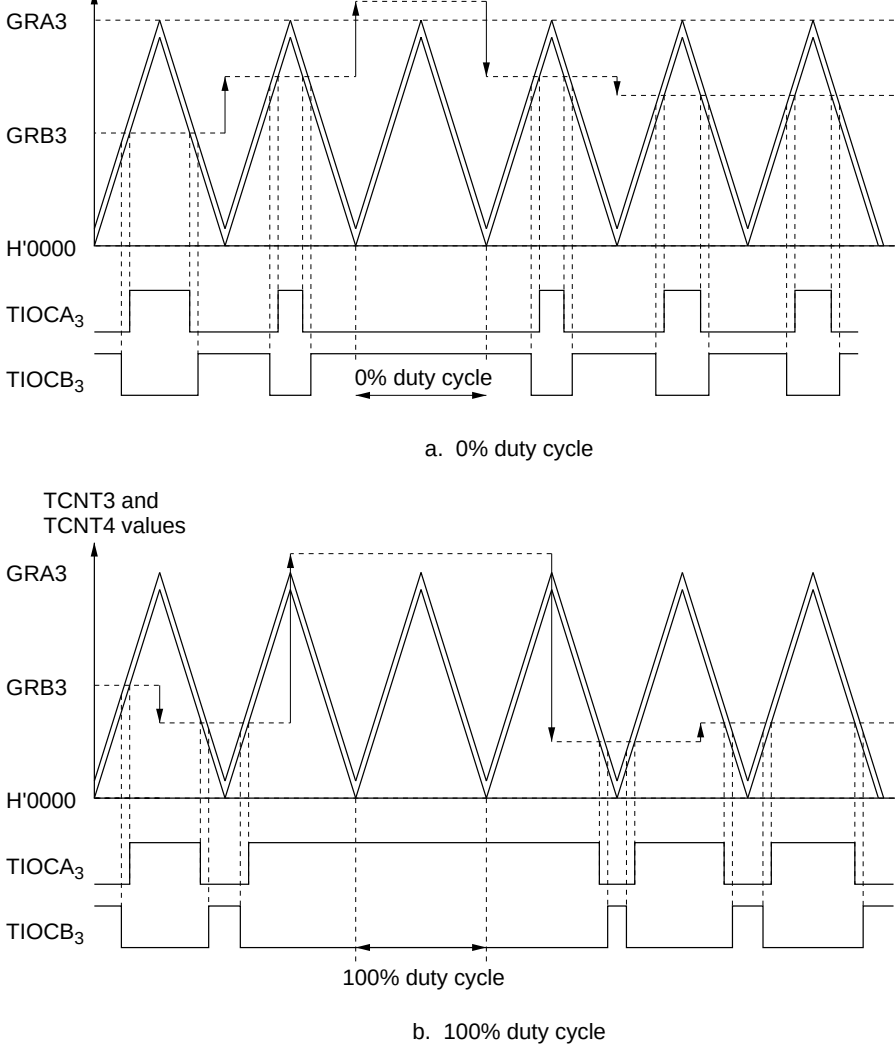


Figure 10.36 Operation in Complementary PWM Mode (Example 2, OLS3 = C

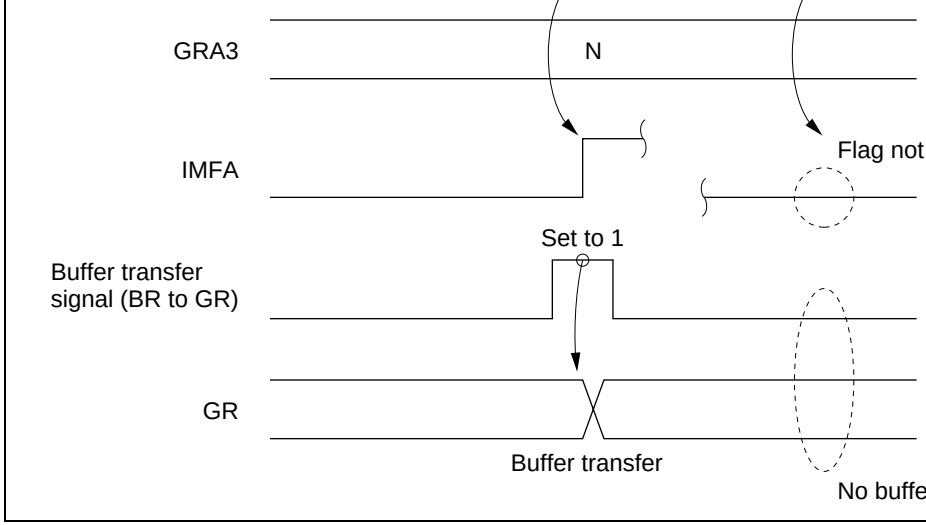


Figure 10.37 Overshoot Timing

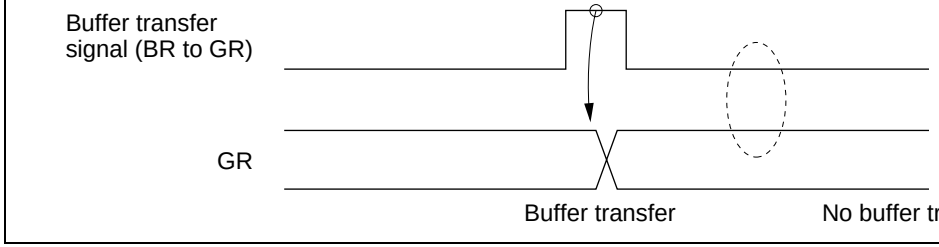


Figure 10.38 Undershoot Timing

In channel 3, IMFA is set to 1 only during up-counting. In channel 4, OVF is set to 1 o an underflow occurs. When buffering is selected, buffer register contents are transferre general register at compare match A3 during up-counting, and when TCNT4 underflow

General Register Settings in Complementary PWM Mode

When setting up general registers for complementary PWM mode or changing their set during operation, note the following points.

- Initial settings

  Do not set values from H'0000 to T – 1 (where T is the initial value of TCNT3). Af counters start and the first compare match A3 event has occurred, however, settings range also become possible.

- Changing settings

  Use the buffer registers. Correct waveform output may not be obtained if a general written to directly.

- Cautions on changes of general register settings

  Figure 10.39 shows six correct examples and one incorrect example.

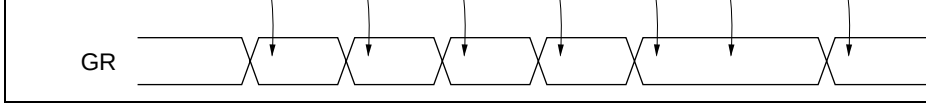


Figure 10.39 Changing a General Register Setting by Buffer Transfer (Exa

— Buffer transfer at transition from up-counting to down-counting

If the general register value is in the range from GRA3 – T + 1 to GRA3, do no buffer register value outside this range. Conversely, if the general register valu this range, do not transfer a value within this range. See figure 10.40.

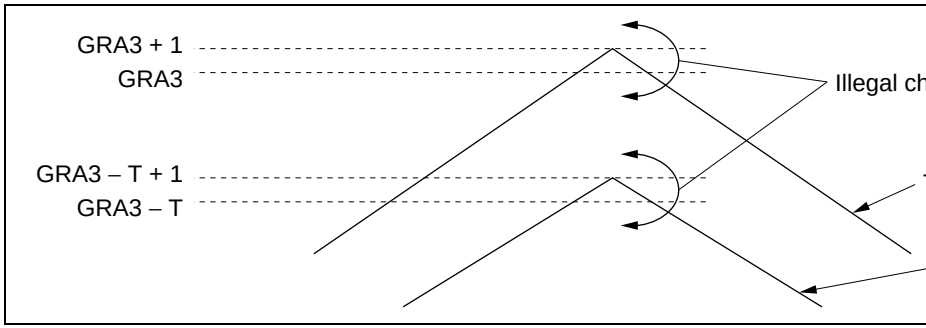


Figure 10.40 Changing a General Register Setting by Buffer Transfer (Cau

— Buffer transfer at transition from down-counting to up-counting

If the general register value is in the range from H'0000 to T – 1, do not transfe register value outside this range. Conversely, when a general register value is o range, do not transfer a value within this range. See figure 10.41.

H'FFFF

Figure 10.41 Changing a General Register Setting by Buffer Transfer (Caut

— General register settings outside the counting range (H'0000 to GRA3)

Waveforms with a duty cycle of 0% or 100% can be output by setting a general a value outside the counting range. When a buffer register is set to a value outsi counting range, then later restored to a value within the counting range, the cou direction (up or down) must be the same both times. See figure 10.42.

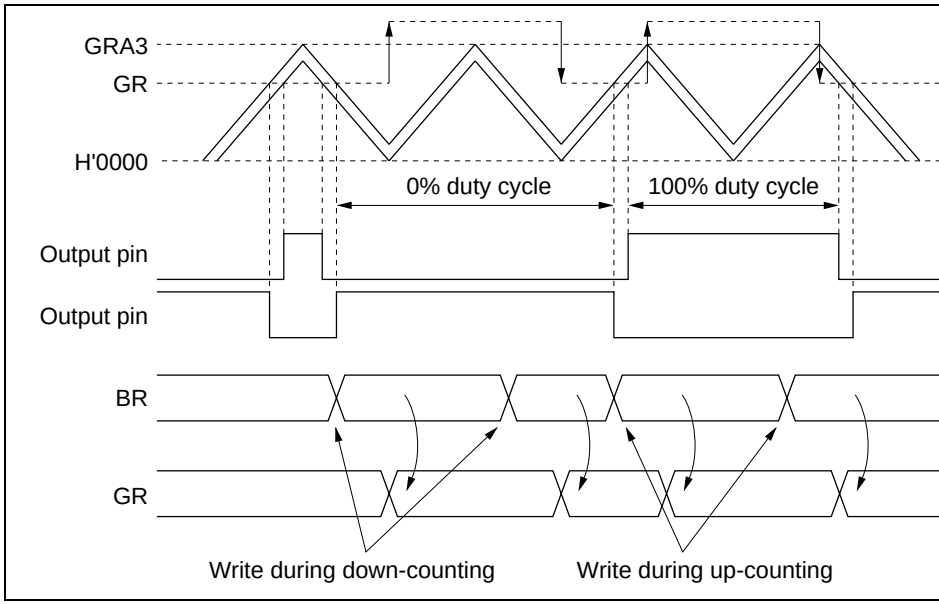


Figure 10.42 Changing a General Register Setting by Buffer Transfer (Exam

Settings can be made in this way by detecting GRA3 compare match or TCNT4 before writing to the buffer register. They can also be made by using GRA3 com to activate the DMAC.

functions can be used, and interrupts can be generated.

Phase counting is available only in channel 2.

Sample Setup Procedure for Phase Counting Mode

Figure 10.43 shows a sample procedure for setting up phase counting mode.

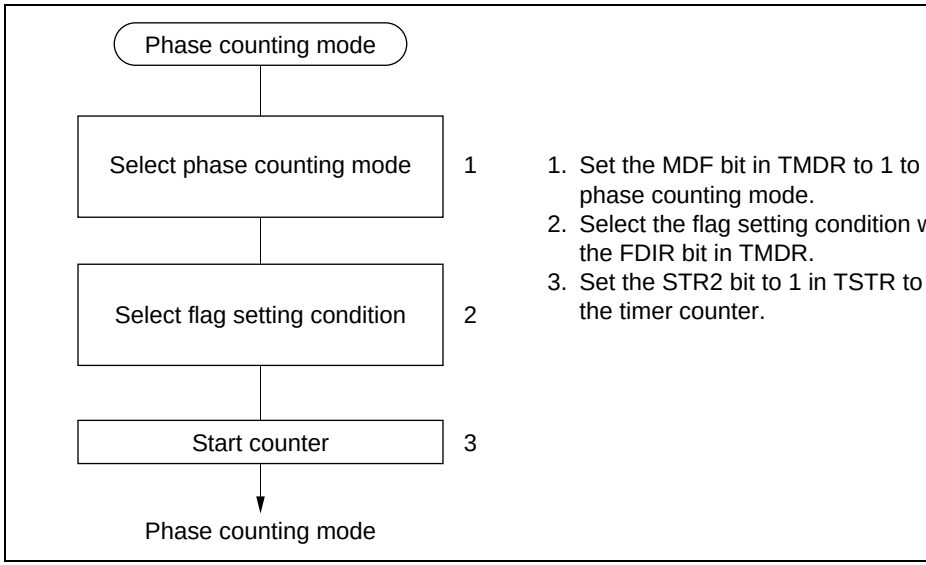


Figure 10.43 Setup Procedure for Phase Counting Mode (Example)

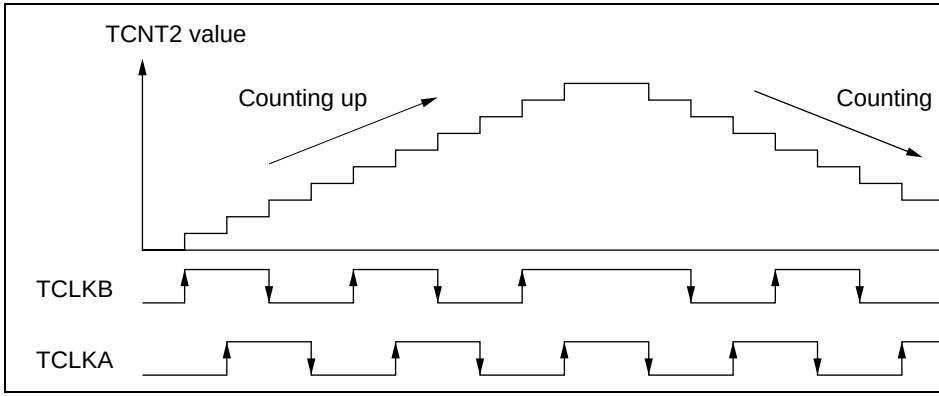


Figure 10.44 Operation in Phase Counting Mode (Example)

Table 10.9 Up/Down Counting Conditions

Counting Direction	Up-Counting				Down-Counting		
TCLKB	↑	High	↓	Low	High	↓	Low
TCLKA	Low	↑	High	↓	↓	Low	↑

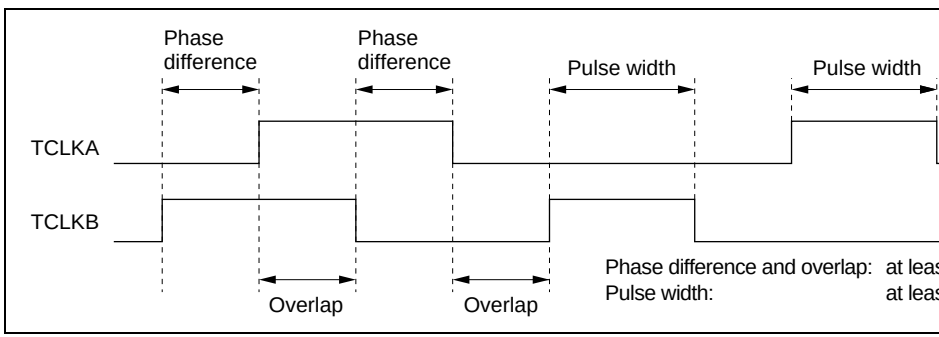


Figure 10.45 Phase Difference, Overlap, and Pulse Width in Phase Counting

The buffer register value is transferred to the general register at compare match.

See figure 10.46.

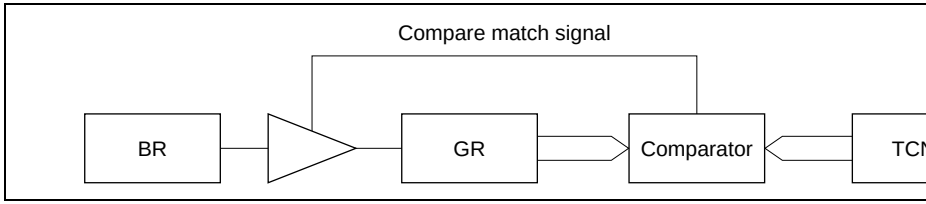


Figure 10.46 Compare Match Buffering

- General register used for input capture

  The TCNT value is transferred to the general register at input capture. The previou register value is transferred to the buffer register.

  See figure 10.47.

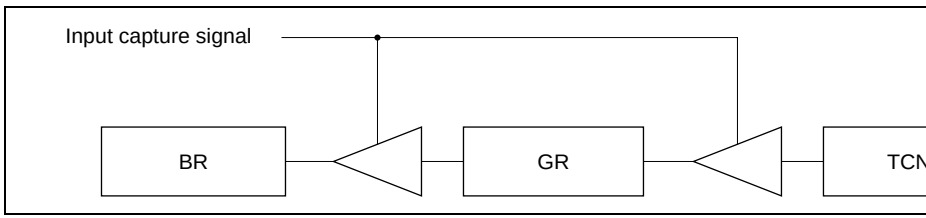


Figure 10.47 Input Capture Buffering

- Complementary PWM mode

  The buffer register value is transferred to the general register when TCNT3 and TC change counting direction. This occurs at the following two times:

  — When TCNT3 compare matches GRA3

  — When TCNT4 underflows

- Reset-synchronized PWM mode

  The buffer register value is transferred to the general register at compare match A3

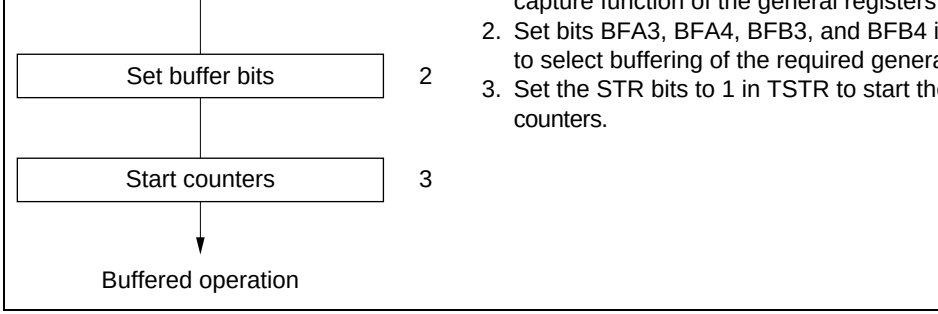


Figure 10.48 Buffering Setup Procedure (Example)

Examples of Buffering

Figure 10.49 shows an example in which GRA is set to function as an output compare r
buffered by BRA, TCNT is set to operate as a periodic counter cleared by GRB compar
and TIOCA and TIOCB are set to toggle at compare match A and B. Because of the bu
when TIOCA toggles at compare match A, the BRA value is simultaneously transferred
This operation is repeated each time compare match A occurs. Figure 10.50 shows the
timing.

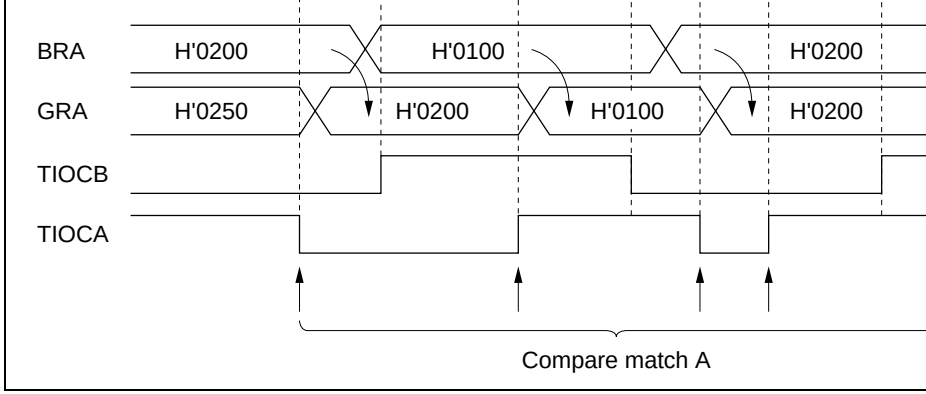


Figure 10.49 Register Buffering (Example 1: Buffering of Output Compare

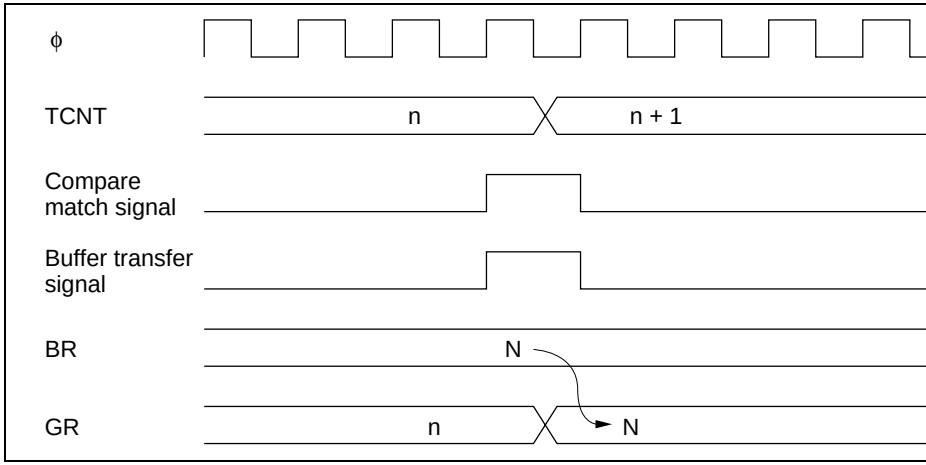


Figure 10.50 Compare Match and Buffer Transfer Timing (Example

Figure 10.51 shows an example in which GRA is set to function as an input capture re buffered by BRA, and TCNT is cleared by input capture B. The falling edge is selecte input capture edge at TIOCB. Both edges are selected as input capture edges at TIOC

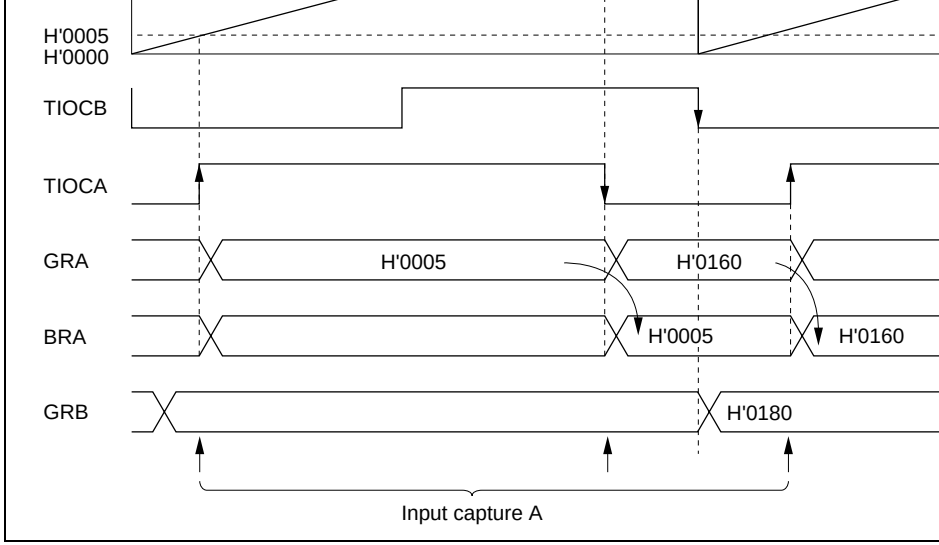


Figure 10.51 Register Buffering (Example 2: Buffering of Input Capture Re

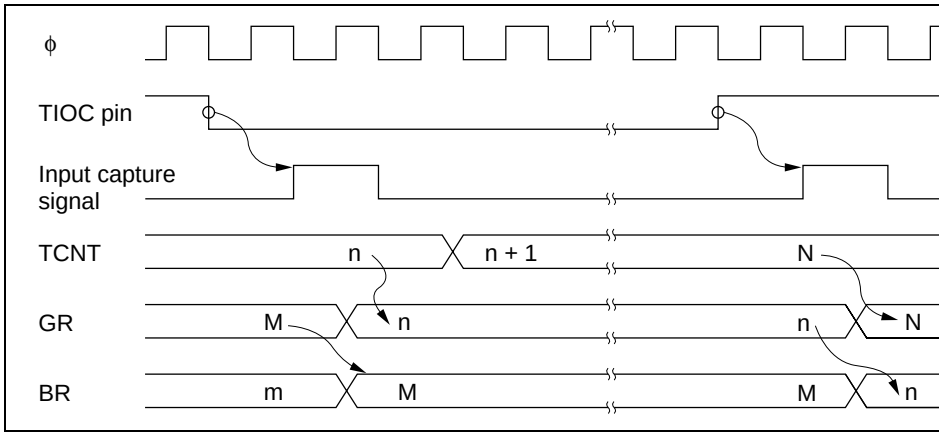


Figure 10.52 Input Capture and Buffer Transfer Timing (Example)

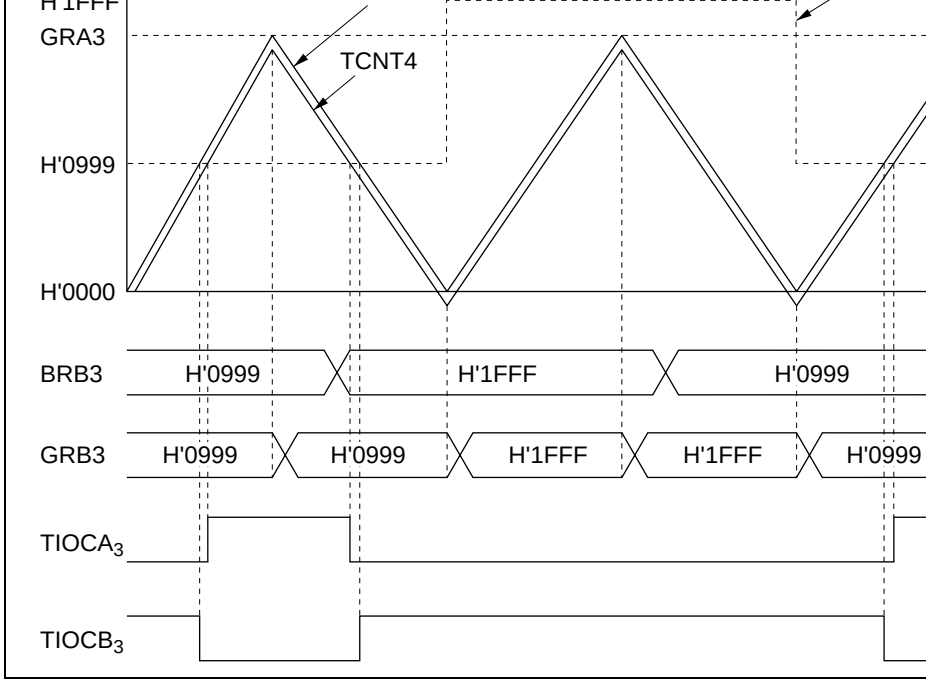


Figure 10.53 Register Buffering (Example 3: Buffering in Complementary PW

arbitrary value can be output by appropriate settings of the data register (DR) and data
register (DDR) of the corresponding input/output port. Figure 10.54 illustrates the timi
enabling and disabling of ITU output by TOER.

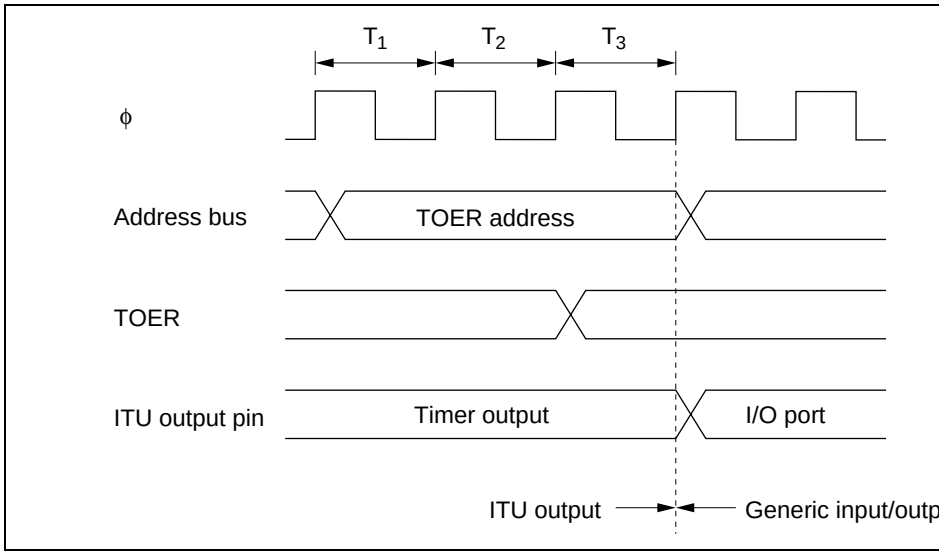
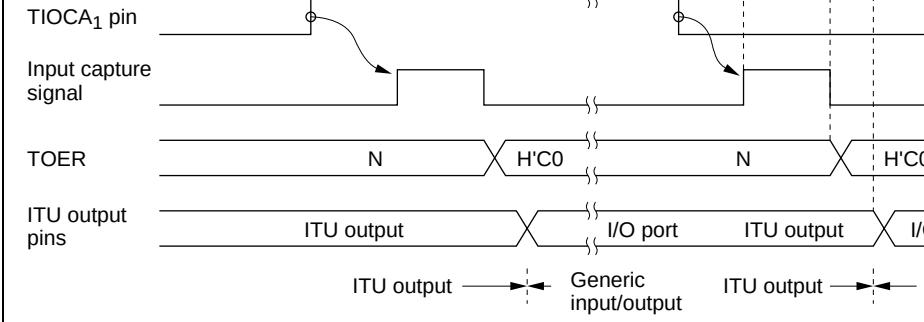


Figure 10.54 Timing of Disabling of ITU Output by Writing to TOER (Exa



N: Arbitrary setting (H'C1 to H'FF)

Figure 10.55 Timing of Disabling of ITU Output by External Trigger (Exa

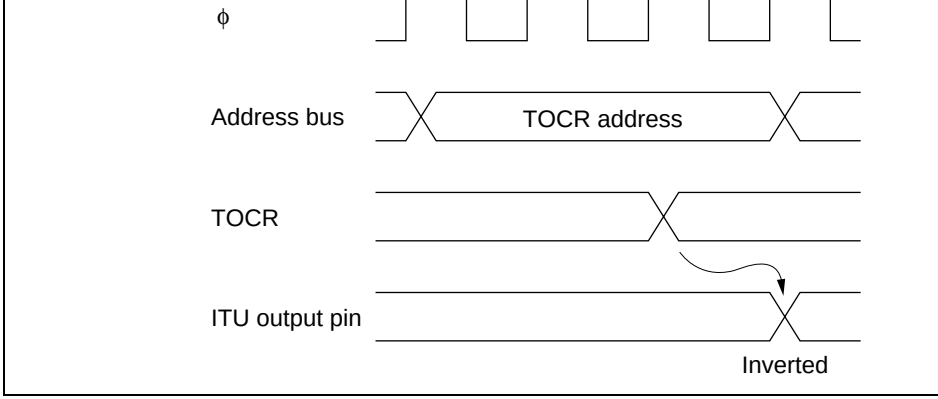


Figure 10.56 Timing of Inverting of ITU Output Level by Writing to TOCR (E

IMFA and IMFB are set to 1 by a compare match signal generated when TCNT match
register (GR). The compare match signal is generated in the last state in which the val
(when TCNT is updated from the matching count to the next count). Therefore, when
matches a general register, the compare match signal is not generated until the next tir
input. Figure 10.57 shows the timing of the setting of IMFA and IMFB.

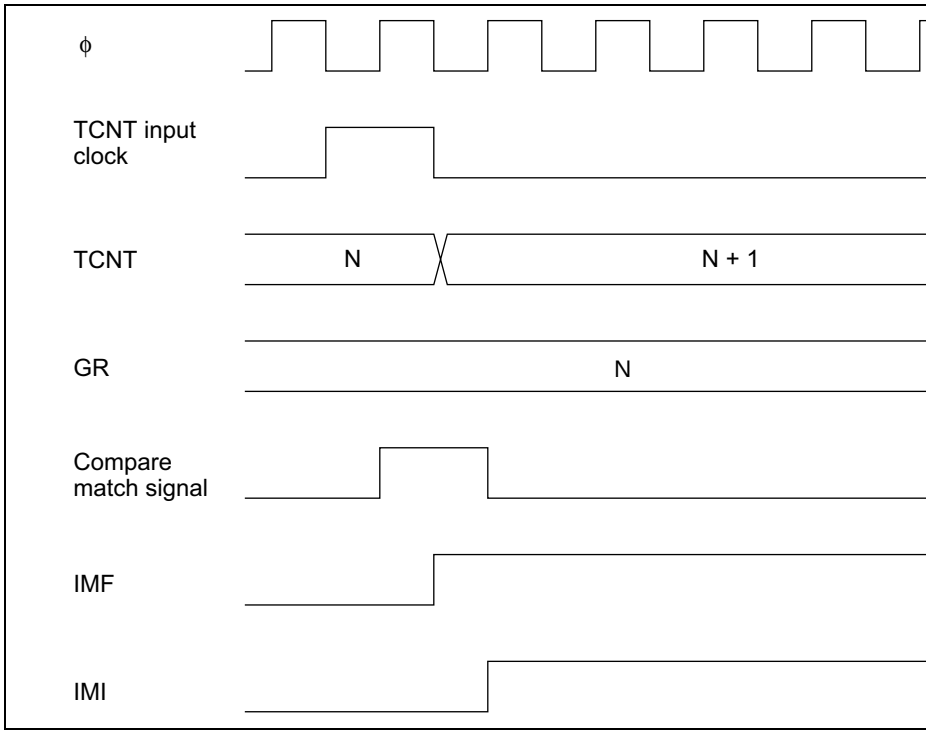


Figure 10.57 Timing of Setting of IMFA and IMFB by Compare Mat

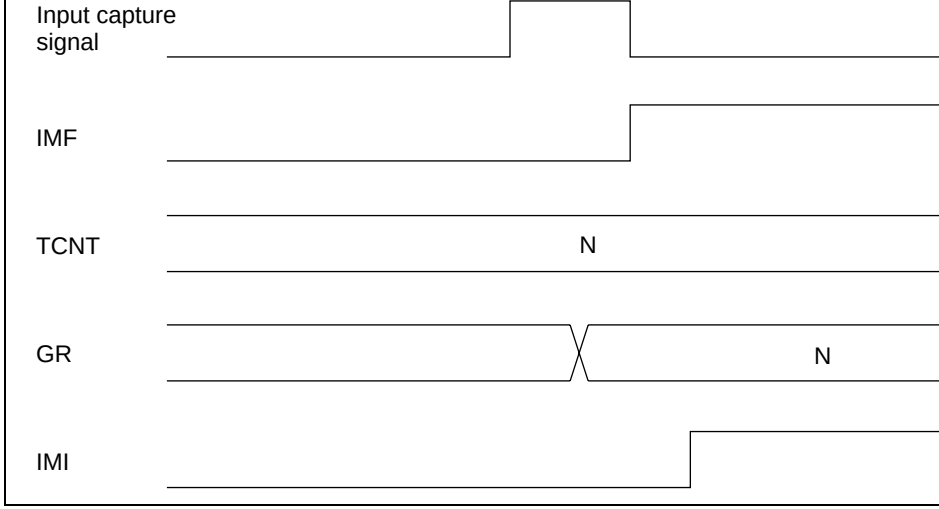


Figure 10.58 Timing of Setting of IMFA and IMFB by Input Capture

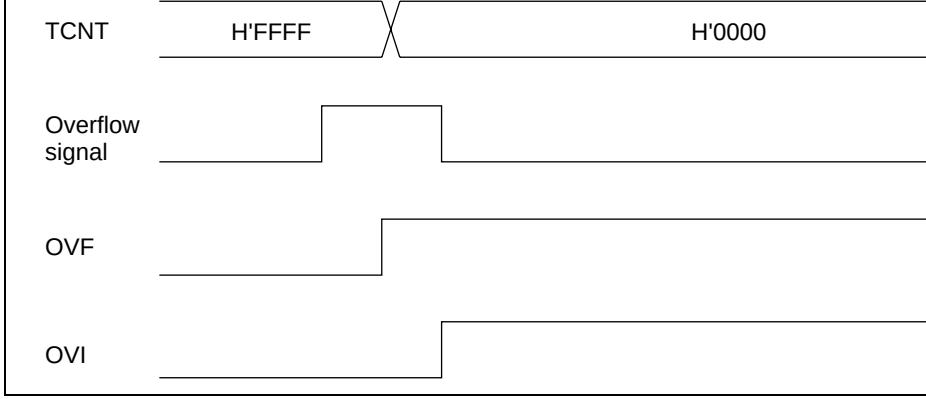


Figure 10.59 Timing of Setting of OVF

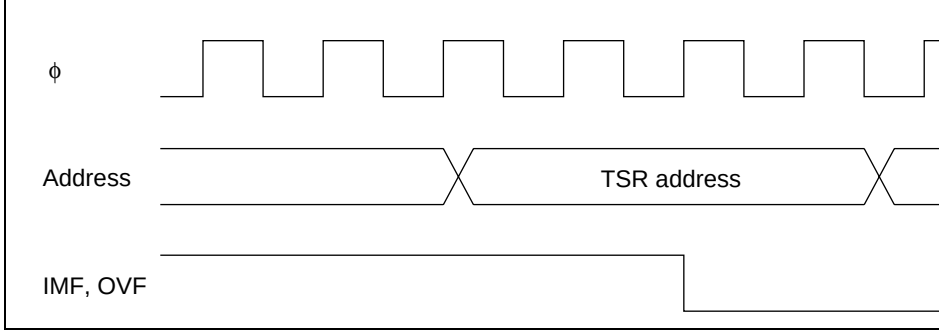


Figure 10.60 Timing of Clearing of Status Flags

10.5.3 Interrupt Sources and DMA Controller Activation

Each ITU channel can generate a compare match/input capture A interrupt, a compare
capture B interrupt, and an overflow interrupt. In total there are 15 interrupt sources, al
independently vectored. An interrupt is requested when the interrupt request flag and in
enable bit are both set to 1.

The priority order of the channels can be modified in interrupt priority registers A and
and IPRB). For details see section 5, Interrupt Controller.

Compare match/input capture A interrupts in channels 0 to 3 can activate the DMA con
(DMAC). When the DMAC is activated a CPU interrupt is not requested.

Table 10.10 lists the interrupt sources.

	IMIB1	Compare match/input capture B1	No
	OVI1	Overflow 1	No
2	IMIA2	Compare match/input capture A2	Yes
	IMIB2	Compare match/input capture B2	No
	OVI2	Overflow 2	No
3	IMIA3	Compare match/input capture A3	Yes
	IMIB3	Compare match/input capture B3	No
	OVI3	Overflow 3	No
4	IMIA4	Compare match/input capture A4	No
	IMIB4	Compare match/input capture B4	No
	OVI4	Overflow 4	No

Note: * The priority immediately after a reset is indicated. Inter-channel priorities ca changed by settings in IPRA and IPRB.

priority and the write is not performed. See figure 10.61.

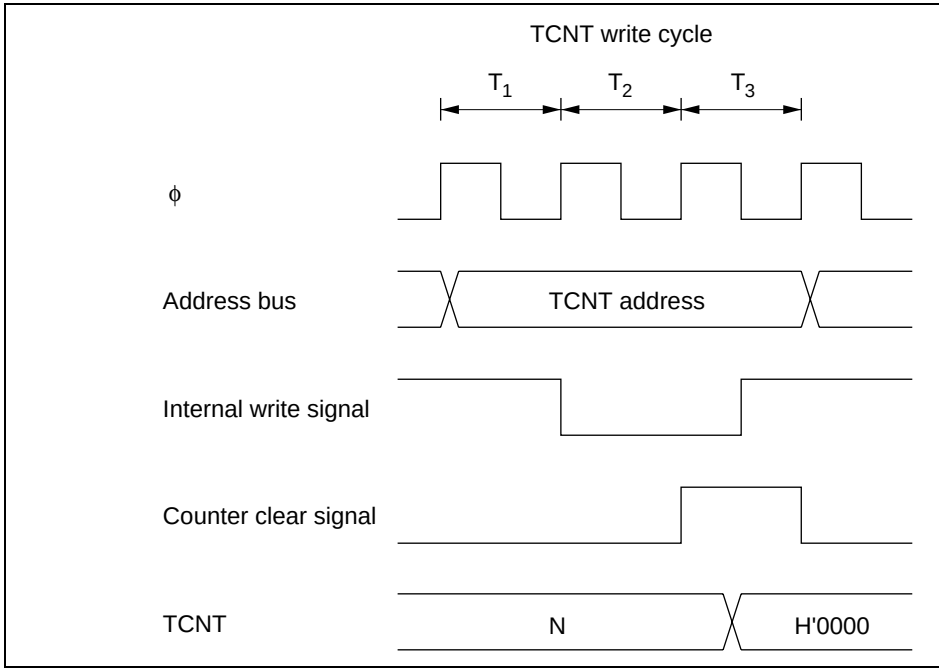


Figure 10.61 Contention between TCNT Write and Clear

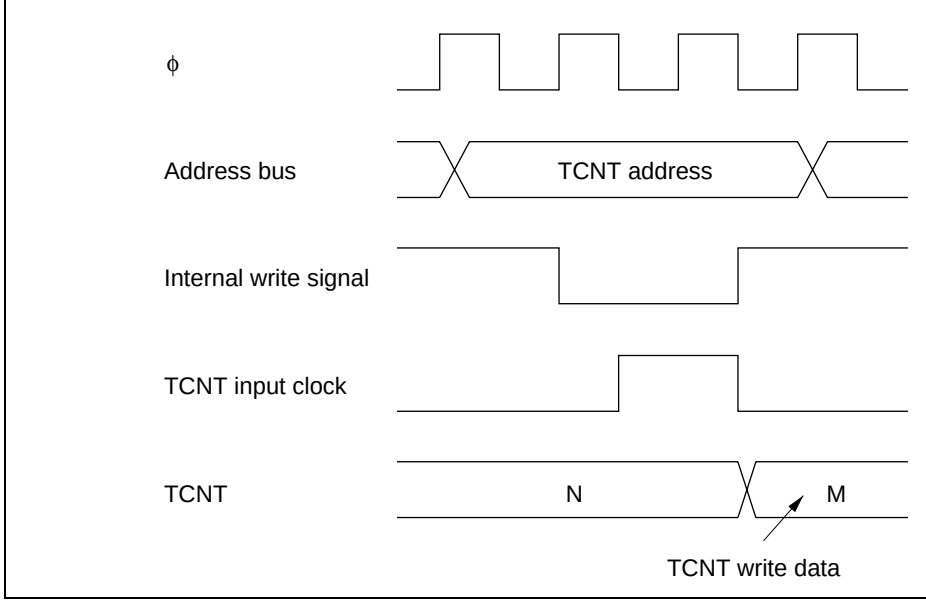


Figure 10.62 Contention between TCNT Word Write and Incremen

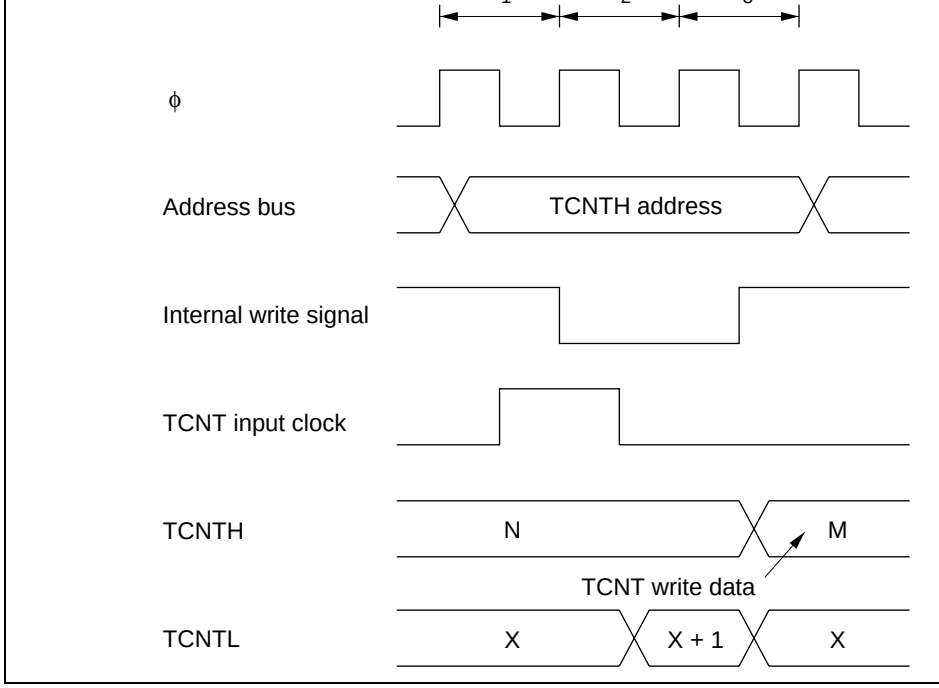


Figure 10.63 Contention between TCNT Byte Write and Increment

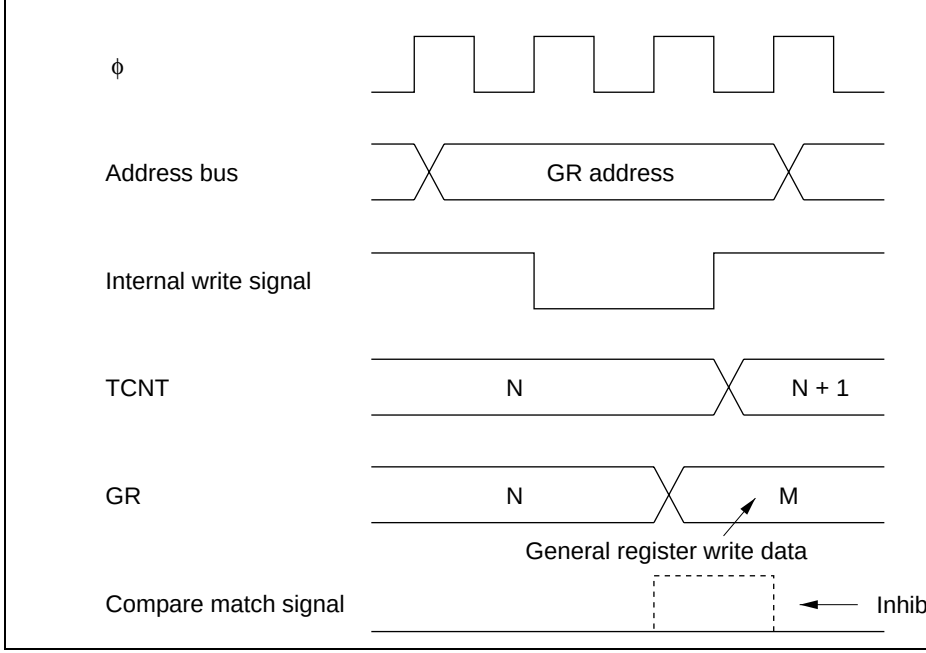


Figure 10.64 Contention between General Register Write and Compare M

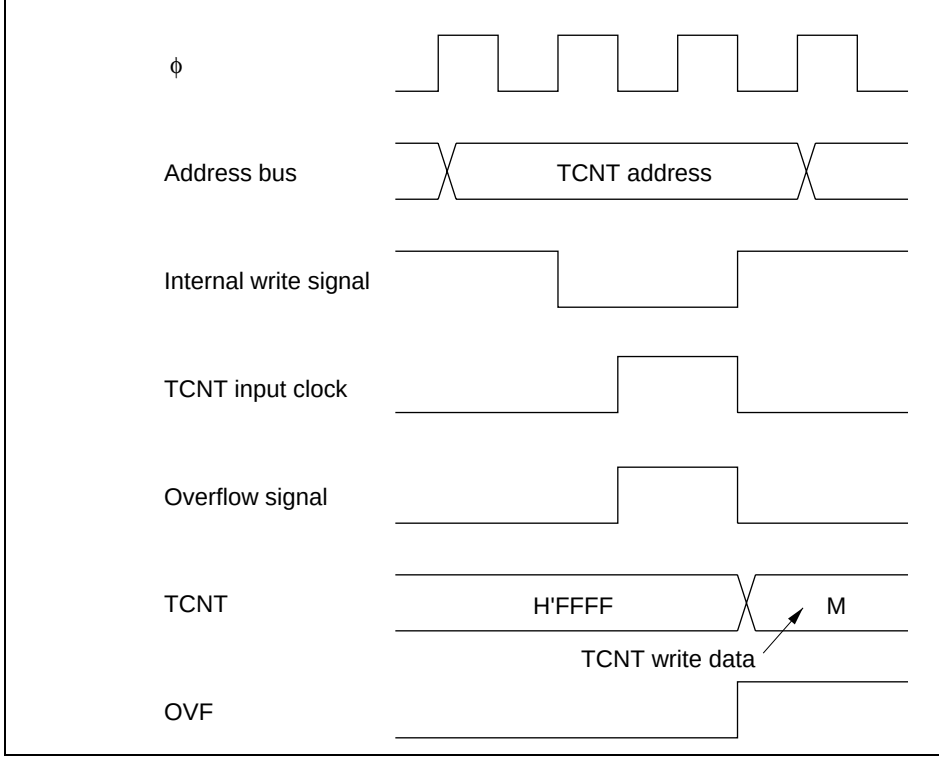


Figure 10.65 Contention between TCNT Write and Overflow

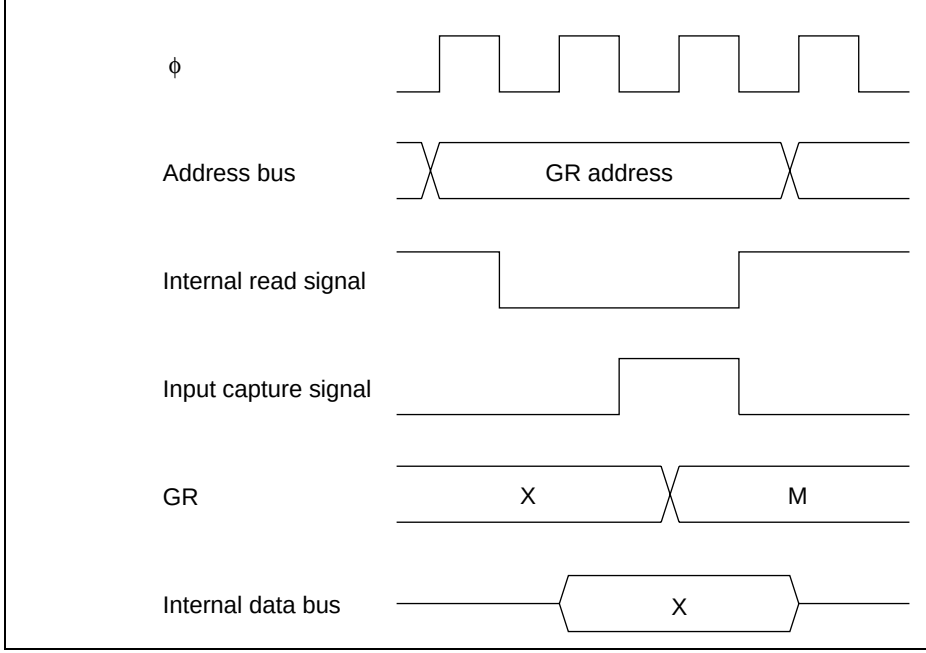


Figure 10.66 Contention between General Register Read and Input Cap

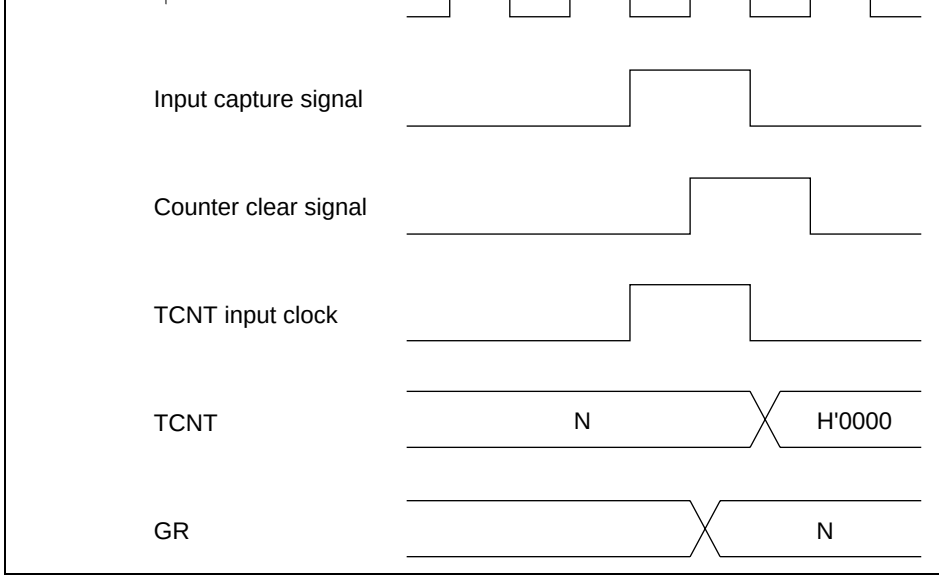


Figure 10.67 Contention between Counter Clearing by Input Capture and C Increment

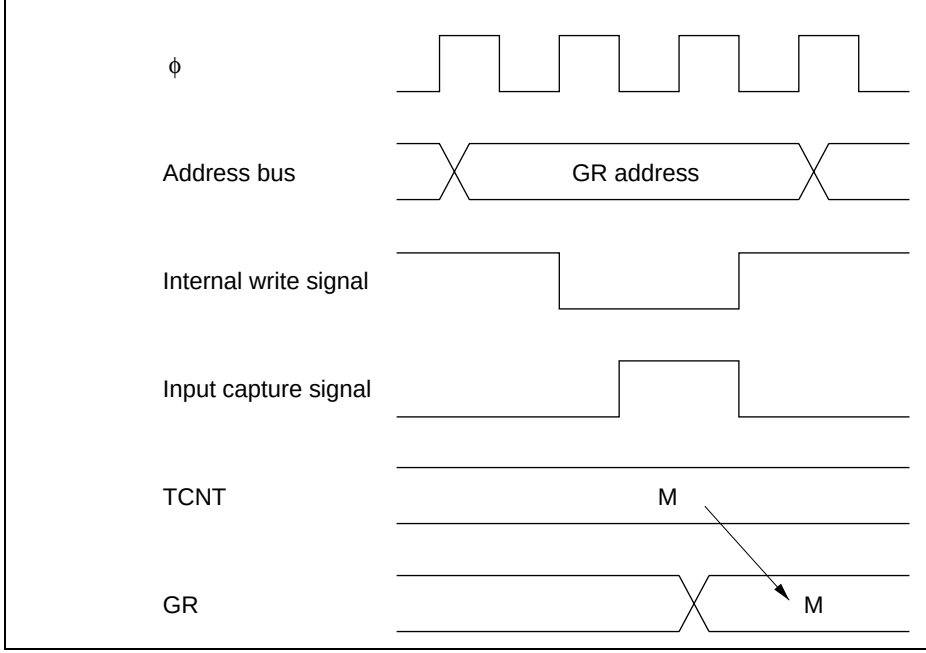


Figure 10.68 Contention between General Register Write and Input Cap

Note on Waveform Period Setting

When a counter is cleared by compare match, the counter is cleared in the last state at TCNT value matches the general register value, at the time when this value would nor updated to the next count. The actual counter frequency is therefore given by the follo formula:

$$f = \frac{\phi}{(N + 1)}$$

(f: counter frequency. φ: system clock frequency. N: value set in general regis

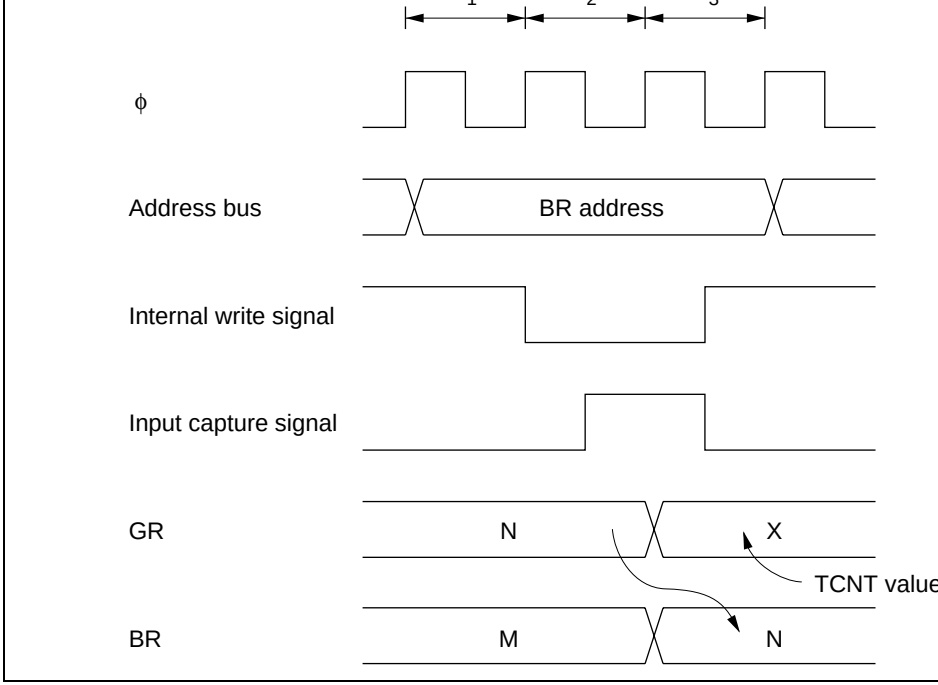
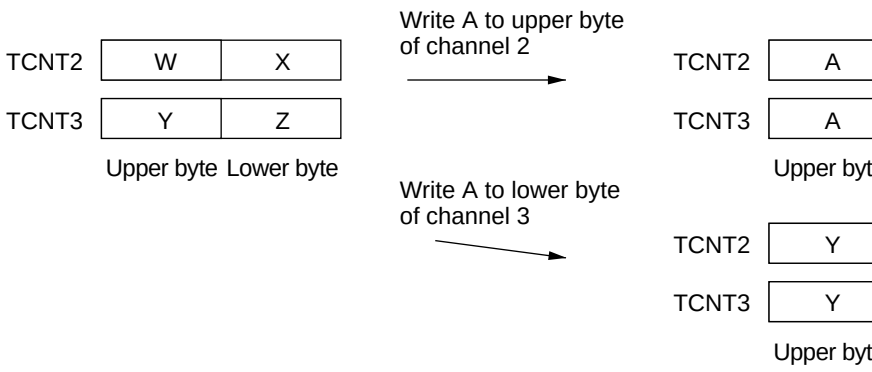
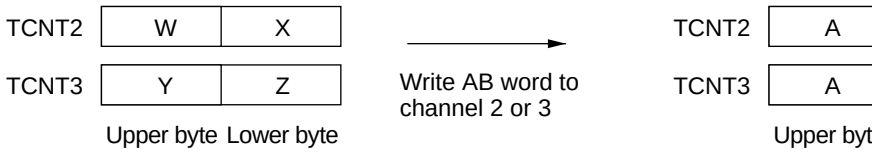


Figure 10.69 Contention between Buffer Register Write and Input Captu



- Word write to channel 2 or word write to channel 3



Note on Setup of Reset-Synchronized PWM Mode and Complementary PWM M

When setting bits CMD1 and CMD0 in TFCR, take the following precautions:

- Write to bits CMD1 and CMD0 only when TCNT3 and TCNT4 are stopped.
- Do not switch directly between reset-synchronized PWM mode and complementar mode. First switch to normal mode (by clearing bit CMD1 to 0), then select reset-s PWM mode or complementary PWM mode.

Operating Mode		Register Settings											
		TSNC	TMDR			TFCR			TOCR		TOER	TIOR0	
		Synchro-nization	MDF	FDIR	PWM	Comple-mentary PWM	Reset-Synchro-nized PWM	Buffer-ing	XTGD	Output Level Select	Master Enable	IOA	IOB
Synchronous preset		SYNC0 = 1	—	—	○	—	—	—	—	—	—	○	○
PWM mode		○	—	—	PWM0 = 1	—	—	—	—	—	—	—	○*
Output compare A		○	—	—	PWM0 = 0	—	—	—	—	—	—	IOA2 = 0 Other bits unrestricted	○
Output compare B		○	—	—	○	—	—	—	—	—	—	○	IOB2 = 0 Other bits unrestricte
Input capture A		○	—	—	PWM0 = 0	—	—	—	—	—	—	IOA2 = 1 Other bits unrestricted	○
Input capture B		○	—	—	PWM0 = 0	—	—	—	—	—	—	○	IOB2 = 1 Other bits unrestricte
Counter clearing	By compare match/input capture A	○	—	—	○	—	—	—	—	—	—	○	○
	By compare match/input capture B	○	—	—	○	—	—	—	—	—	—	○	○
	Syn-chronous clear	SYNC0 = 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 match B occur simultaneously, the compare ma

Operating Mode		Register Settings											
		TSNC	TMDR			TFCR			TOCR		TOER	TIOR1	
		Synchro-nization	MDF	FDIR	PWM	Comple-mentary PWM	Reset-Synchro-nized PWM	Buffer-ing	XTGD	Output Level Select	Master Enable	IOA	
Synchronous preset		SYNC1 = 1	—	—	○	—	—	—	—	—	—	○	
PWM mode		○	—	—	PWM1 = 1	—	—	—	—	—	—	—	
Output compare A		○	—	—	PWM1 = 0	—	—	—	—	—	—	IOA2 = 0 Other bits unrestricted	
Output compare B		○	—	—	○	—	—	—	—	—	—	○	IOB Oth unr
Input capture A		○	—	—	PWM1 = 0	—	—	—	○*2	—	—	IOA2 = 1 Other bits unrestricted	
Input capture B		○	—	—	PWM1 = 0	—	—	—	—	—	—	○	IOB Oth unr
Counter clearing	By compare match/input capture A	○	—	—	○	—	—	—	—	—	—	○	
	By compare match/input capture B	○	—	—	○	—	—	—	—	—	—	○	
	Syn-chronous clear	SYNC1 = 1	—	—	○	—	—	—	—	—	—	○	

Legend: ○: Setting available (valid).
—: Setting does not affect this mode.

Notes: 1. The input capture function cannot be used in PWM mode. If compare match A and compare match B occur simultaneously, the com
2. Valid only when channels 3 and 4 are operating in complementary PWM mode or reset-synchronized PWM mode.

Operating Mode		Register Settings												
		TSNC	TMDR			TFCR			TOCR			TOER	TIOR2	
		Synchro-nization	MDF	FDIR	PWM	Comple-mentary PWM	Reset-Synchro-nized PWM	Buffer-ing	XTGD	Output Level Select	Master Enable	IOA		
Synchronous preset		SYNC2 = 1	—	—	○	—	—	—	—	—	—	○		
PWM mode		○	—	—	PWM2 = 1	—	—	—	—	—	—	—		
Output compare A		○	—	—	PWM2 = 0	—	—	—	—	—	—	IOA2 = 0 Other bits unrestricted		
Output compare B		○	—	—	○	—	—	—	—	—	—	○	IOB Oth unr	
Input capture A		○	—	—	PWM2 = 0	—	—	—	—	—	—	IOA2 = 1 Other bits unrestricted		
Input capture B		○	—	—	PWM2 = 0	—	—	—	—	—	—	○	IOB Oth unr	
Counter clearing	By compare match/input capture A	○	—	—	○	—	—	—	—	—	—	○		
	By compare match/input capture B	○	—	—	○	—	—	—	—	—	—	○		
	Syn-chronous clear	SYNC2 = 1	—	—	○	—	—	—	—	—	—	○		
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 match B occur simultaneously, the comp

Operating Mode		Register Settings												
		TSNC	TMDR			TFCR			TOCR		TOER	TIOR3		
		Synchro-nization	MDF	FDIR	PWM	Comple-mentary PWM	Reset-Synchro-nized PWM	Buffer-ing	XTGD	Output Level Select	Master Enable	IOA		
Synchronous preset		SYNC3 = 1	—	—	○	○*3	○	○	—	—	○*1	○		
PWM mode		○	—	—	PWM3 = 1	CMD1 = 0	CMD1 = 0	○	—	—	○	—		
Output compare A		○	—	—	PWM3 = 0	CMD1 = 0	CMD1 = 0	○	—	—	○	IOA2 = 0 Other bits unrestricted		
Output compare B		○	—	—	○	CMD1 = 0	CMD1 = 0	○	—	—	○	○	IO Ot un	
Input capture A		○	—	—	PWM3 = 0	CMD1 = 0	CMD1 = 0	○	—	—	EA3 ignored Other bits unrestricted	IOA2 = 1 Other bits unrestricted		
Input capture B		○	—	—	PWM3 = 0	CMD1 = 0	CMD1 = 0	○	—	—	EB3 ignored Other bits unrestricted	○	IO Ot un	
Counter clearing	By compare match/input capture A	○	—	—	○	Illegal setting: CMD1 = 1 CMD0 = 0	○*4	○	—	—	○*1	○		
	By compare match/input capture B	○	—	—	○	CMD1 = 0	CMD1 = 0	○	—	—	○*1	○		
	Syn-chronous clear	SYNC3 = 1	—	—	○	Illegal setting: CMD1 = 1 CMD0 = 0	○	○	—	—	○*1	○		
Complementary PWM mode		○*3	—	—	—	CMD1 = 1 CMD0 = 0	CMD1 = 1 CMD0 = 0	○	○*6	○	○	—		
Reset-synchronized PWM mode		○	—	—	—	CMD1 = 1 CMD0 = 1	CMD1 = 1 CMD0 = 1	○	○*6	○	○	—		
Buffering (BRA)		○	—	—	○	○	○	BFA3 = 1 Other bits unrestricted	—	—	○*1	○		
Buffering (BRB)		○	—	—	○	○	○	BFA3 = 1 Other bits unrestricted	—	—	○*1	○		

Legend: ○: Setting available (valid).
—: Setting does not affect this mode.

Operating Mode		Register Settings												
		TSNC	TMDR			TFCR			TOCR		TOER	TIOR4		
		Synchro-nization	MDF	FDIR	PWM	Comple-mentary PWM	Reset-Synchro-nized PWM	Buffer-ing	XTGD	Output Level Select	Master Enable	IOA		
Synchronous preset		SYNC4 = 1	—	—	○	○*3	○	○	—	—	○*1	○		
PWM mode		○	—	—	PWM4 = 1	CMD1 = 0	CMD1 = 0	○	—	—	○	—		
Output compare A		○	—	—	PWM4 = 0	CMD1 = 0	CMD1 = 0	○	—	—	○	IOA2 = 0 Other bits unrestricted		
Output compare B		○	—	—	○	CMD1 = 0	CMD1 = 0	○	—	—	○	○	IOB2 = 0 Other bits unrestricted	
Input capture A		○	—	—	PWM4 = 0	CMD1 = 0	CMD1 = 0	○	—	—	EA4 ignored Other bits unrestricted	IOA2 = 1 Other bits unrestricted		
Input capture B		○	—	—	PWM4 = 0	CMD1 = 0	CMD1 = 0	○	—	—	EB4 ignored Other bits unrestricted	○	IOB2 = 1 Other bits unrestricted	
Counter clearing	By compare match/input capture A	○	—	—	○	Illegal setting: CMD1 = 1 CMD0 = 0	○*4	○	—	—	○*1	○		
	By compare match/input capture B	○	—	—	○	Illegal setting: CMD1 = 1 CMD0 = 0	○*4	○	—	—	○*1	○		
	Syn-chronous clear	SYNC4 = 1	—	—	○	Illegal setting: CMD1 = 1 CMD0 = 0	○*4	○	—	—	○*1	○		
Complementary PWM mode		○*3	—	—	—	CMD1 = 1 CMD0 = 0	CMD1 = 1 CMD0 = 0	○	○	○	○	—		
Reset-synchronized PWM mode		○	—	—	—	CMD1 = 1 CMD0 = 1	CMD1 = 1 CMD0 = 1	○	○	○	○	—		
Buffering (BRA)		○	—	—	○	○	○	BFA4 = 1 Other bits unrestricted	—	—	○*1	○		
Buffering (BRB)		○	—	—	○	○	○	BFA4 = 1 Other bits unrestricted	—	—	○*1	○		

Legend: ○: Setting available (valid).
—: Setting does not affect this mode.

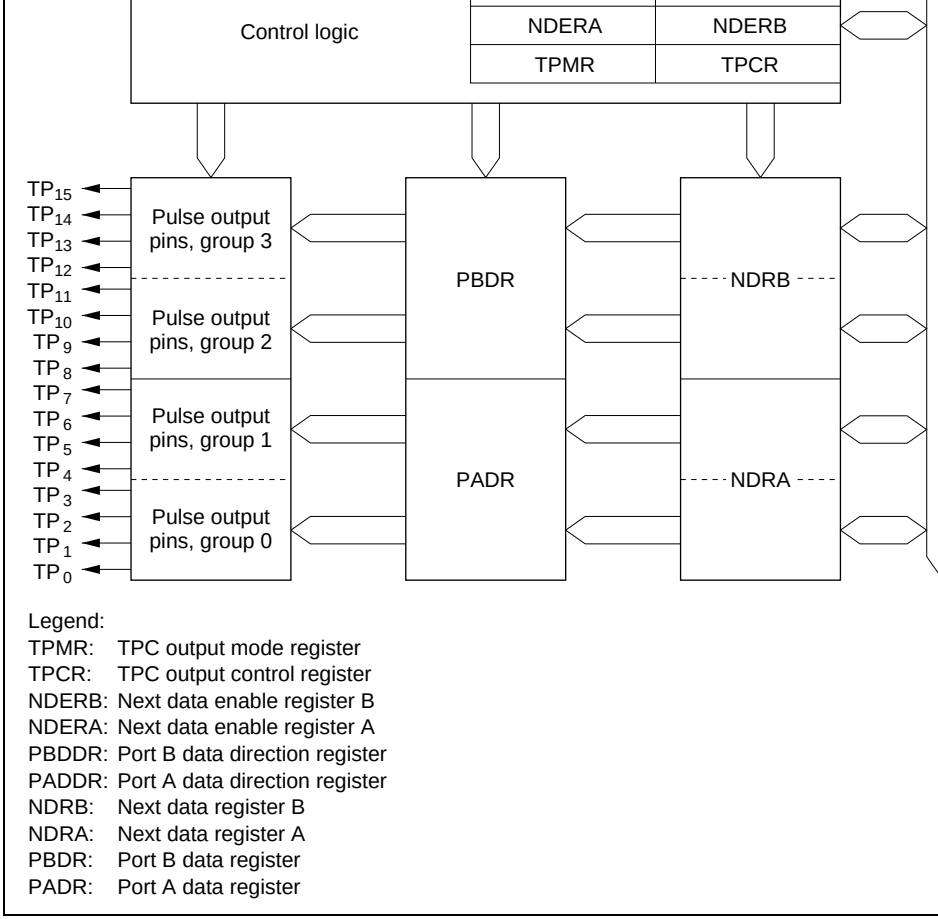
Notes: 1. Master enable bit settings are valid only during waveform output.

independently.

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
  ITU 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
  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 output
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 output
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 output
TPC output 13	TP_{13}	Output	
TPC output 14	TP_{14}	Output	
TPC output 15	TP_{15}	Output	

H'FFD4	Port B data direction register	PBDDR	W	H'00
H'FFD6	Port B data register	PBDR	R/(W)*2	H'00
H'FFA0	TPC output mode register	TPMR	R/W	H'F0
H'FFA1	TPC output control register	TPCR	R/W	H'FF
H'FFA2	Next data enable register B	NDERB	R/W	H'00
H'FFA3	Next data enable register A	NDERA	R/W	H'00
H'FFA5/ H'FFA7*3	Next data register A	NDRA	R/W	H'00
H'FFA4 H'FFA6*3	Next data register B	NDRB	R/W	H'00

Notes: 1. Lower 16 bits of the address.

2. Bits used for TPC output cannot be written.

3. The NDRA address is H'FFA5 when the same output trigger is selected for T groups 0 and 1 by settings in TPCR. When the output triggers are different, t address is H'FFA7 for group 0 and H'FFA5 for group 1. Similarly, the addres is H'FFA4 when the same output trigger is selected for TPC output groups 2 settings in TPCR. When the output triggers are different, the NDRB address for group 2 and H'FFA4 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 9.11, 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 9.11, Port A.

Port B data 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 9.12, 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.

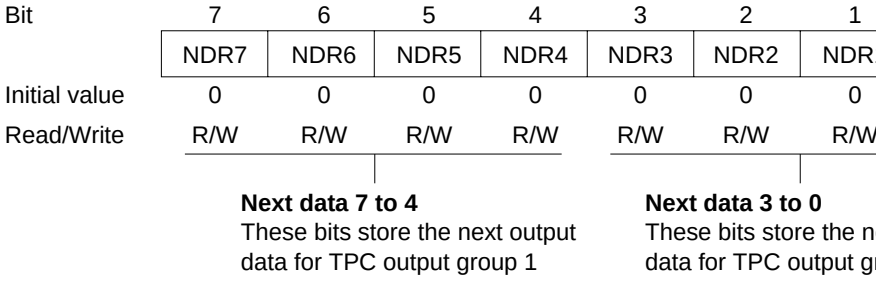
For further information about PBDR, see section 9.12, Port B.

software standby mode.

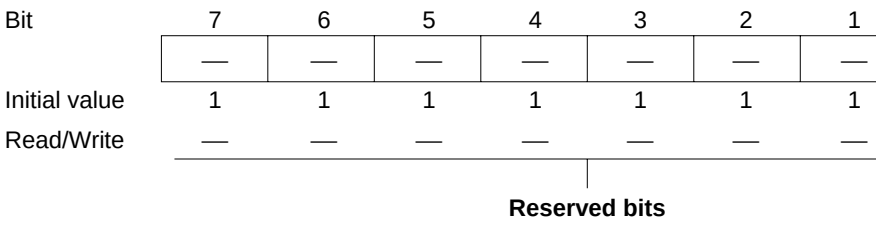
Same Trigger for TPC Output Groups 0 and 1

If TPC output groups 0 and 1 are triggered by the same compare match event, the ND
is H'FFA5. The upper 4 bits belong to group 1 and the lower 4 bits to group 0. Addres
consists entirely of reserved bits that cannot be modified and are always read as 1.

Address H'FFA5



Address H'FFA7



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	—	—	—

Next data 7 to 4
These bits store the next output data for TPC output group 1

Reserved bits

Address H'FFA7

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

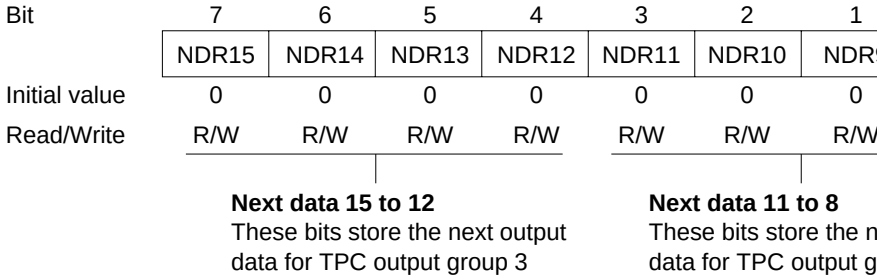
Next data 3 to 0
These bits store the ne
data for TPC output gr

software standby mode.

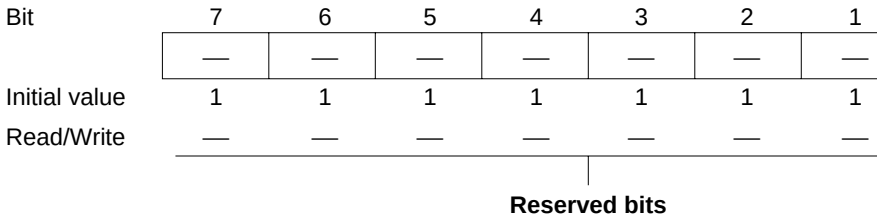
Same Trigger for TPC Output Groups 2 and 3

If TPC output groups 2 and 3 are triggered by the same compare match event, the ND
is H'FFA4. The upper 4 bits belong to group 3 and the lower 4 bits to group 2. Addres
consists entirely of reserved bits that cannot be modified and are always read as 1.

Address H'FFA4



Address H'FFA6



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	—	—	—

Next data 15 to 12
These bits store the next output data for TPC output group 3

Reserved bits

Address H'FFA6

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 data for TPC output

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 ITU compare match even the TPC output control register (TPCR) occurs, the NDRA value is automatically tran the corresponding PADR bit, updating the output value. If TPC output is disabled, the not transferred from NDRA to PADR and the output value does not change.

NDERA is initialized to H'00 by a reset and in hardware standby mode. It is not initial 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

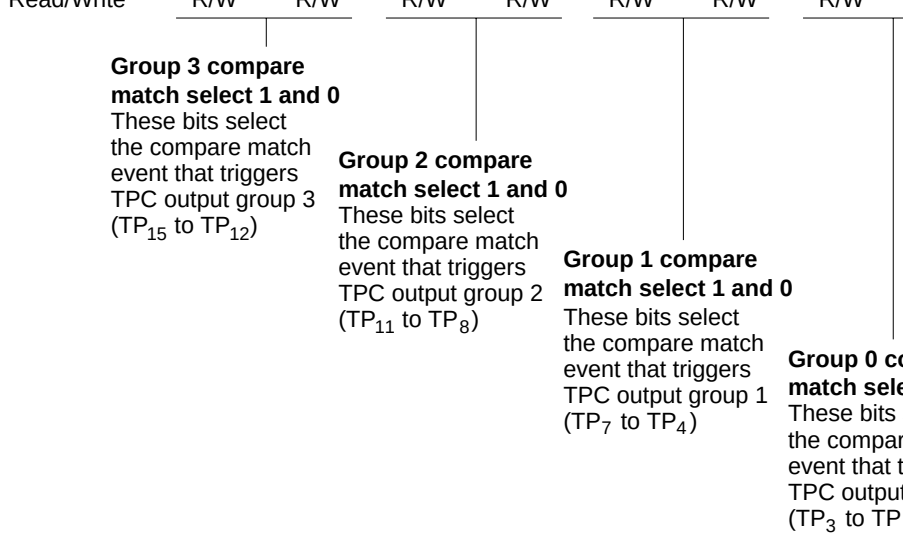
Next data enable 15 to 8
These bits enable or disable TPC output groups 3 and 2

If a bit is enabled for TPC output by NDERB, then when the ITU compare match event the TPC output control register (TPCR) occurs, the NDRB value is automatically transf corresponding PBDR bit, updating the output value. If TPC output is disabled, the bit v transferred from NDRB to PBDR and the output value does not change.

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 (I (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 match in ITU channel 0
	1	TPC output group 3 (TP_{15} to TP_{12}) is triggered match in ITU channel 1
1	0	TPC output group 3 (TP_{15} to TP_{12}) is triggered match in ITU channel 2
	1	TPC output group 3 (TP_{15} to TP_{12}) is triggered match in ITU channel 3

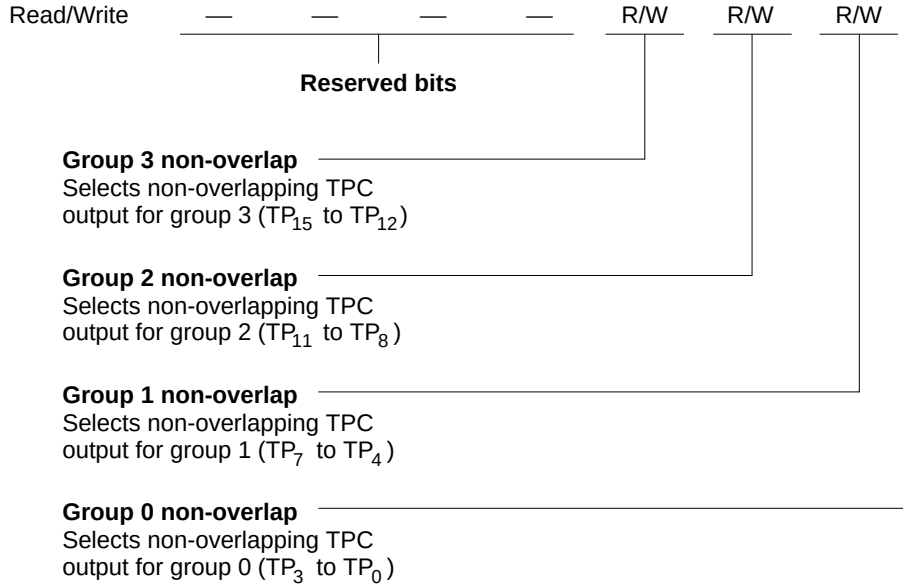
1	0	TPC output group 2 (TP_{11} to TP_8) is triggered by match in ITU channel 2
	1	TPC output group 2 (TP_{11} to TP_8) is triggered by match in ITU channel 3 (In

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 match in ITU channel 0
	1	TPC output group 1 (TP_7 to TP_4) is triggered by match in ITU channel 1
1	0	TPC output group 1 (TP_7 to TP_4) is triggered by match in ITU channel 2
	1	TPC output group 1 (TP_7 to TP_4) is triggered by match in ITU channel 3 (In

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 match in ITU channel 0
	1	TPC output group 0 (TP_3 to TP_0) is triggered by match in ITU channel 1
1	0	TPC output group 0 (TP_3 to TP_0) is triggered by match in ITU channel 2
	1	TPC output group 0 (TP_3 to TP_0) is triggered by match in ITU channel 3 (In



The output trigger period of a non-overlapping TPC output waveform is set in general (GRB) in the ITU channel selected for output triggering. The non-overlap margin is se register A (GRA). The output values change at compare match A and B. For details se 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: Read-only bits, 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 m the selected ITU channel) (In
1	Non-overlapping TPC output in group 2 (independent 1 and 0 outpu compare match A and B in the selected ITU 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 m the selected ITU channel) (In
1	Non-overlapping TPC output in group 1 (independent 1 and 0 outpu compare match A and B in the selected ITU 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 m the selected ITU channel) (In
1	Non-overlapping TPC output in group 0 (independent 1 and 0 outpu compare match A and B in the selected ITU channel)

Figure 11.2 illustrates the TPC output operation. Table 11.3 summarizes the TPC oper
conditions.

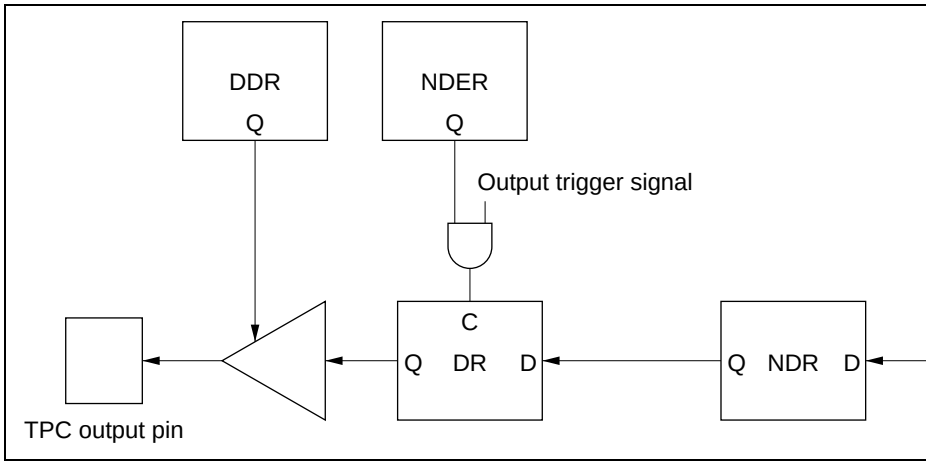


Figure 11.2 TPC Output Operation

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 when 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
11.3.4, Non-Overlapping TPC Output.

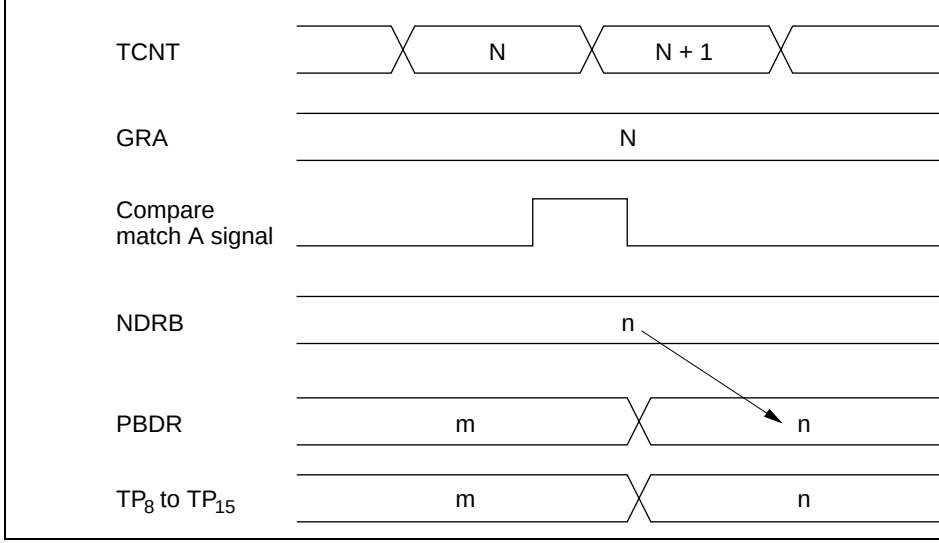
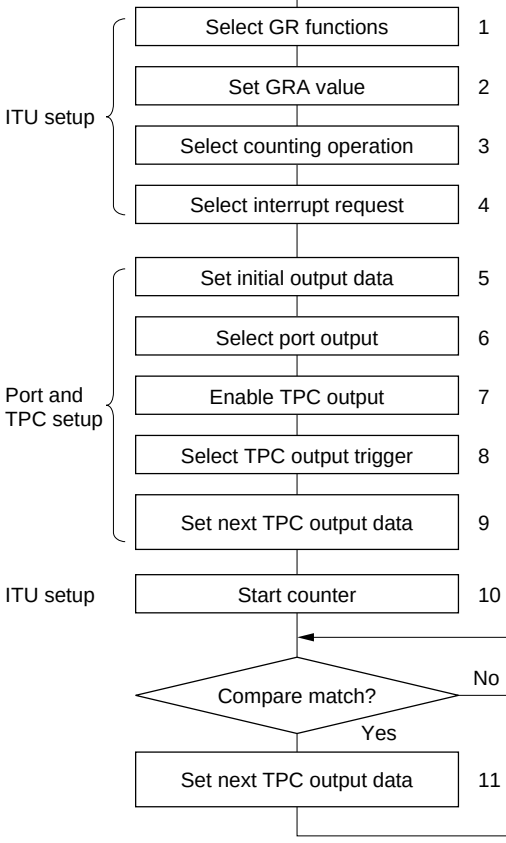
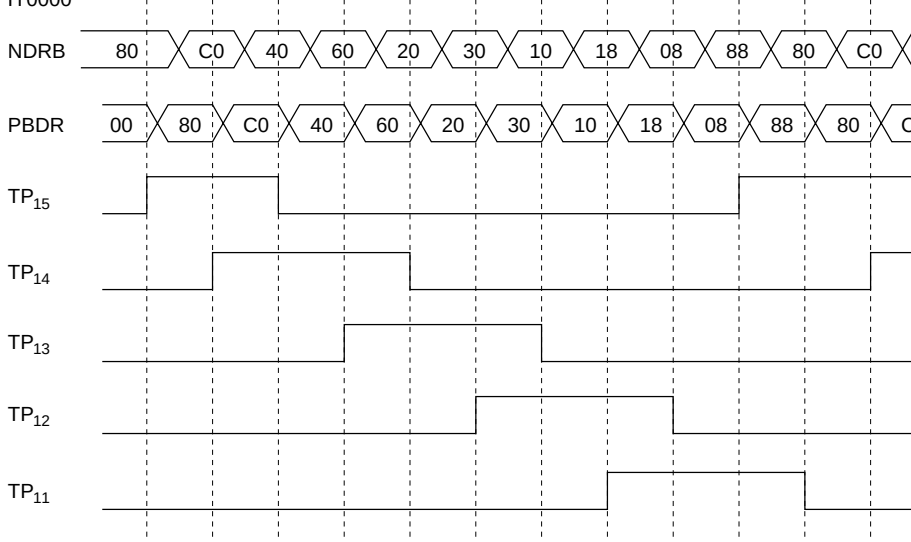


Figure 11.3 Timing of Transfer of Next Data Register Contents and Output (E



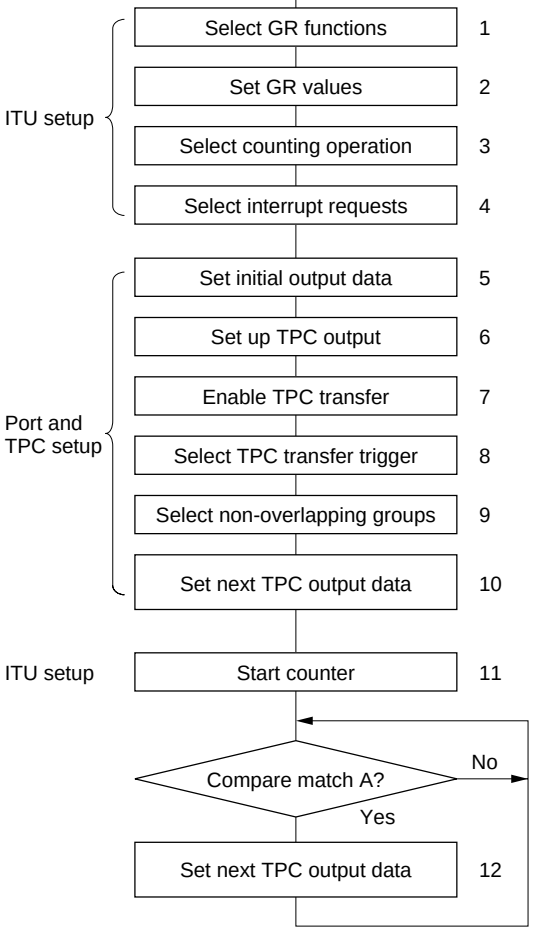
1. Set TIOR to make GRA an output
register (with output inhibited).
2. Set the TPC output trigger period.
3. Select the counter clock source wi
TPSC2 to TPSC0 in TCR. Select t
clear source with bits CCLR1 and
4. Enable the IMFA interrupt in TIER.
The DMAC can also be set up to t
data to the next data register.
5. Set the initial output values in the D
of the input/output port pins to be u
TPC output.
6. Set the DDR bits of the input/outpu
pins to be used for TPC output to 1
7. Set the NDER bits of the pins to be
TPC output to 1.
8. Select the ITU compare match eve
used as the TPC output trigger in T
9. Set the next TPC output values in t
10. Set the STR bit to 1 in TSTR to star
timer counter.
11. At each IMFA interrupt, set the next
values in the NDR bits.

Figure 11.4 Setup Procedure for Normal TPC Output (Example)



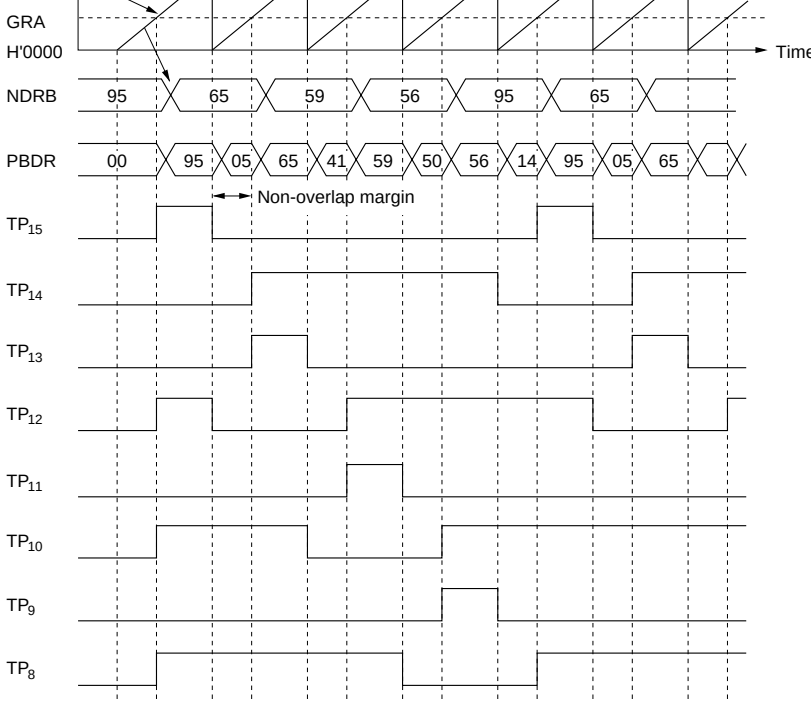
- The ITU channel to be used as the output trigger channel is set up so that GRA is an output com register and the counter will be cleared by compare match A. The trigger period is set in GRA. The IMIEA bit is set to 1 in TIER to enable the compare match A interrupt.
- H'F8 is written in PBDDR and NDERB, and bits G3CMS1, G3CMS0, G2CMS1, and G2CMS0 ar TPCR to select compare match in the ITU channel set up in step 1 as the output trigger. Output data H'80 is written in NDRB.
- The timer counter in this ITU channel is started. When compare match A occurs, the NDRB cont are transferred to PBDR and output. The compare match/input capture A (IMFA) interrupt servic 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 wri H'40, H'60, H'20, H'30, H'10, H'18, H'08, H'88… at successive IMFA interrupts. If the DMAC is se activation by this interrupt, pulse output can be obtained without loading the CPU.

Figure 11.5 Normal TPC Output Example (Five-Phase Pulse Output)



1. Set TIOR to make GRA and GRB
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 wi
TPSC2 to TPSC0 in TCR. Select t
clear source with bits CCLR1 and
4. Enable the IMFA interrupt in TIER
The DMAC can also be set up to t
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/outpu
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 ITU compare
event to be used as the TPC outpu
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 sta
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 output trigger ITU channel is set up so that GRA and GRB are output compare registers and the counter will be cleared by compare match B. The TPC output trigger period is set in GRB. The non-overlap margin is set in GRA. The IMIEA bit is set to 1 in TIER to enable IMFA interrupts.
- H'FF is written in PBDDR and NDERB, and bits G3CMS1, G3CMS0, G2CMS1, and G2CMS0 are set in TPCR to select compare match in the ITU channel set up in step 1 as the output trigger. Bits G3NOV and G2NOV are set to 1 in TPMR to select non-overlapping output. Output data H'95 is written in NDRB.
- The timer counter in this ITU channel is started. When compare match B occurs, outputs change from 1 to 0. When compare match A occurs, outputs change from 0 to 1 (the change from 0 to 1 is delayed by the value of GRA). The IMFA interrupt service routine writes the next output data (H'65) in NDRB.
- Four-phase complementary non-overlapping pulse output can be obtained by writing H'59, H'56, H'95 at successive IMFA interrupts. If the DMAC is set for activation by this interrupt, pulse output can be obtained without loading the CPU.

Figure 11.7 Non-Overlapping TPC Output Example (Four-Phase Complementary Non-Overlapping Pulse Output)

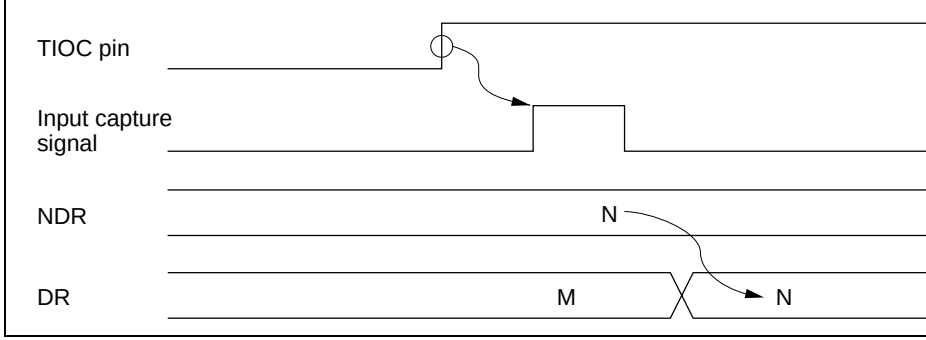


Figure 11.8 TPC Output Triggering by Input Capture (Example)

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.

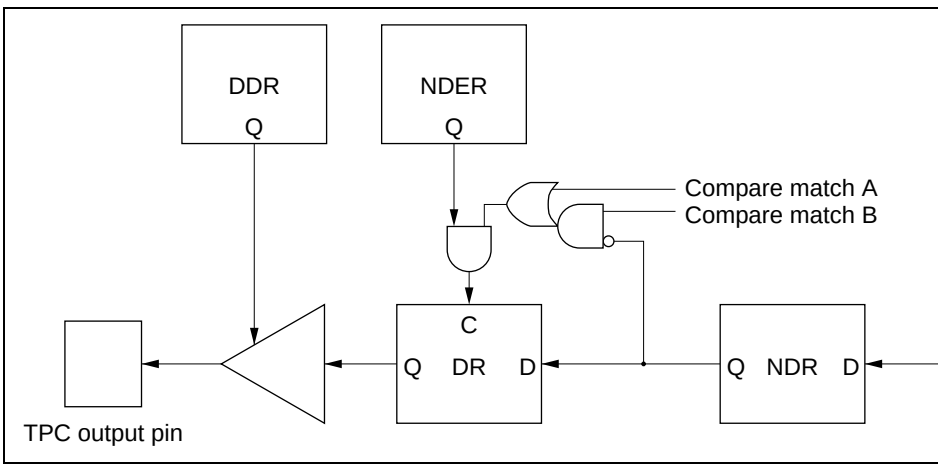


Figure 11.9 Non-Overlapping TPC Output

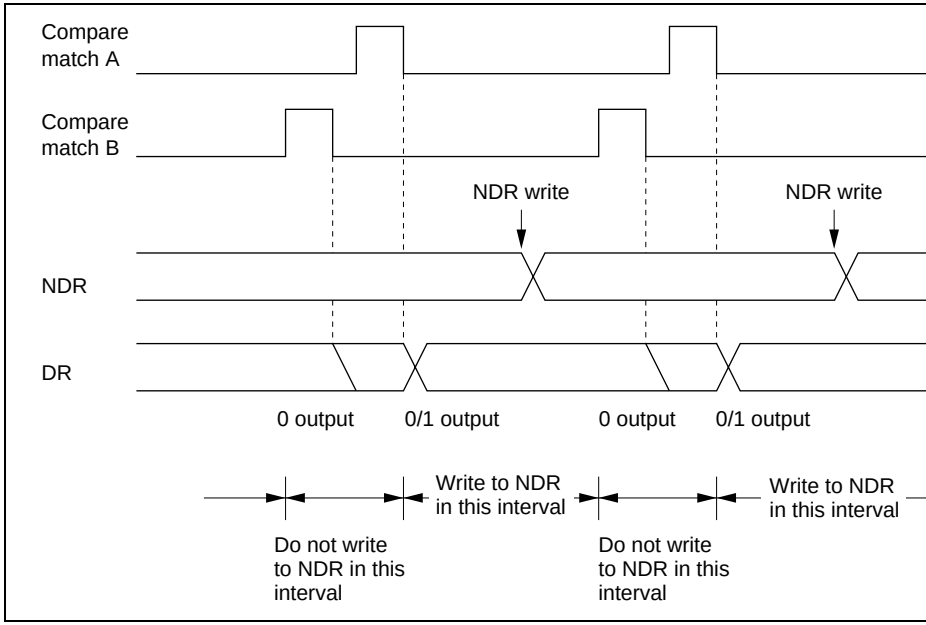


Figure 11.10 Non-Overlapping Operation and NDR Write Timing

the timer counter (TCNT) to overflow before being rewritten. In interval timer operati
interval timer interrupt is requested at each TCNT overflow.

12.1.1 Features

WDT features are listed below.

- Selection of eight counter clock sources
  $\phi/2$, $\phi/32$, $\phi/64$, $\phi/128$, $\phi/256$, $\phi/512$, $\phi/2048$, or $\phi/4096$
- 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.
- The entire chip can be reset internally by a reset signal output from the watchdog t
  The reset signal generated by timer counter overflow during watchdog timer opera
  the entire chip internally. In an H8/3048F-ONE (single power supply with flash m
  $\overline{\text{RESO}}$ pin acts as the FWE pin; no external reset signal can be output.

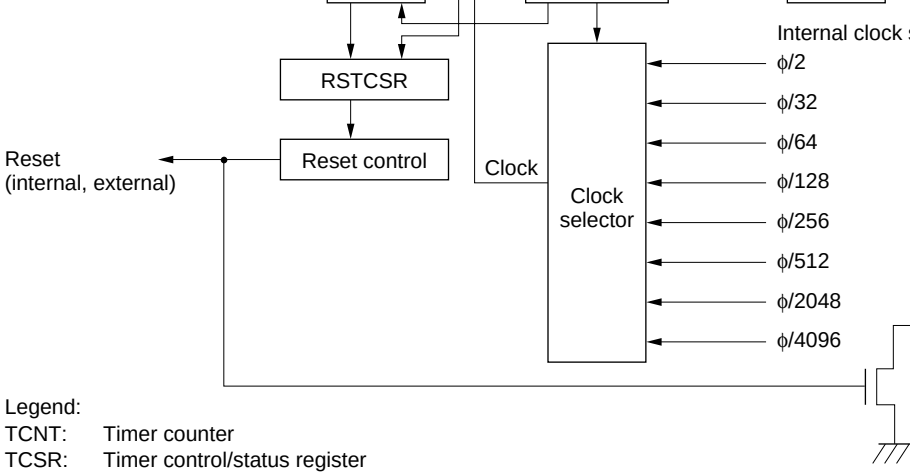


Figure 12.1 WDT Block Diagram

12.1.3 Pin Configuration

Output pins used by the WDT[*1] are shown in table 12.11.

Table 12.1 WDT Pins

Pin Name	Abbreviation	I/O	Function
Reset output	$\overline{\text{RESO}}$	Output[*2]	External output of watchdog timer re

Notes: 1. Not provided in on-chip flash memory versions.
2. Open-drain output pin

	H'FFA9	Timer counter	TCNT	R/W
H'FFAA	H'FFAB	Reset control/status register	RSTCSR	R/(W)*3

Notes: 1. Lower 16 bits of the address.
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

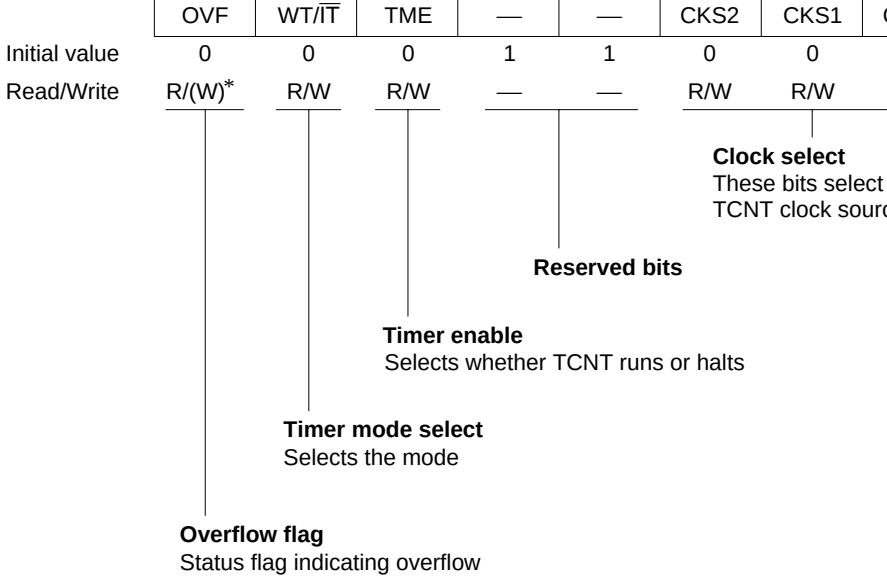
12.2.1 Timer Counter (TCNT)

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

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.

Note: * TCNT is write-protected by a password. For details see section 12.2.4, Not Register Rewriting.



Note: * 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 (In
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.

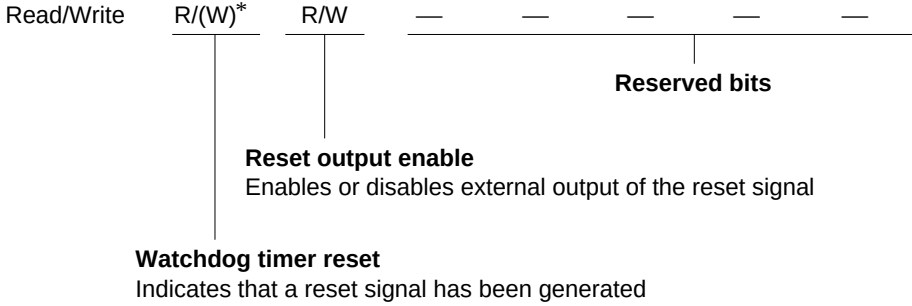
When WT/$\overline{IT}$ = 1, clear the SYSCR software standby bit (SSBY) to 0, then set the TM
When SSBY is set to 1, clear TME to 0.

Bit 5: TME	Description	
0	TCNT is initialized to H'00 and halted	(
1	TCNT is counting and CPU interrupt requests are enabled	

Bits 4 and 3—Reserved: Read-only bits, always read as 1.

Bits 2 to 0—Clock Select 2 to 0 (CKS2 to CKS0): These bits select one of eight inte
sources, 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 method for writing to RSTCSR is different from that for general registers t
inadvertent overwriting. For details see section 12.2.4, Notes on Register Rew
* Only 0 can be written in bit 7, to clear the flag.

Bits 7 and 6 are initialized by input of a reset signal at 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 ch
internally. If bit RSTOE is set to 1, this reset signal is also output (low) at the $\overline{\text{RESO}}$ pi
initialize external system devices. Note that there is no $\overline{\text{RESO}}$ pin in the versions with c
flash memory.

Bit 7 WRST	Description
0	[Clearing conditions] • Reset signal at $\overline{\text{RES}}$ pin. • Read WRST when WRST = 1, then write 0 in WRST. (In
1	[Setting condition] Set when TCNT overflow generates a reset signal during watchdog time

Bits 5 to 0—Reserved: These bits cannot be modified and are always read as 1.

12.2.4 Notes on Register Rewriting

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 instruction. They cannot be written
instructions. Figure 12.2 shows the format of data written to TCNT and TCSR. TCNT
both have the same write address. The write data must be contained in the lower byte
written word. The upper byte must contain H'5A (password for TCNT) or H'A5 (passw
TCSR). This transfers the write data from the lower byte to TCNT or TCSR.

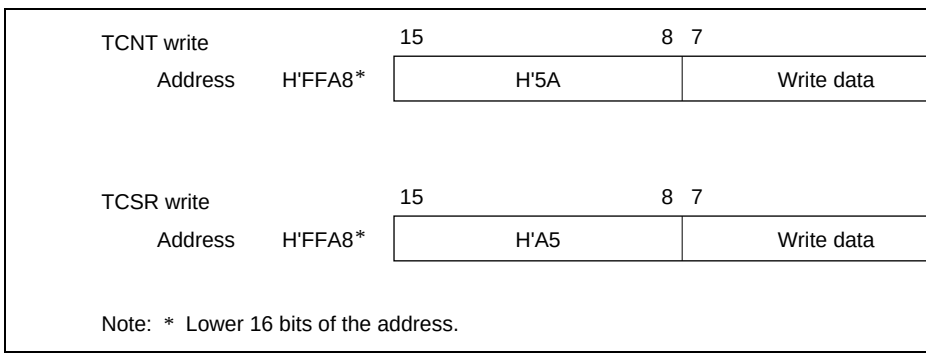


Figure 12.2 Format of Data Written to TCNT and TCSR

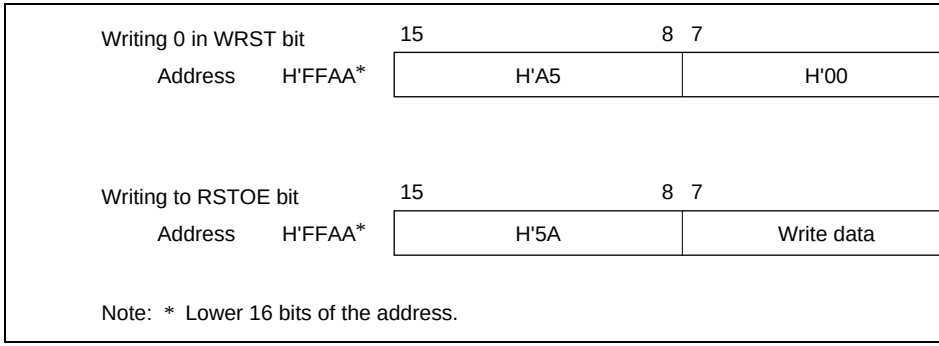


Figure 12.3 Format of Data Written to RSTCSR

Reading TCNT, TCSR, and RSTCSR

These registers are read like other registers. Byte access instructions can be used. The r addresses are H'FFA8 for TCSR, H'FFA9 for TCNT, and H'FFAB for RSTCSR, as list 12.3.

Table 12.3 Read Addresses of TCNT, TCSR, and RSTCSR

Address*	Register
H'FFA8	TCSR
H'FFA9	TCNT
H'FFAB	RSTCSR

Note: * Lower 16 bits of the address.

WT/IT and TME bits to 1 in TCSR. Software must prevent TCNT overflow by rewrit
TCNT value (normally by writing H'00) before overflow occurs. If TCNT fails to be r
overflows due to a system crash etc., the chip is internally reset for a duration of 518 s

The watchdog reset signal can be externally output from the $\overline{\text{RESO}}$ pin to reset externa
devices. The reset signal is output externally for 132 states. External output can be ena
disabled by the RSTOE bit in RSTCSR. Note that there is no $\overline{\text{RESO}}$ pin in the version
chip flash memory.

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

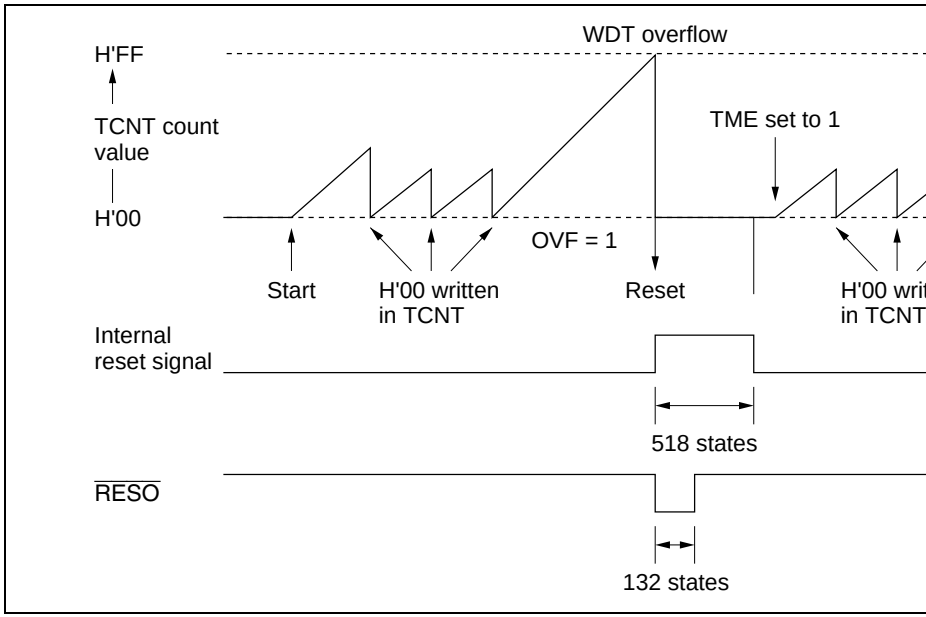


Figure 12.4 Watchdog Timer Operation

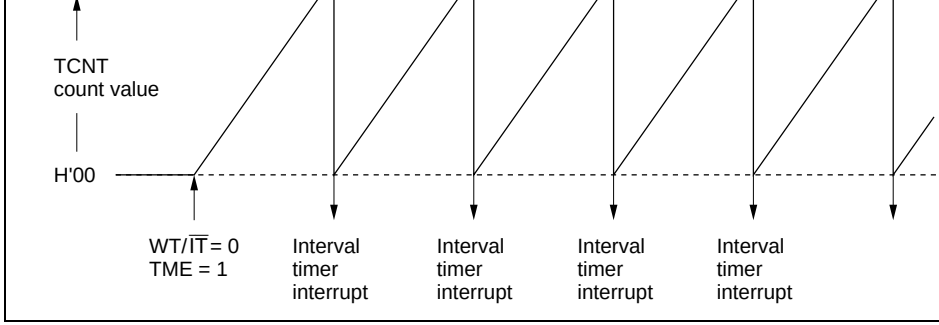


Figure 12.5 Interval Timer Operation

12.3.3 Timing of Setting of Overflow Flag (OVF)

Figure 12.6 shows the timing of setting of the OVF flag in TCSR. The OVF flag is set t
TCNT overflows. At the same time, a reset signal is generated in watchdog timer opera
interval timer interrupt is generated in interval timer operation.

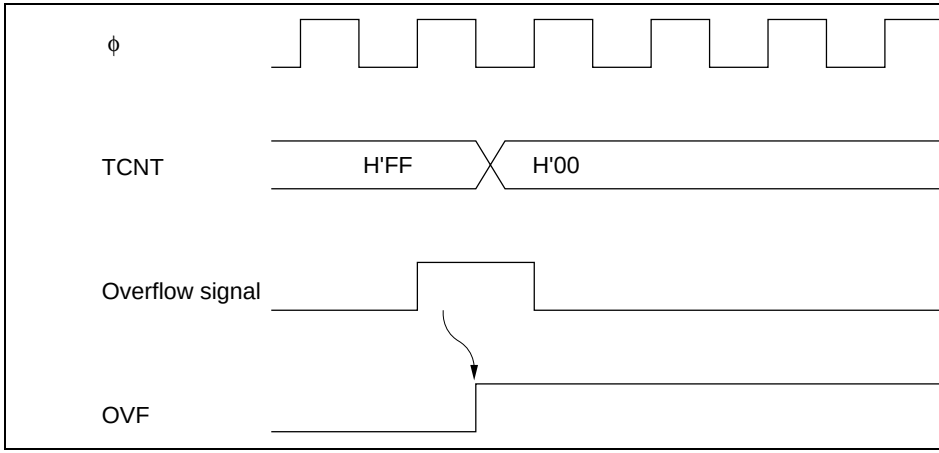


Figure 12.6 Timing of Setting of OVF

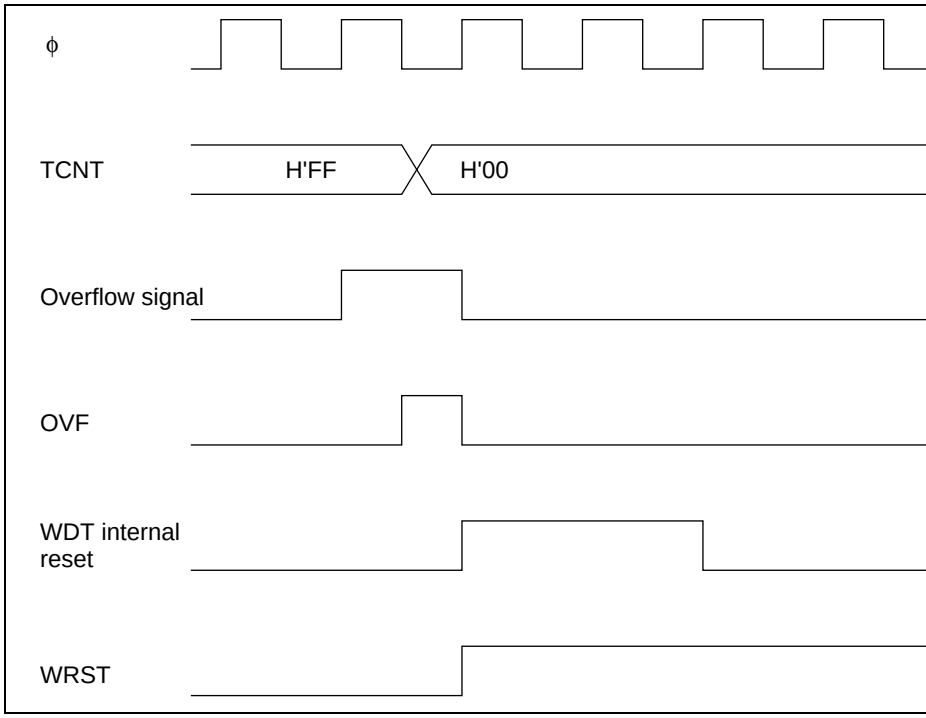


Figure 12.7 Timing of Setting of WRST Bit and Internal Reset

If a timer counter clock pulse is generated during the T_3 state of a write cycle to TCNT, takes priority and the timer count is not incremented. See figure 12.8.

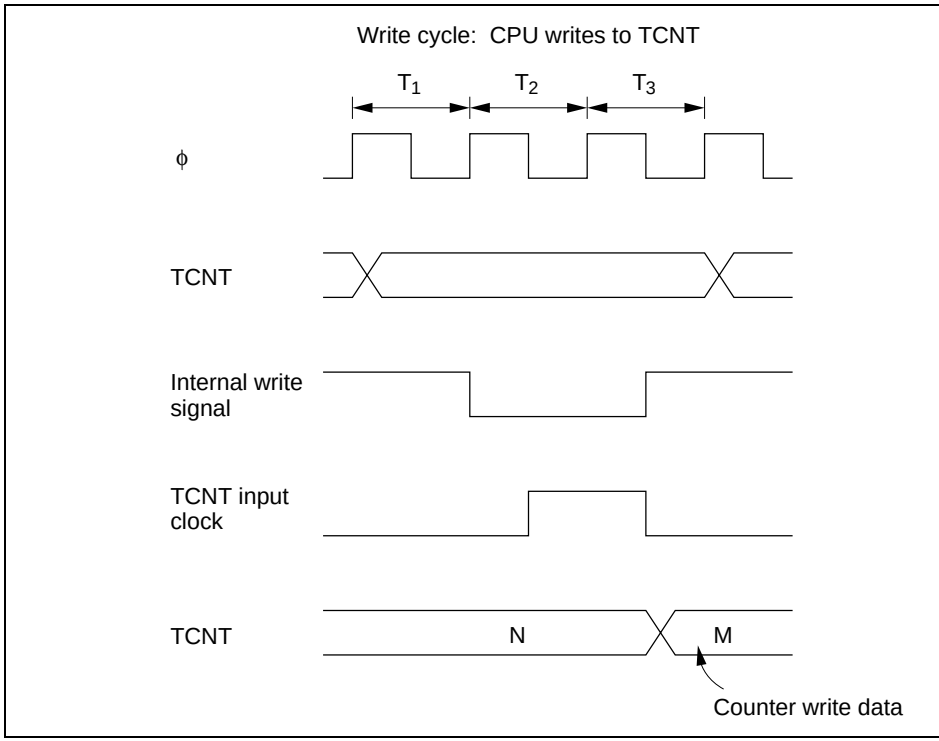


Figure 12.8 Contention between TCNT Write and Increment

Changing CKS2 to CKS0 Values

Halt TCNT by clearing the TME bit to 0 in TCSR before changing the values of bits CK CKS0.

- When the internal I/O registers related to the on-chip WDT are rewritten.
- When software standby mode is incorrectly entered.
- When the break mode is incorrectly entered.

In addition, as stated in the NMI above, if an abnormal level is input into the power su
the system control pins, correct operations cannot be guaranteed.

Except the above cases, the on-chip WDT functions as a device that supports recovery
system crash. Accordingly, when a fail-safe function is required in your system, an ad
circuit may be required as necessary.

among two or more processors.

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.

Channel 0 (SCI0) also has a smart card interface function conforming to the ISO/IEC7
(Identification Card) standard. This function supports serial communication with a sm
details, see section 14, Smart Card Interface.

13.1.1 Features

SCI features are listed below.

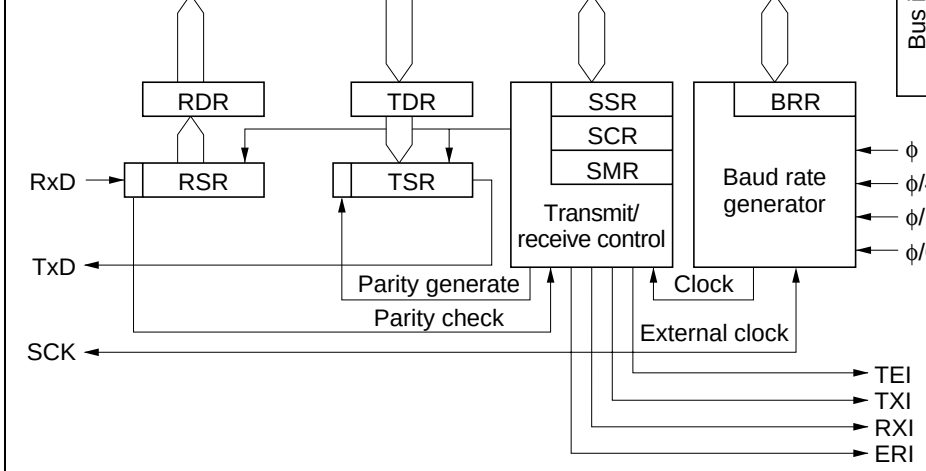
- Selection of asynchronous or synchronous mode for serial communication
  - Asynchronous mode

    Serial data communication is synchronized one character at a time. The SCI ca
    communicate with a universal asynchronous receiver/transmitter (UART), asyn
    communication interface adapter (ACIA), or other chip that employs standard
    asynchronous serial communication. It can also communicate with two or mor
    processors using the multiprocessor communication function. There are twelve
    serial data communication formats.
    - Data length: 7 or 8 bits
    - Stop bit length: 1 or 2 bits
    - Parity bit: even, odd, or 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 error oc

The transmitting and receiving sections are independent, so the SCI can transmit an simultaneously. The transmitting and receiving sections are both double-buffered, s data can be transmitted and received continuously.

- Built-in baud rate generator with selectable bit rates
- 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.



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

Figure 13.1 SCI Block Diagram

	Transmit data pin	TxD_0	Output	SCI_0 transmit da
1	Serial clock pin	SCK_1	Input/output	SCI_1 clock input/
	Receive data pin	RxD_1	Input	SCI_1 receive dat
	Transmit data pin	TxD_1	Output	SCI_1 transmit da

13.1.4 Register Configuration

The SCI has internal registers as listed in table 13.2. These registers select asynchronou synchronous mode, specify the data format and bit rate, and control the transmitter and sections.

Table 13.2 Registers

Channel	Address*1	Name	Abbreviation	R/W	In
0	H'FFB0	Serial mode register	SMR	R/W	H'
	H'FFB1	Bit rate register	BRR	R/W	H'
	H'FFB2	Serial control register	SCR	R/W	H'
	H'FFB3	Transmit data register	TDR	R/W	H'
	H'FFB4	Serial status register	SSR	R/(W)*2	H'
	H'FFB5	Receive data register	RDR	R	H'
1	H'FFB8	Serial mode register	SMR	R/W	H'
	H'FFB9	Bit rate register	BRR	R/W	H'
	H'FFBA	Serial control register	SCR	R/W	H'
	H'FFBB	Transmit data register	TDR	R/W	H'
	H'FFBC	Serial status register	SSR	R/(W)*2	H'
	H'FFBD	Receive data register	RDR	R	H'

Notes: 1. Lower 16 bits of the address.
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 (b
thereby converting the data to parallel data. When 1 byte has been received, it is auton
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 finishes receiving 1 byte of serial data, it transfers the received data fro
RDR for storage. RSR is then ready to receive the next data. This double buffering all
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.

The SCI loads transmit data from TDR into TSR, then transmits the data serially from t
pin, LSB (bit 0) first. After transmitting one data byte, the SCI automatically loads the
transmit data from TDR into TSR and starts transmitting it. If the TDRE flag is set to 1
however, the SCI does not load the TDR contents into TSR. The CPU cannot read or w
directly.

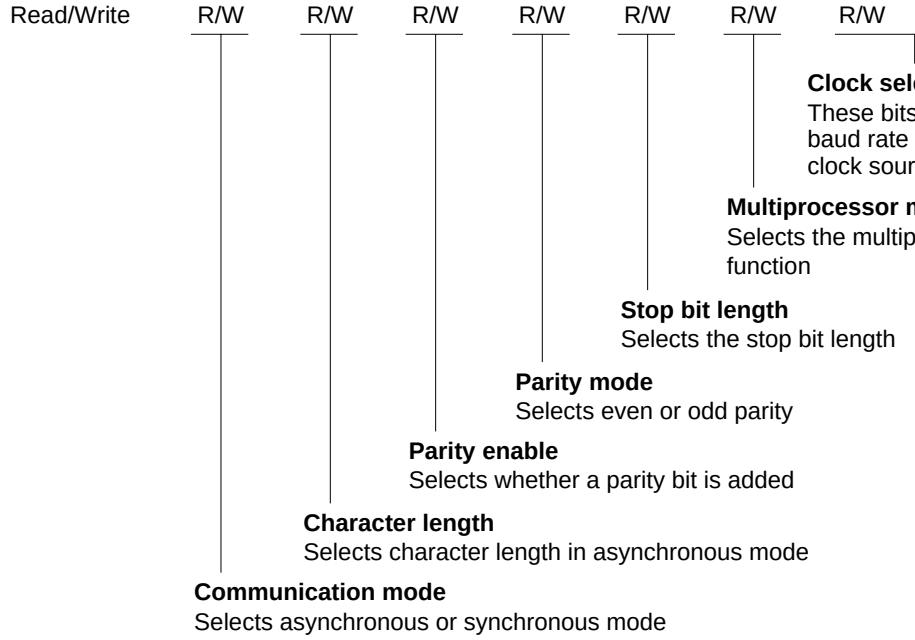
13.2.4 Transmit Data Register (TDR)

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 T
TSR and starts serial transmission. Continuous serial transmission is possible by writin
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 in s
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/$\overline{A}$): Selects whether the SCI operates in asynchron synchronous mode.

Bit 7: C/$\overline{A}$	Description	
0	Asynchronous mode	(
1	Synchronous mode	

Bit 5—Parity Enable (PE): In asynchronous mode, this bit enables or disables the add parity bit to transmit data, and the checking of the parity bit in receive data. In synchron the parity bit is neither added nor checked, regardless of the PE setting.

Bit 5: PE	Description	
0	Parity bit not added or checked	(In
1	Parity bit added and checked*	

Note: * When PE is set to 1, an even or odd parity bit is added to transmit data acco even or odd parity mode selected by the O/$\overline{E}$ bit, and the parity bit in receive checked to see that it matches the even or odd mode selected by the O/$\overline{E}$ bi

Bit 4—Parity Mode (O/$\overline{E}$): Selects even or odd parity. The O/$\overline{E}$ bit setting is valid in asynchronous mode when the PE bit is set to 1 to enable the adding and checking of a p The O/$\overline{E}$ setting is ignored in synchronous mode, or when parity adding and checking is in asynchronous mode.

Bit 4: O/$\overline{E}$	Description	
0	Even parity*1	(In
1	Odd parity*2	

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 d have an even number of 1s in the received character and parity bit combined

2. When odd parity is selected, the parity bit added to transmit data makes an o of 1s in the transmitted character and parity bit combined. Receive data mus odd number of 1s in the received character and parity bit combined.

2. Two stop bits (with value 1) are added at the end of each transmitted chara

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 multip format is selected, parity settings made by the PE and O/$\overline{E}$ bits are ignored. The MP b valid only in asynchronous mode. It is ignored in synchronous mode.

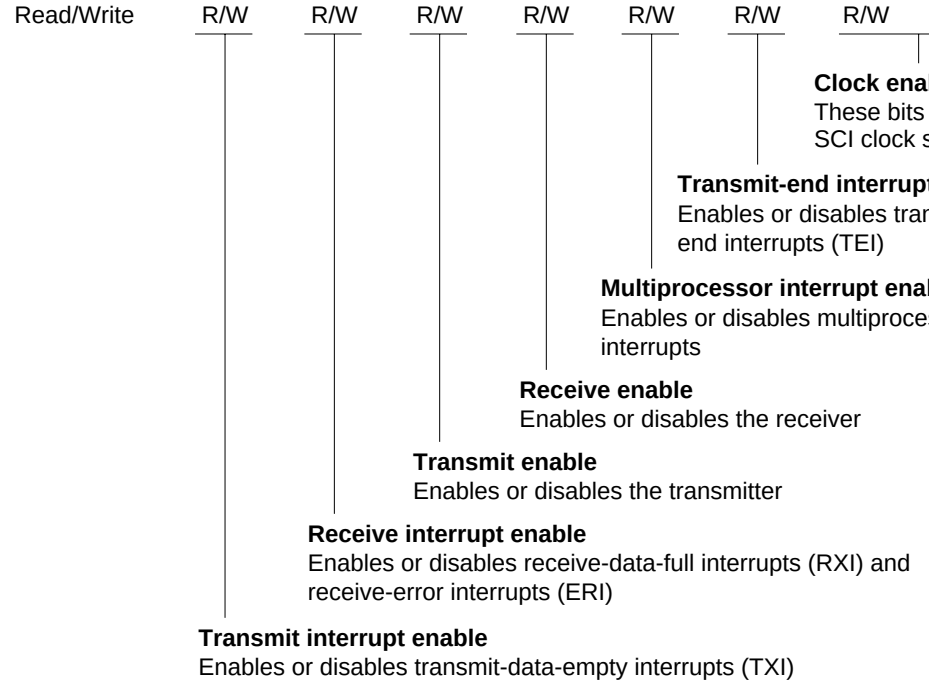
For further information on the multiprocessor communication function, see section 13 Multiprocessor Communication.

Bit 2: MP	Description	
0	Multiprocessor function disabled	(
1	Multiprocessor format selected	

Bits 1 and 0—Clock Select 1 and 0 (CKS1, CKS0): These bits select the clock sour chip baud rate generator. Four clock sources are available: ϕ, ϕ/4, ϕ/16, and ϕ/64.

For the relationship between the clock source, bit rate register setting, and baud rate, s 13.2.8, Bit Rate Register (BRR).

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



The CPU can always read and write SCR. SCR is initialized to H'00 by a reset and in st
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 is set to 1 in SSR 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 are e

Note: * RXI and ERI interrupt requests can be cleared by reading the value 1 from
FER, PER, or ORER flag, then clearing it to 0; or by clearing the RIE bit to

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 bit is locked at 1 in SSR.
2. In the enabled state, serial transmitting starts when the TDRE bit in SSR is
after writing of transmit data into TDR. Select the transmit format in SMR be
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 setting is valid only in asynchronous mode, and only if the MP bit is set to 1
The MPIE setting is ignored in synchronous mode or when the MP bit is cleared to 0.

Bit 3: MPIE	Description
0	Multiprocessor interrupts are disabled (normal receive operation) (I [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 d multiprocessor bit set to 1 is received.

Note: * The SCI does not transfer receive data from RSR to RDR, does not detect re errors, and does not set the RDRF, FER, and ORER flags in SSR. When it r data in which MPB = 1, the SCI sets the MPB bit to 1 in SSR, automatically c MPIE bit to 0, enables RXI and ERI interrupts (if the RIE bit is set to 1 in SCI allows the 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 new transmit data when the MSB is transmitte

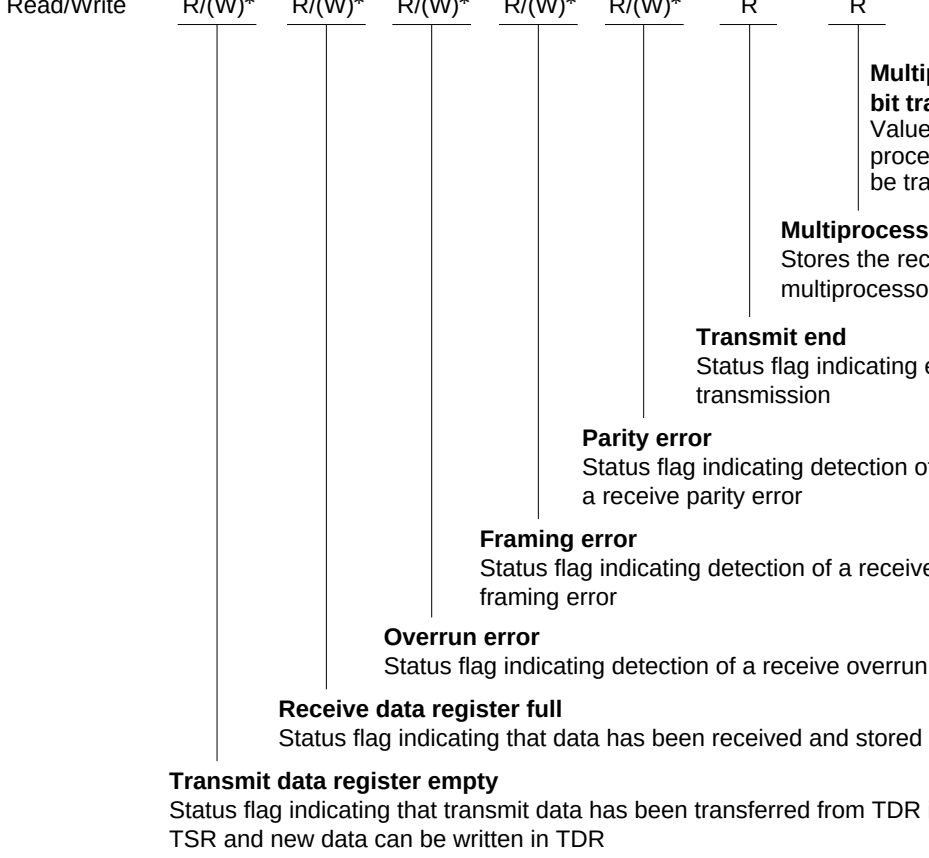
Bit 2: TEIE	Description
0	Transmit-end interrupt requests (TEI) are disabled* (In
1	Transmit-end interrupt requests (TEI) are enabled*

Note: * TEI interrupt requests can be cleared by reading the value 1 from the TDRE SSR, then clearing the TDRE flag to 0, thereby also clearing the TEND flag t clearing the TEIE bit to 0.

section 15.3, Operation.

Bit 1: CKE1	Bit 0: CKE0	Description	
0	0	Asynchronous mode	Internal clock, SCK pin available for gene input/output*1
		Synchronous mode	Internal clock, SCK pin used for serial clo
	1	Asynchronous mode	Internal clock, SCK pin used for clock outp
		Synchronous mode	Internal clock, SCK pin used for serial clo
1	0	Asynchronous mode	External clock, SCK pin used for clock inp
		Synchronous mode	External clock, SCK pin used for serial clo
	1	Asynchronous mode	External clock, SCK pin used for clock inp
		Synchronous mode	External clock, SCK pin used for serial clo

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.



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

The CPU can always read and write SSR, but cannot write 1 in the TDRE, RDRF, ORE and FER flags. These flags can be cleared to 0 only if they have first been read while se 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	(

Bit 6—Receive Data Register Full (RDRF): Indicates that RDR contains new receiv

Bit 6: RDRF	Description	
0	RDR does not contain new receive data [Clearing conditions] • The chip is reset or enters standby mode. • Software reads RDRF while it is set to 1, then writes 0. • The DMAC reads data from RDR.	(
1	RDR contains new receive data [Setting condition] When serial data is received normally and transferred from RSR to	

Note: The RDR contents and RDRF flag are not affected by detection of receive erro clearing of the RE bit to 0 in SCR. They retain their previous values. If the RDR set to 1 when reception of the next data ends, an overrun error occurs and rece 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 retains previous value.
2. RDR continues to hold the receive data before the overrun error, so subsequ data is lost. Serial receiving cannot continue while the ORER flag is set to 1. synchronous mode, serial transmitting is also disabled.

Bit 4—Framing Error (FER): Indicates that data reception ended abnormally due to a error in asynchronous mode.

Bit 4: FER	Description
0	Receiving is in progress or has ended normally (Initia [Clearing conditions] • The chip is reset or enters standby mode. • Software reads FER while it is set to 1, then writes 0.
1	A receive framing error occurred*2 [Setting condition] The stop bit at the end of receive data is checked and found to be 0

Notes: 1. Clearing the RE bit to 0 in SCR does not affect the FER flag, which retains it value.
2. When the stop bit length is 2 bits, only the first bit is checked. The second st checked. When a framing error occurs the SCI transfers the receive data int does not set the RDRF flag. Serial receiving cannot continue while the FER to 1. In synchronous mode, serial transmitting is also disabled.

1	A receive parity error occurred*2 [Setting condition] The number of 1s in receive data, including the parity bit, does not 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 value.
2. When a parity error occurs the SCI transfers the receive data into RDR but 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): Indicates that when the last bit of a serial character w transmitted TDR did not contain new transmit data, so transmission has ended. The TE a read-only bit and cannot be written.

Bit 2: TEND	Description
0	Transmission is in progress [Clearing conditions] • Software reads TDRE while it is set to 1, then writes 0 in the TE • 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. • TDRE is 1 when the last bit of a serial character is transmitted.

previous value.

Bit 0—Multiprocessor Bit Transfer (MPBT): Stores the value of the multiprocessor transmit data when a multiprocessor format is selected for transmitting in asynchronous The MPBT setting is ignored in synchronous mode, when a multiprocessor format is no or when the SCI is not transmitting.

Bit 0: 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 select 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 s mode. The two SCI channels have independent baud rate generator control, so different be set in the two channels.

Table 13.3 shows examples of BRR settings in asynchronous mode. Table 13.4 shows e BRR settings in synchronous mode.

300	0	207	0.16	0	217	0.21	0	255	0.00	1	7
600	0	103	0.16	0	108	0.21	0	127	0.00	0	1
1200	0	51	0.16	0	54	–0.70	0	63	0.00	0	7
2400	0	25	0.16	0	26	1.14	0	31	0.00	0	3
4800	0	12	0.16	0	13	–2.48	0	15	0.00	0	1
9600	0	6	–6.99	0	6	–2.48	0	7	0.00	0	9
19200	0	2	8.51	0	2	13.78	0	3	0.00	0	4
31250	0	1	0.00	0	1	4.86	0	1	22.88	0	2
38400	0	1	–18.62	0	1	–14.67	0	1	0.00	0	1

	ϕ (MHz)										
	3.6864			**4**			**4.9152**				
Bit Rate (bits/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	8
150	1	191	0.00	1	207	0.16	1	255	0.00	2	6
300	1	95	0.00	1	103	0.16	1	127	0.00	1	1
600	0	191	0.00	0	207	0.16	0	255	0.00	1	6
1200	0	95	0.00	0	103	0.16	0	127	0.00	0	1
2400	0	47	0.00	0	51	0.16	0	63	0.00	0	6
4800	0	23	0.00	0	25	0.16	0	31	0.00	0	3
9600	0	11	0.00	0	12	0.16	0	15	0.00	0	1
19200	0	5	0.00	0	6	–6.99	0	7	0.00	0	7
31250	0	3	–7.84	0	3	0.00	0	4	–1.70	0	4
38400	0	2	0.00	0	2	8.51	0	3	0.00	0	3

600	1	77	0.16	1	79	0.00	1	95	0.00	1	10
1200	0	155	0.16	0	159	0.00	0	191	0.00	0	20
2400	0	77	0.16	0	79	0.00	0	95	0.00	0	10
4800	0	38	0.16	0	39	0.00	0	47	0.00	0	51
9600	0	19	–2.34	0	19	0.00	0	23	0.00	0	25
19200	0	9	–2.34	0	9	0.00	0	11	0.00	0	12
31250	0	5	0.00	0	5	2.40	0	6	5.33	0	7
38400	0	4	–2.34	0	4	0.00	0	5	0.00	0	6

	ϕ (MHz)										
	9.8304			**10**			**12**			**1**	
Bit Rate (bits/s)	**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	21
150	2	127	0.00	2	129	0.16	2	155	0.16	2	15
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	15
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	15
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

600	1	168	0.16	1	181	0.16	1	191	0.00	1	2
1200	1	84	–0.43	1	90	0.16	1	95	0.00	1	1
2400	0	168	0.16	0	181	0.16	0	191	0.00	0	2
4800	0	84	–0.43	0	90	0.16	0	95	0.00	0	1
9600	0	41	0.76	0	45	–0.93	0	47	0.00	0	5
19200	0	20	0.76	0	22	–0.93	0	23	0.00	0	2
31250	0	12	0.00	0	13	0.00	0	14	–1.70	0	1
38400	0	10	–3.82	0	10	3.57	0	11	0.00	0	1

	ϕ (MHz)								
	18			**20**			**25**		
Bit Rate (bits/s)	**n**	**N**	**Error (%)**	**n**	**N**	**Error (%)**	**n**	**N**	**Error (%)**
110	3	79	–0.12	3	88	–0.25	3	110	–0.02
150	2	233	0.16	3	64	0.16	3	80	0.47
300	2	116	0.16	2	129	0.16	2	162	–0.15
600	1	233	0.16	2	64	0.16	2	80	0.47
1200	1	116	0.16	1	129	0.16	1	162	–0.15
2400	0	233	0.16	1	64	0.16	1	80	0.47
4800	0	116	0.16	0	129	0.16	0	162	–0.15
9600	0	58	–0.69	0	64	0.16	0	80	0.47
19200	0	28	1.02	0	32	–1.36	0	40	–0.76
31250	0	17	0.00	0	19	0.00	0	24	0.00
38400	0	14	–2.34	0	15	1.73	0	19	1.73

1 k	1	124	1	249	2	124	—	—	2	202	2	249	3	69	3	77
2.5 k	0	199	1	99	1	199	1	249	2	80	2	99	2	112	2	12
5 k	0	99	0	199	1	99	1	124	1	162	1	199	1	224	1	24
10 k	0	49	0	99	0	199	0	249	1	80	1	99	1	112	1	12
25 k	0	19	0	39	0	79	0	99	0	129	0	159	0	179	0	19
50 k	0	9	0	19	0	39	0	49	0	64	0	79	0	89	0	99
100 k	0	4	0	9	0	19	0	24	—	—	0	39	0	44	0	49
250 k	0	1	0	3	0	7	0	9	0	12	0	15	0	17	0	19
500 k	0	0*	0	1	0	3	0	4	—	—	0	7	0	8	0	9
1 M			0	0*	0	1	—	—	—	—	0	3	0	4	0	4
2 M					0	0*	—	—	—	—	0	1	—	—	—	—
2.5 M					—	—	0	0*	—	—	—	—	—	—	—	—
4 M											0	0*	—	—	—	—

Legend:

Blank: No setting available

—: Setting possible, but error occurs

*: Continuous transmit/receive not possible

Note: Settings with an error of 1% or less are recommended.

$$N = \frac{\phi \times 10^6}{8 \times 2^{2n-1} \times B} \times 10^6 - 1$$

B: Bit rate (bits/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	ϕ/4	0	1
2	ϕ/16	1	0
3	ϕ/64	1	1

The bit rate error in asynchronous mode is calculated as follows.

$$\text{Error (\%)} = \left\{ \frac{\phi \times 10^6}{(N+1) \times B \times 64 \times 2^{2n-1}} - 1 \right\} \times 100$$

2.097152	65536	0	0
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

selected by the C/A bit in SMR and the CKE1 and CKE0 bits in SCR, as shown in tab

Asynchronous Mode:

- Data length is selectable: 7 or 8 bits.
- Parity and multiprocessor bits are selectable. So is the stop bit length (1 or 2 bits). 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 outputs 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.

			1	0				Preser
				1				
	1		0	0		7-bit data		Absent
				1				
			1	0				Preser
				1				
	0	1	—	0	Asynchronous mode (multi-processor format)	8-bit data	Present	Absent
				1				
	1			0		7-bit data		
				1				
1	—	—	—	—	Synchronous mode	8-bit data	Absent	

Table 13.9 SMR and SCR Settings and SCI Clock Source Selection

SMR	SCR Settings			SCI Transmit/Receive Cloc	
Bit 7: C/$\overline{A}$	**Bit 1: CKE1**	**Bit 0: CKE0**	**Mode**	**Clock Source**	**SCK Pin Function**
0	0	0	Asynchronous mode	Internal	SCI does not use the SCK
		1			Outputs a clock with freque matching the bit rate
	1	0		External	Inputs a clock with frequen times the bit rate
		1			
1	0	0	Synchronous mode	Internal	Outputs the serial clock
		1			
	1	0		External	Inputs the serial clock
		1			

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 stop bit (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.

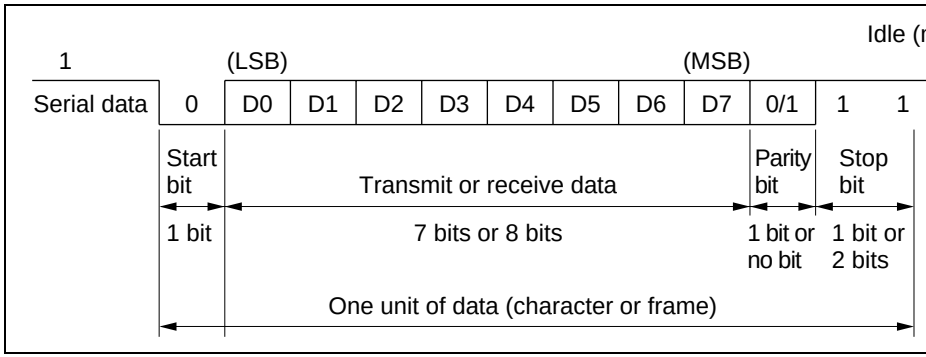


Figure 13.2 Data Format in Asynchronous Communication (Example: 8-Bit Data with Parity and 2 Stop Bits)

0	0	0	0	S	8-bit data	STOP		
0	0	0	1	S	8-bit data	STOP	S	
0	1	0	0	S	8-bit data	P	S	
0	1	0	1	S	8-bit data	P	S	
1	0	0	0	S	7-bit data	STOP		
1	0	0	1	S	7-bit data	STOP	STOP	
1	1	0	0	S	7-bit data	P	STOP	
1	1	0	1	S	7-bit data	P	STOP	S
0	—	1	0	S	8-bit data	MPB	S	
0	—	1	1	S	8-bit data	MPB	S	
1	—	1	0	S	7-bit data	MPB	STOP	
1	—	1	1	S	7-bit data	MPB	STOP	S

Legend:

S: Start bit

STOP: Stop bit

P: Parity bit

MPB: Multiprocessor bit

When the SCI operates on an internal clock, it can output a clock signal at the SCK pi
frequency of this output clock is equal to the bit rate. The phase is aligned as in figure
the rising edge of the clock occurs at the center of each transmit data bit.

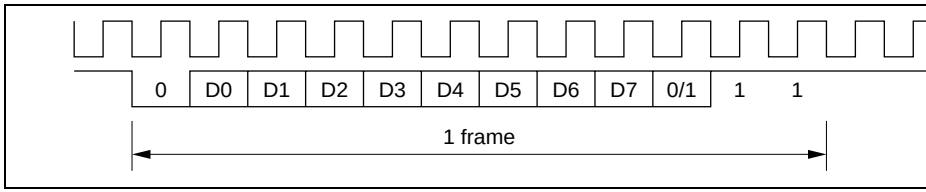


**Figure 13.3 Phase Relationship between Output Clock and Serial Dat
(Asynchronous Mode)**

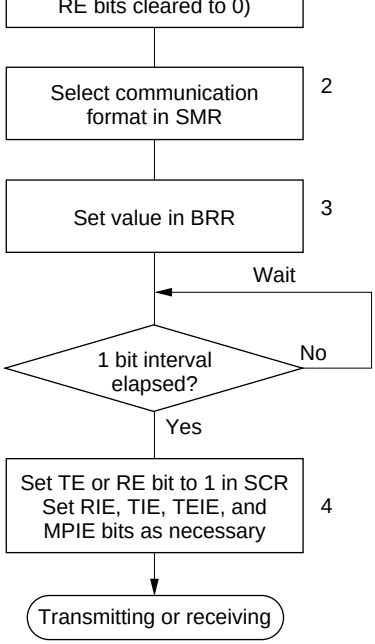
Transmitting and Receiving Data

SCI Initialization (Asynchronous Mode): Before transmitting or receiving, clear the
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
following the procedure given below. Clearing TE to 0 sets the TDRE flag to 1 and in
TSR. Clearing RE to 0, however, does not initialize the RDRF, PER, FER, and ORER
RDR, which retain their previous contents.

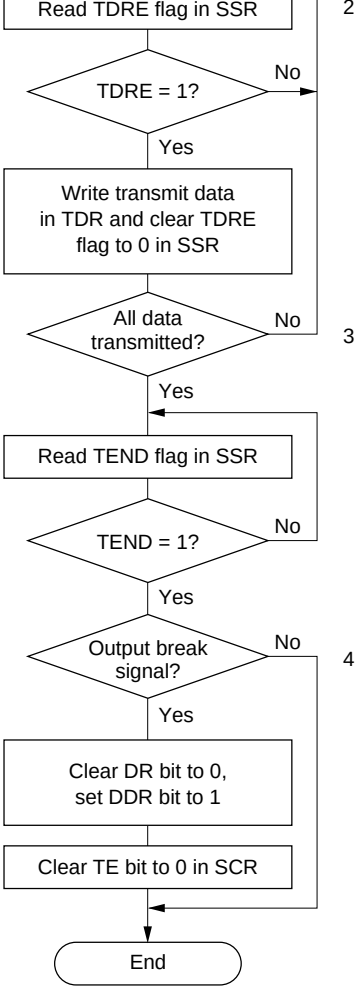
When an external clock is used, the clock should not be stopped during initialization o
operation. SCI operation becomes unreliable if the clock is stopped.

Figure 13.4 is a sample flowchart for initializing the SCI.



TIE, TEIE, and MPIE bits as necessary. Setting
or RE bit enables the SCI to use the TxD or RxD

Figure 13.4 Sample Flowchart for SCI Initialization



that the TDRE flag is 1, indicating that data can
written, write data in TDR, then clear the TDRE
flag to 0. When the DMAC is activated by a tran
data-empty interrupt request (TXI) to write data
the TDRE flag is checked and cleared automati

4. To output a break signal at the end of serial tra
set the DDR bit to 1 and clear the DR bit to 0
(DDR and DR are I/O port registers), then clea
TE bit to 0 in SCR.

Figure 13.5 Sample Flowchart for Transmitting Serial Data

a. Start bit: One 0 bit is output.
b. Transmit data: 7 or 8 bits are output, LSB first.
c. Parity bit or multiprocessor bit: One parity bit (even or odd parity) or one multip
bit is output. Formats in which neither a parity bit nor a multiprocessor bit is ou
also be selected.
d. Stop bit: One or two 1 bits (stop bits) are output.
e. Mark state: Output of 1 bits continues until the start bit of the next transmit data

3. 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
then continues output of 1 bits in the mark state. If the TEIE bit is set to 1 in SCR, a
end interrupt (TEI) is requested at this time.

Figure 13.6 shows an example of SCI transmit operation in asynchronous mode.

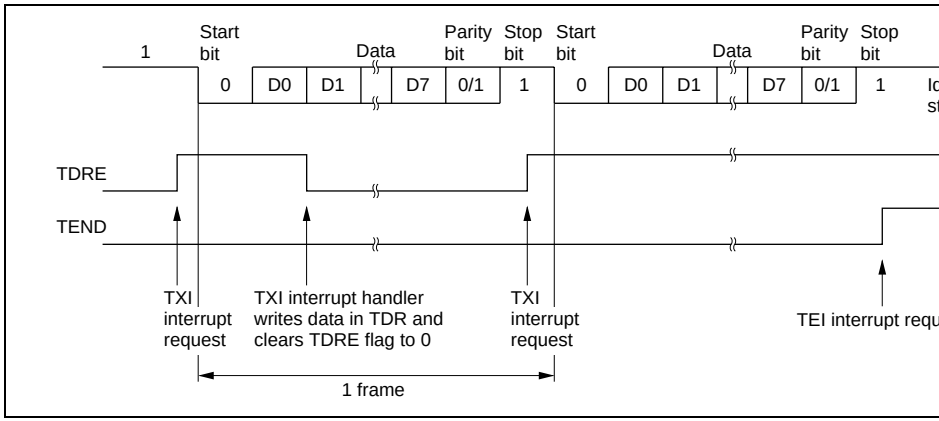
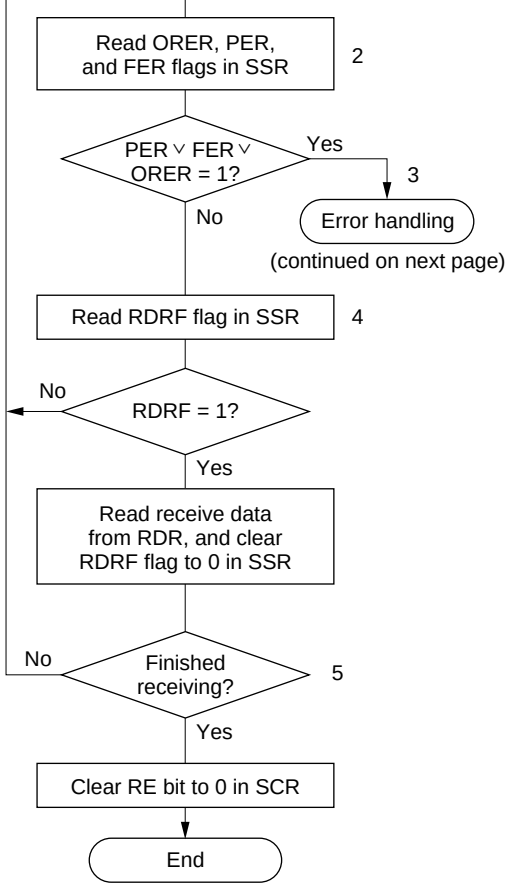


**Figure 13.6 Example of SCI Transmit Operation in Asynchronous Mod
(8-Bit Data with Parity and 1 Stop Bit)**



handling, clear the ORER, PER, ar
flags all to 0. Receiving cannot resu
of the ORER, PER, and FER flags
set to 1. When a framing error occu
RxD pin can be read to detect the b

4. SCI status check and receive data
SSR, check that RDRF is set to 1,
receive data from RDR and clear th
flag to 0. Notification that the RDRF
changed from 0 to 1 can also be gi
RXI interrupt.
5. To continue receiving serial data: c
RDRF flag, read RDR, and clear th
flag to 0 before the stop bit of the c
frame is received. If the DMAC is a
by an RXI interrupt to read the RDR
the RDRF flag is cleared automatic

Figure 13.7 Sample Flowchart for Receiving Serial Data (1)

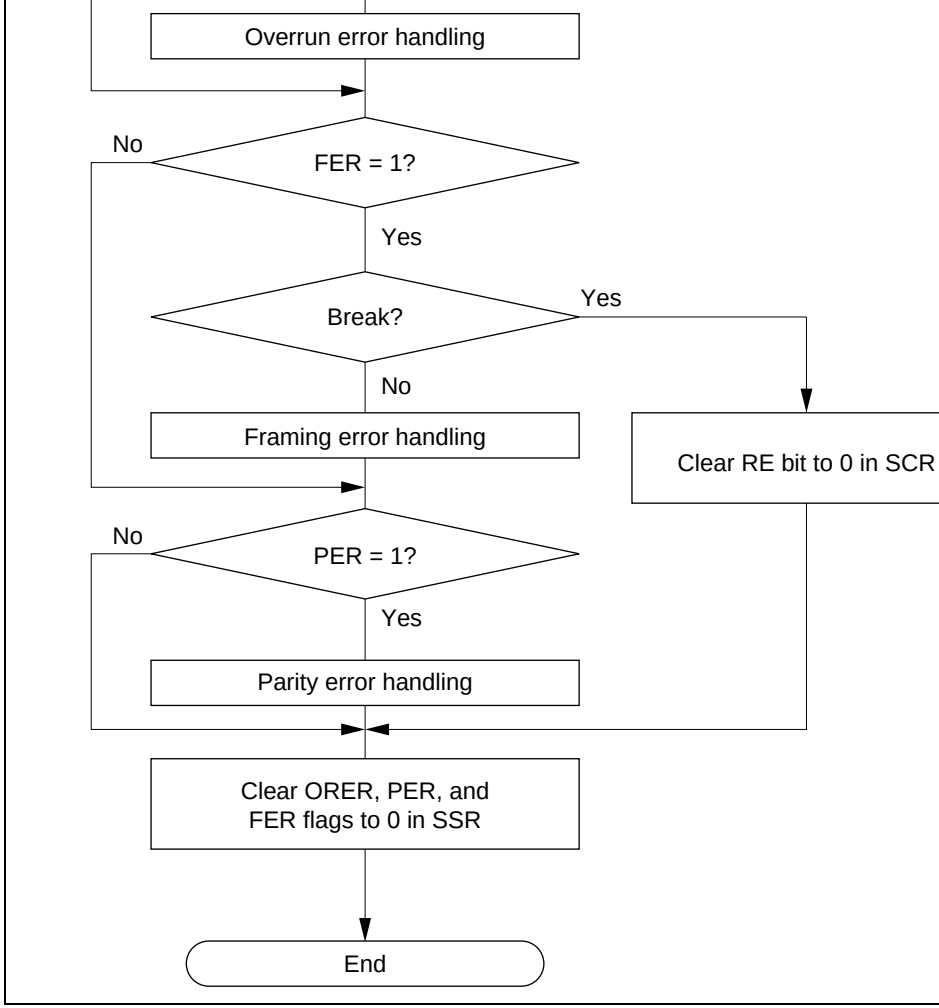


Figure 13.7 Sample Flowchart for Receiving Serial Data (2)

a. Parity check: The number of 1s in the receive data must match the even or odd setting of the O/$\overline{E}$ bit in SMR.

b. Stop bit check: The stop bit value must be 1. If there are two stop bits, only the is checked.

c. Status check: The RDRF flag must be 0 so that receive data can be transferred into RDR.

If these checks all pass, the RDRF flag is set to 1 and the received data is stored in of the checks fails (receive error)*, the SCI operates as indicated in table 13.11.

Note: * When a receive error occurs, further receiving is disabled. In receiving, the is not set to 1. Be sure to clear the error flags to 0.

4. 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, 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 not from RSR to RDR
Framing error	FER	Stop bit is 0	Receive data tran RSR to RDR
Parity error	PER	Parity of receive data differs from even/odd parity setting in SMR	Receive data tran RSR to RDR

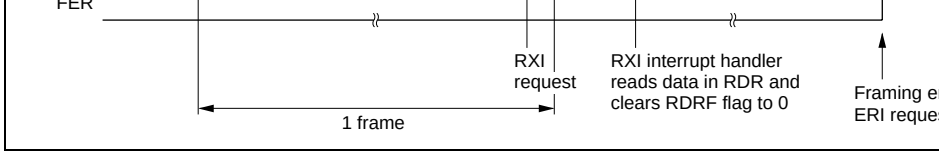


Figure 13.8 Example of SCI Receive Operation (8-Bit Data with Parity and One

13.3.3 Multiprocessor Communication

The multiprocessor communication function enables several processors to share a singl communication line. The processors communicate in asynchronous mode using a forma additional multiprocessor bit (multiprocessor format).

In multiprocessor communication, each receiving processor is addressed by an ID. A se communication cycle consists of an ID-sending cycle that identifies the receiving proce data-sending cycle. The multiprocessor bit distinguishes ID-sending cycles from data-s cycles.

The transmitting processor starts by sending the ID of the receiving processor with whi to communicate as data with the multiprocessor bit set to 1. Next the transmitting proce transmit data with the multiprocessor bit cleared to 0.

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

Figure 13.9 shows an example of communication among different processors using a multiprocessor format.

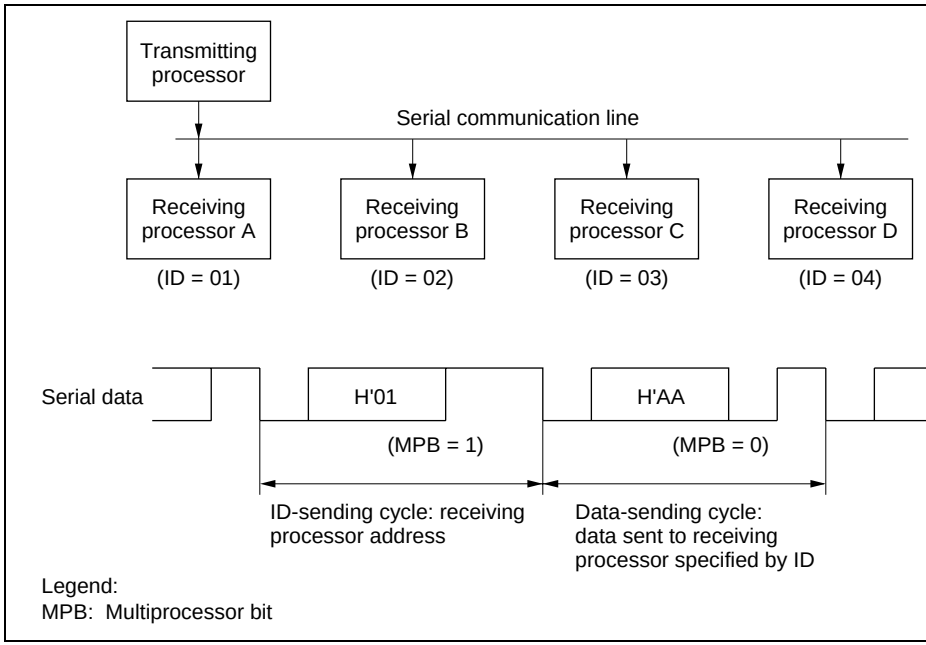
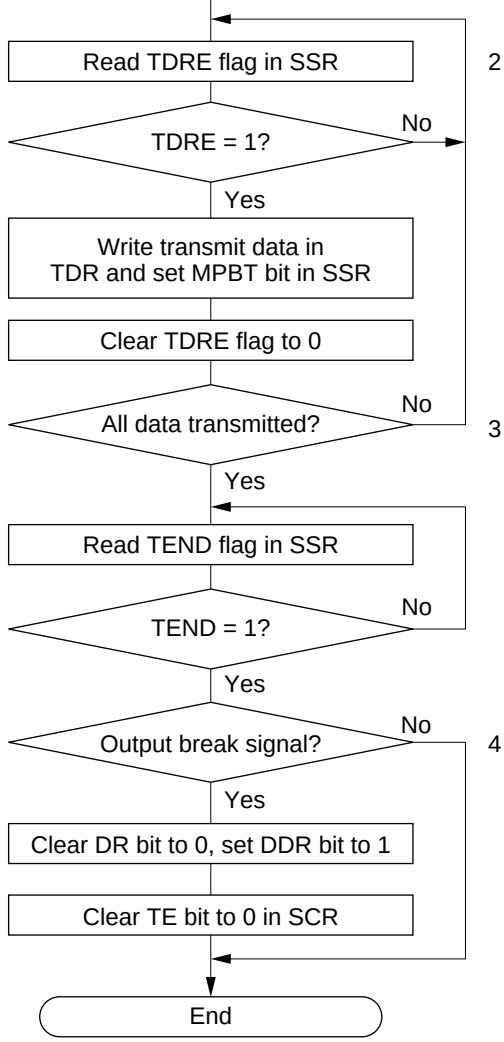


Figure 13.9 Example of Communication among Processors Using Multiprocess (Sending Data H'AA to Receiving Processor A)



write: read SSR, check that th
flag is 1, then write transmit
data in TDR. Also set the MPE
0 or 1 in SSR. Finally, clear th
flag to 0.

3. To continue transmitting serial
after checking that the TDRE f
indicating that data can be
written, write data in TDR, ther
the TDRE flag to 0. When the
is activated by a transmit-data
interrupt request (TXI) to write
TDR, the TDRE flag is checke
cleared automatically.
4. To output a break signal at the
serial transmission: set the DD
1 and clear the DR bit to 0 (DD
DR are I/O port registers), ther
the TE bit to 0 in SCR.

Figure 13.10 Sample Flowchart for Transmitting Multiprocessor Serial D

a. Start bit: One 0 bit is output.
b. Transmit data: 7 or 8 bits are output, LSB first.
c. Multiprocessor bit: One multiprocessor bit (MPBT value) is output.
d. Stop bit: One or two 1 bits (stop bits) are output.
e. Mark state: Output of 1 bits continues until the start bit of the next transmit dat

3. The SCI checks the TDRE flag when it outputs the stop bit. If the TDRE flag is 0, loads data from TDR into TSR, outputs the stop bit, then begins serial transmissio frame. If the TDRE flag is 1, the SCI sets the TEND flag in SSR to 1, outputs the s continues output of 1 bits in the mark state. If the TEIE bit is set to 1 in SCR, a tra interrupt (TEI) is requested at this time.

Figure 13.11 shows an example of SCI transmit operation using a multiprocessor form

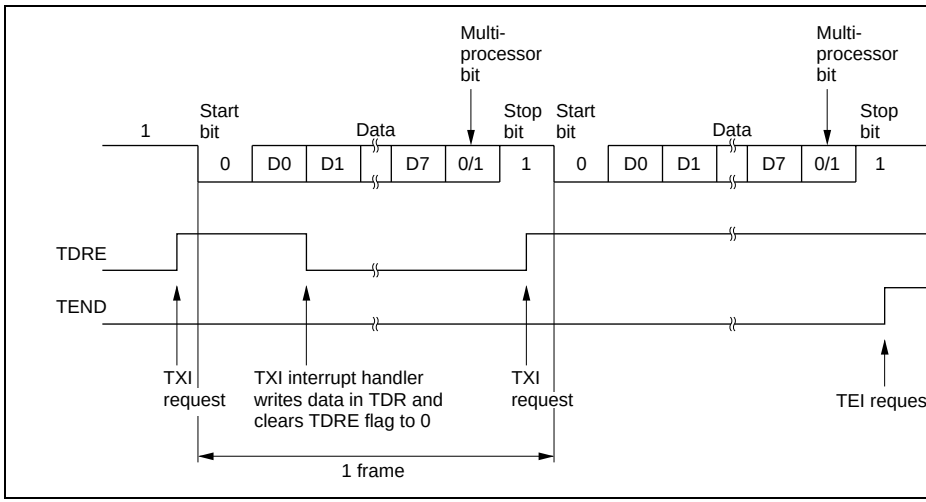
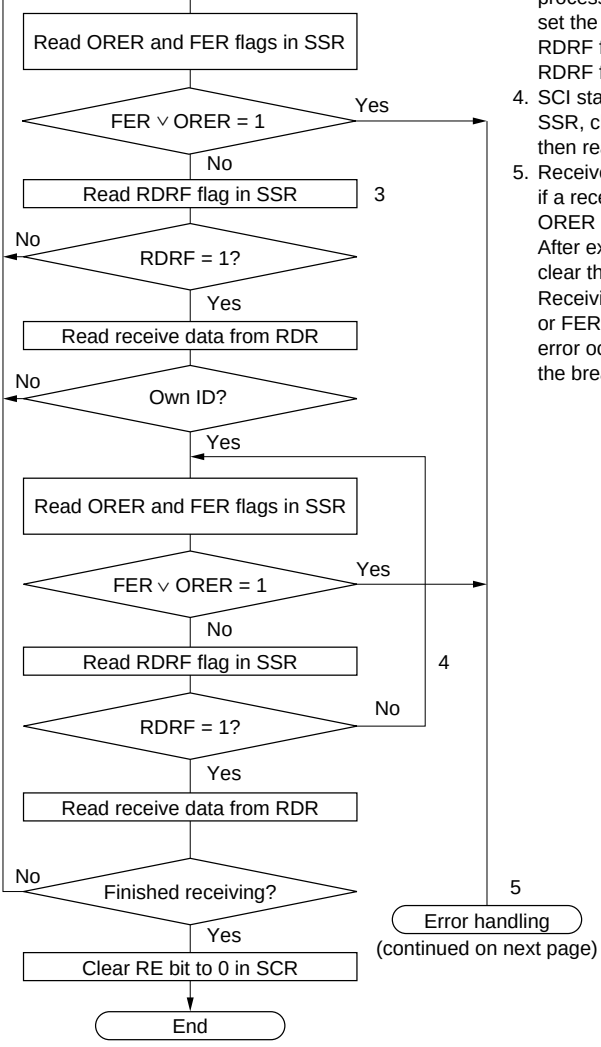


Figure 13.11 Example of SCI Transmit Operation (8-Bit Data with Multiprocessor Bit and One Stop Bit)



processor's own ID. If the ID does not m
set the MPIE bit to 1 again and clear the
RDRF flag to 0. If the ID matches, clear
RDRF flag to 0.

4. SCI status check and data receiving: rea
SSR, check that the RDRF flag is set to
then read data from RDR.
5. Receive error handling and break detect
if a receive error occurs, read the
ORER and FER flags in SSR to identify
After executing the necessary error hand
clear the ORER and FER flags both to 0
Receiving cannot resume while either th
or FER flag remains set to 1. When a fra
error occurs, the RxD pin can be read to
the break state.

Figure 13.12 Sample Flowchart for Receiving Multiprocessor Serial Data

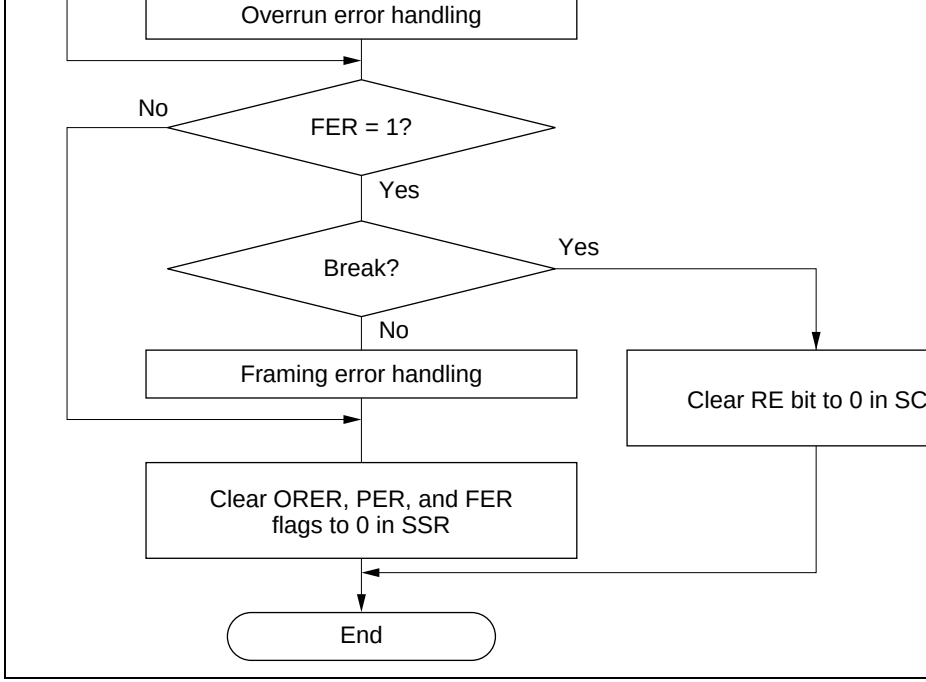
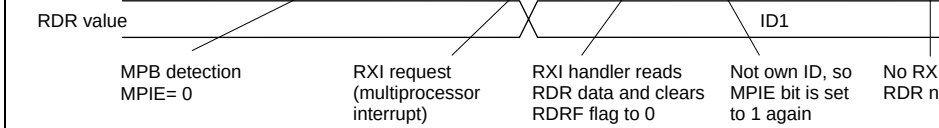
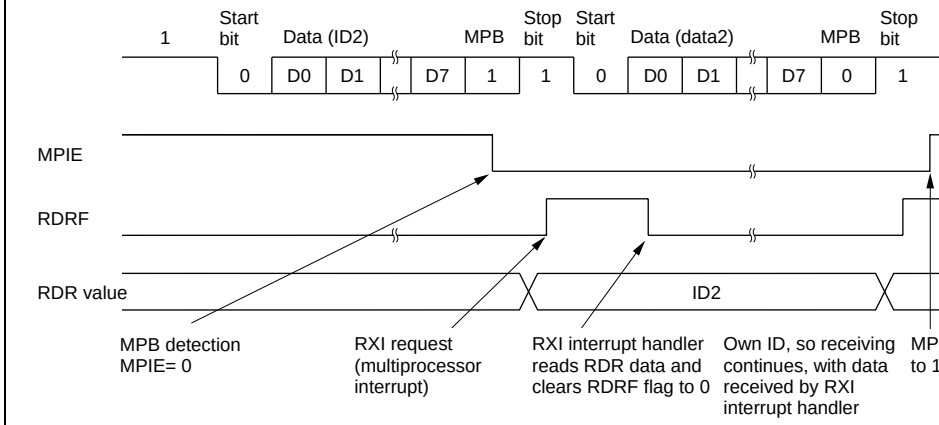


Figure 13.12 Sample Flowchart for Receiving Multiprocessor Serial Dat

Figure 13.13 shows an example of SCI receive operation using a multiprocessor forma



a. Own ID does not match data



b. Own ID matches 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.

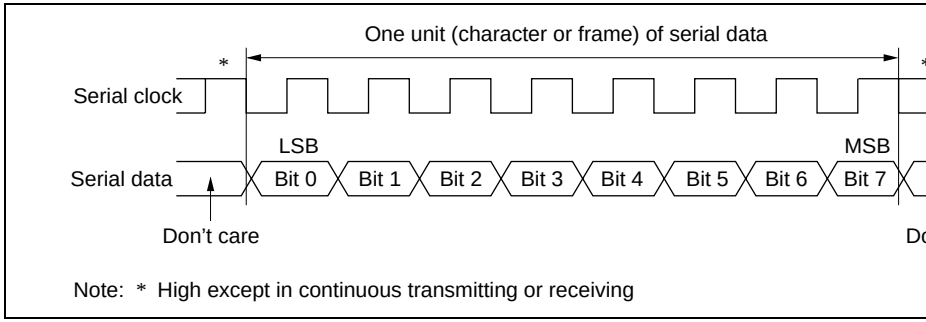


Figure 13.14 Data Format in Synchronous Communication

In synchronous serial communication, each data bit is placed on the communication li
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 transmitted in order from LSB (first) to MSB
output of the MSB, the communication line remains in the state of the MSB. In synch
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.

transmitting or receiving, the clock signal remains in the high state. However, when rec
only, overrun error may occur or the serial clock continues output until the RE bit clear
When transmitting or receiving in single characters, select the external clock.

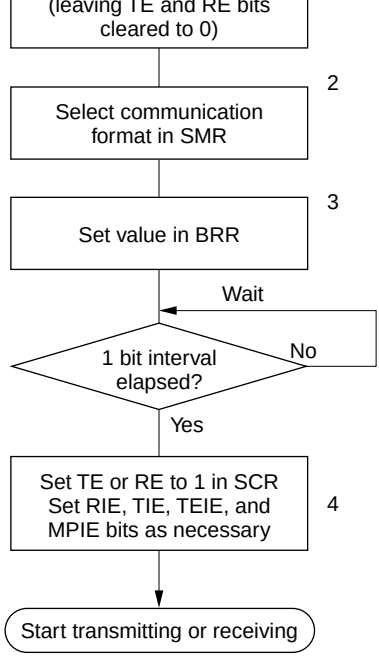
Transmitting and Receiving Data

SCI Initialization (Synchronous Mode): Before transmitting or receiving, clear the T

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 the TE bit to 0 sets the TDRE flag to 1 a
initializes TSR. Clearing the RE bit to 0, however, does not initialize the RDRF, PER,
ORE flags and RDR, which retain their previous contents.

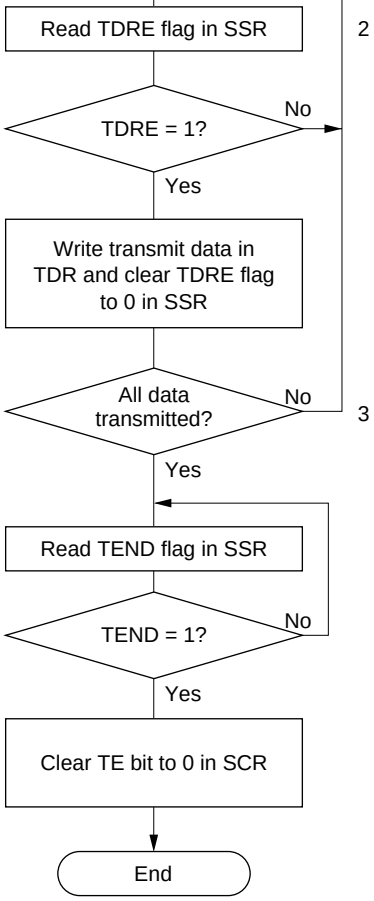
Figure 13.15 is a sample flowchart for initializing the SCI.



TxD or RxD pin.

Note: In simultaneous transmitting and receiving, the RE bits should be cleared to 0 or set to 1 simu

Figure 13.15 Sample Flowchart for SCI Initialization



that the TDRE flag is 1, indicating that data
written, write data in TDR, then clear the TD
to 0. When the DMAC is activated by a tran
data-empty interrupt request (TXI) to write d
TDR, the TDRE flag is checked and cleared
automatically.

Figure 13.16 Sample Flowchart for Serial Transmitting

is selected, the SCI outputs data in synchronization with the input clock. Data is ou
the TxD pin in order from LSB (bit 0) to MSB (bit 7).

3. The SCI checks the TDRE flag when it outputs the MSB (bit 7). If the TDRE flag
loads data from TDR into TSR and begins serial transmission of the next frame. If
flag is 1, the SCI sets the TEND flag to 1 in SSR, and after transmitting the MSB,
TxD pin in the MSB state. If the TEIE bit in SCR is set to 1, a transmit-end interru
requested at this time.

4. 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.

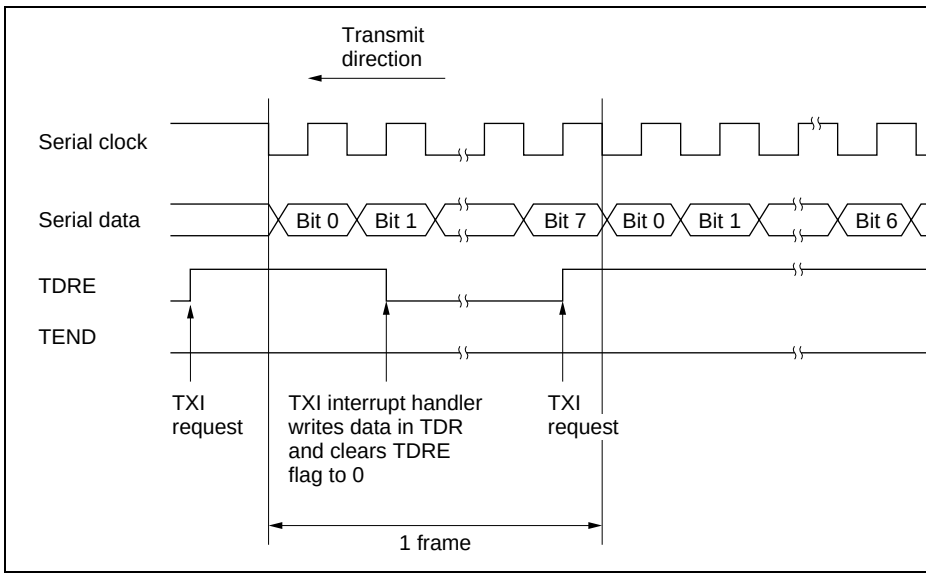
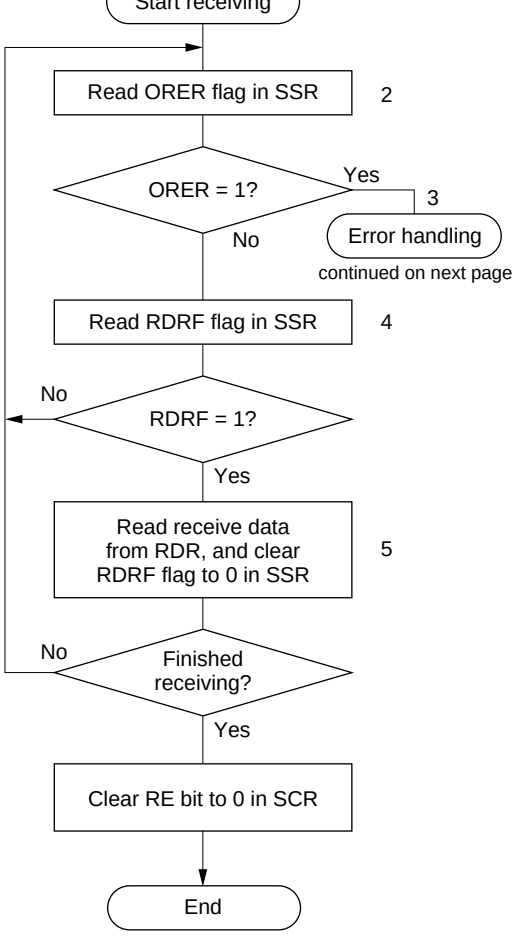


Figure 13.17 Example of SCI Transmit Operation



occurs, read the ORER flag in SSR,
executing the necessary error handl
the ORER flag to 0. Neither transmit
receiving can resume while the ORI
remains set to 1.

4. SCI status check and receive data r
SSR, check that the RDRF flag is se
then read receive data from RDR an
the RDRF flag to 0. Notification that
flag has changed from 0 to 1 can als
given by the RXI interrupt.
5. To continue receiving serial data: ch
RDRF flag, read RDR, and clear the
flag to 0 before the MSB (bit 7) of th
frame is received. If the DMAC is ac
by a receive-data-full interrupt reque
to read RDR, the RDRF flag is clear
automatically.

Figure 13.18 Sample Flowchart for Serial Receiving (1)

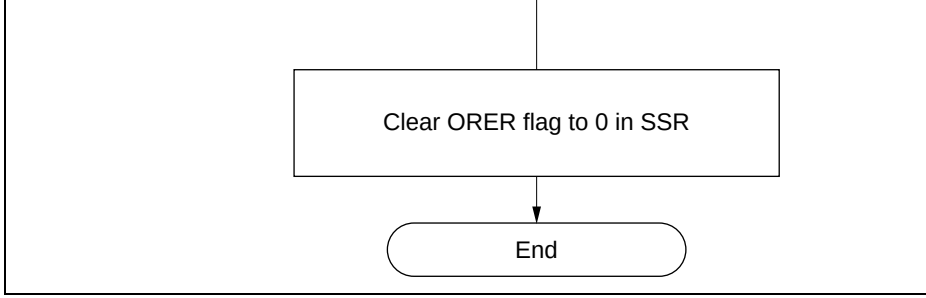


Figure 13.18 Sample Flowchart for Serial Receiving (2)

In receiving, the SCI operates as follows.

1. The SCI synchronizes with serial clock input or output and initializes internally.
2. 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 dat transferred from RSR to RDR. If this check passes, the RDRF flag is set to 1 and t data is stored in RDR. If the check does not pass (receive error), the SCI operates a in table 13.11.
3. After setting the RDRF flag to 1, if the RIE bit is set to 1 in SCR, the SCI requests data-full interrupt (RXI). If the ORER flag is set to 1 and the RIE bit in SCR is als the SCI requests a receive-error interrupt (ERI).

Figure 13.19 shows an example of SCI receive operation.

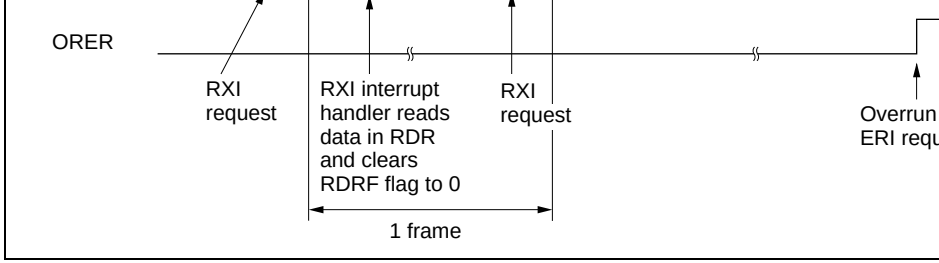
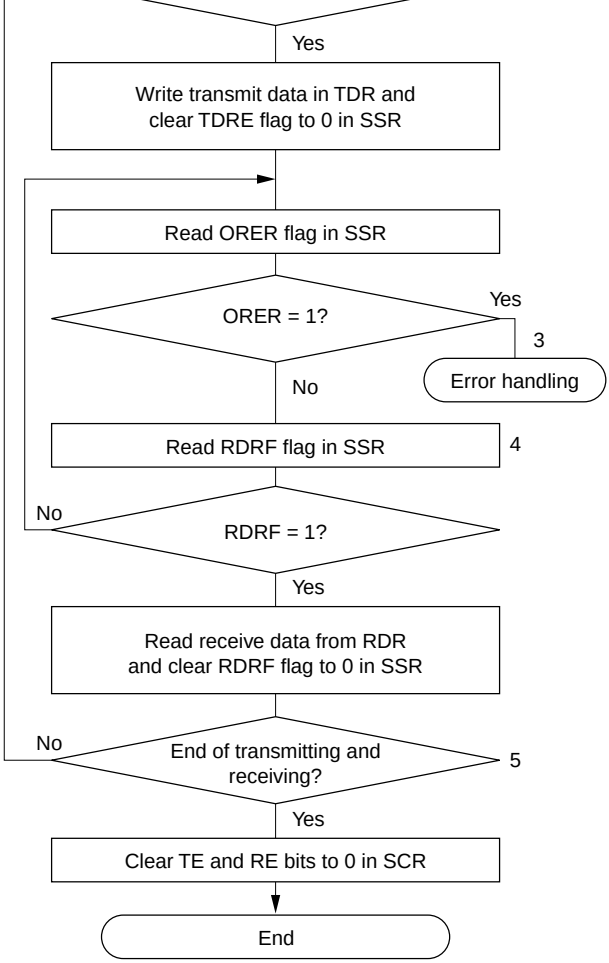


Figure 13.19 Example of SCI Receive Operation

Transmitting and Receiving Serial Data Simultaneously (Synchronous Mode): Fig shows a sample flowchart for transmitting and receiving serial data simultaneously and the procedure to follow.



Note: When switching from transmitting or receiving to simultaneous transmitting and receiving, clear both the TE bit and the RE bit to 0, then set both bits to 1.

3. Receive error handling: if error occurs, read the OR SSR, then after executing sary error handling, clear flag to 0.
   Neither transmitting nor re can resume while the OR remains set to 1.
4. SCI status check and rece data read: read SSR, che the RDRF flag is 1, then r receive data from RDR ar the RDRF flag to 0. Notific that the RDRF flag has ch from 0 to 1 can also be gi by the RXI interrupt.
5. To continue transmitting a receiving serial data: chec RDRF flag, read RDR, an the RDRF flag to 0 before MSB (bit 7) of the current is received. Also check tha the TDRE flag is set to 1, ing that data can be writte data in TDR, then clear th flag to 0 before the MSB (the current frame is transı When the DMAC is activa a transmit-data-empty inte request (TXI) to write data the TDRE flag is checked cleared automatically. Wh DMAC is activated by a re data-full interrupt request read RDR, the RDRF flag cleared automatically.

Figure 13.20 Sample Flowchart for Serial Transmitting

requested when the TEND flag is set to 1 in SSR. The TXI interrupt request can activat
DMAC to transfer data. Data transfer by the DMAC automatically clears the TDRE fla
TEI interrupt request cannot activate the DMAC.

The RXI interrupt is requested when the RDRF flag is set to 1 in SSR. The ERI interru
requested when the ORER, PER, or FER flag is set to 1 in SSR. The RXI interrupt requ
activate the DMAC to transfer data. Data transfer by the DMAC automatically clears th
flag to 0. The 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	Description	Priority
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

Data can be written into TDR regardless of the state of the TDRE flag. If new data is w
TDR when the TDRE flag is 0, the old data stored in TDR will be lost because this da
yet been transferred to TSR. Before writing transmit data in TDR, be sure to check tha
flag is set to 1.

Simultaneous Multiple Receive Errors

Table 13.13 indicates the state of SSR status flags when multiple receive errors occur
simultaneously. When an overrun error occurs the RSR contents are not transferred to
receive data is lost.

Table 13.13 SSR Status Flags and Transfer of Receive Data

SSR Status Flags				Receive Data Transfer	
RDRF	ORER	FER	PER	RSR → RDR	Receive Errors
1	1	0	0	×	Overrun error
0	0	1	0	O	Framing error
0	0	0	1	O	Parity error
1	1	1	0	×	Overrun error + framing error
1	1	0	1	×	Overrun error + parity error
0	0	1	1	O	Framing error + parity error
1	1	1	1	×	Overrun error + framing error +

Legend:
O: Receive data is transferred from RSR to RDR.
×: Receive data is not transferred from RSR to RDR.

When the TE bit is cleared to 0 the TxD pin becomes an I/O port, the level and directio output) of which are determined by DR and DDR bits. This feature can be used to send signal.

After the serial transmitter is initialized, the DR value substitutes for the mark state unt bit is set to 1 (the TxD pin function is not selected until the TE bit is set to 1). The DDR bits should therefore both be set to 1 beforehand.

To send a break signal during serial transmission, clear the DR bit to 0, then clear the T When the TE bit is cleared to 0 the transmitter is initialized, regardless of its current sta TxD pin becomes an output port outputting the value 0.

Receive Error Flags and Transmitter Operation (Synchronous Mode Only)

When a receive error flag (ORER, PER, or FER) is set to 1 the SCI will not start transm even if the TDRE flag is cleared to 0. Be sure to clear the receive error flags to 0 when transmit. Note 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 asynchronous mode the SCI operates on a base clock with 16 times the bit rate frequ receiving, the SCI synchronizes internally with the fall of the start bit, which it samples base clock. Receive data is latched at the rising edge of the eighth base clock pulse. See 13.21.

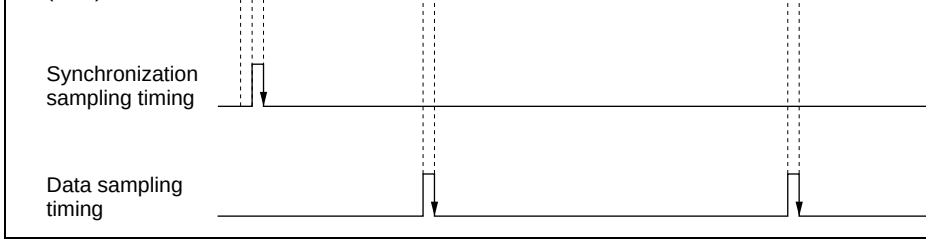


Figure 13.21 Receive Data Sampling Timing in Asynchronous Mode

The receive margin in asynchronous mode can therefore be expressed as in equation (

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

M: Receive margin (%)
N: Ratio of clock frequency to bit rate (N = 16)
D: Clock duty cycle (D = 0 to 1.0)
L: Frame length (L = 9 to 12)
F: Absolute deviation of clock frequency

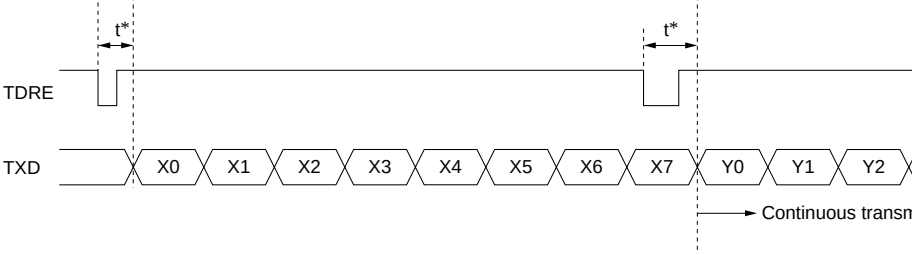
From equation (1), if F = 0 and D = 0.5 the receive margin is 46.875%, as given by eq

$$D = 0.5,\ F = 0$$
$$M = \{0.5 - 1/(2 \times 16)\} \times 100\%$$
$$= 46.875\% \quad \text{..... (2)}$$

This is a theoretical value. A reasonable margin to allow in system designs is 20% to

Restrictions on Usage of DMAC

To have the DMAC read RDR, be sure to select the SCI receive-data-full interrupt (R activation source with bits DTS2 to DTS0 in DTCR.



Note: * Ensure that t ≥ 5 states.

Figure 13.22 Serial Clock Transmission (Example)

3. C/$\overline{A}$ bit = 0 ... switchover to port output
4. Occurrence of low-level output (see figure 13.23)

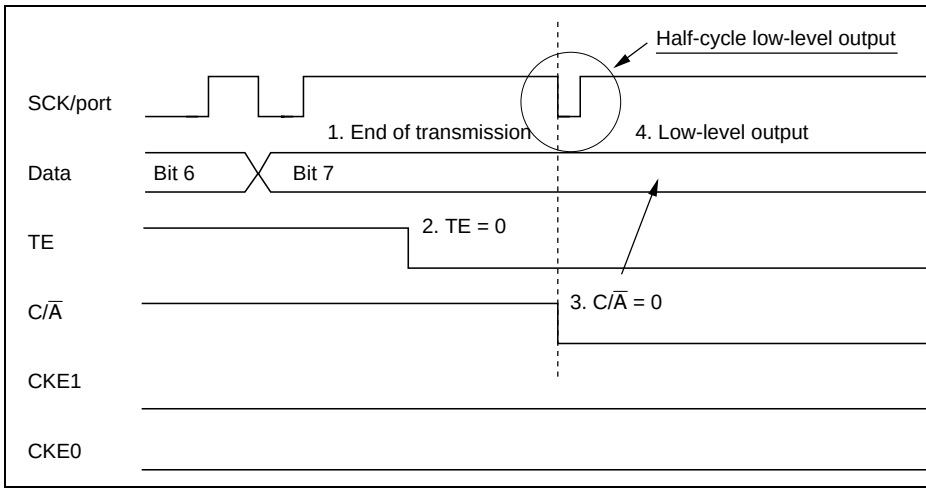


Figure 13.23 Operation when Switching from SCK Pin to Port Pin

2. TE bit = 0
3. CKE1 bit = 1
4. C/$\overline{A}$ bit = 0 ... switchover to port output
5. CKE1 bit = 0

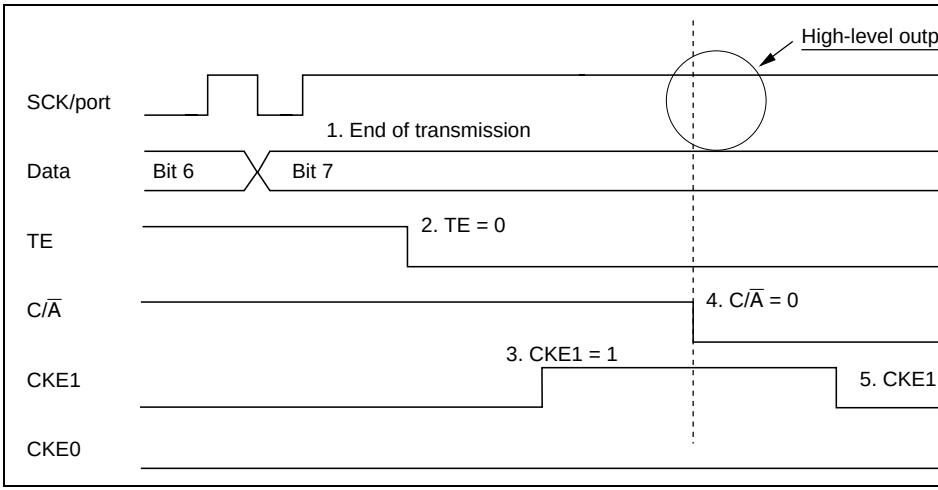


Figure 13.24 Operation when Switching from SCK Pin Function to Port Pin F (Example of Preventing Low-Level Output)

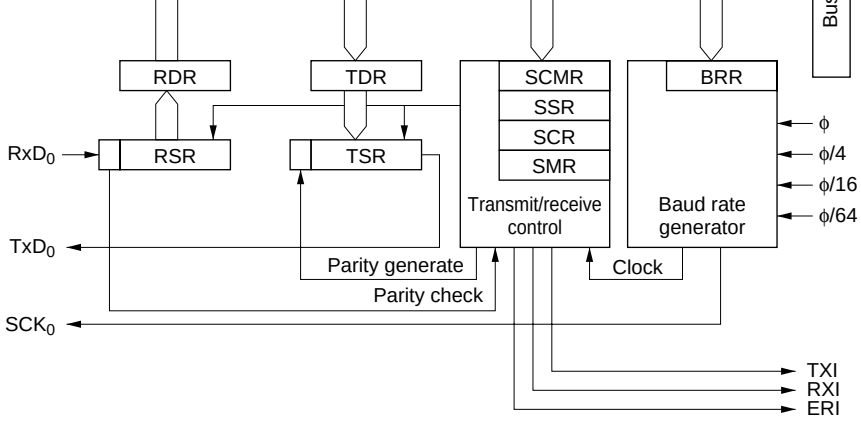
setting.

14.1.1 Features

Features of the smart-card interface supported by the H8/3048B Group are listed belo

- 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 receive-error interrupts are requested independently. The transmit-data-empty and receive-data-full interrupts can activa controller (DMAC) to transfer data.



Legend:
SCMR: Smart card mode register
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

Figure 14.1 Smart Card Interface Block Diagram

Transmit data pin	TxD_0	Output	Transmit data output

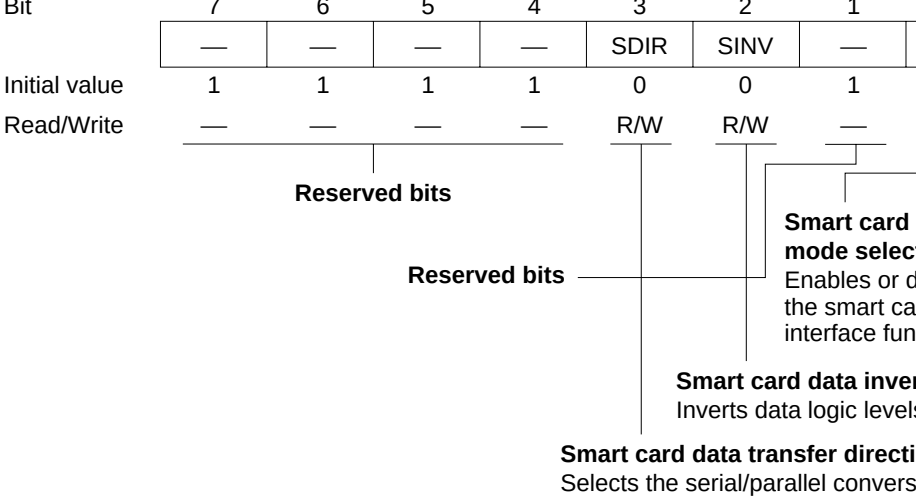
14.1.4 Register Configuration

The smart card interface has the internal registers listed in table 14.2. BRR, TDR, and their normal serial communication interface functions, as described in section 13, Seri Communication Interface.

Table 14.2 Registers

Address*1	Name	Abbreviation	R/W	Initial
H'FFB0	Serial mode register	SMR	R/W	H'00
H'FFB1	Bit rate register	BRR	R/W	H'FF
H'FFB2	Serial control register	SCR	R/W	H'00
H'FFB3	Transmit data register	TDR	R/W	H'FF
H'FFB4	Serial status register	SSR	R/(W)*2	F'84
H'FFB5	Receive data register	RDR	R	H'00
H'FFB6	Smart card mode register	SCMR	R/W	H'F2

Notes: 1. Lower 16 bits of the address.
2. Only 0 can be written, to clear 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.

Bit 3: SDIR	Description	
0	TDR contents are transmitted LSB-first	(I
	Received data is stored LSB-first in RDR	
1	TDR contents are transmitted MSB-first	
	Received data is stored MSB-first 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

14.2.2 Serial Status Register (SSR)

The function of SSR bit 4 is modified in the smart card interface. This change also cau
modification to the setting conditions for bit 2 (TEND).

Bit	7	6	5	4	3	2	1
	TDRE	RDRF	ORER	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

Transmit end
Status flag indicating
end of transmission

Error signal status (ERS)
Status flag indicating that an
error signal has been received

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

	[Clearing conditions] The chip is reset or enters standby 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 p [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 Communication Interface. The setting conditions for transmit end (TEND, bit 2), are modified as follows.

Bit 2: TEND	Description
0	Transmission is in progress [Clearing conditions] Software reads TDRE while it is set to 1, then writes 0 in the TDRE The DMAC writes data in TDR.
1	End of transmission (I [Setting conditions] 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 serial character is transmitted (normal transmission)

Note: An etu (elementary time unit) is the time needed to transmit one bit.

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): Set at 0 when using the regular smart card interface. In GS set to 1. When transmission is complete, initially the TEND flag set timing appears fo clock output restriction mode. Clock output restriction mode comprises serial control and bit 0.

Bit 7: GM	Description
0	Using the regular smart card interface mode • The TEND flag is set 12.5 etu after the beginning of the start bit • Clock output on/off control only
1	Using the GSM mode smart card interface mode • 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 (set by SCR)

Bits 6 to 0—Operate in the same way as for the normal SCI.

For details, see section 13.2.5, Serial Mode Register (SMR).

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

Bits 7 to 2—Operate in the same way as for the normal SCI.

For details, see section 13.2.6, Serial Control Register (SCR).

Bits 1 and 0—Clock Enable (CKE1, CKE0): Setting enable or disable for the SCI clo selection and clock output from the SCK pin. In smart card interface mode, it is possibl between enabling and disabling of the normal clock output, and specify a fixed high lev low level for the clock output.

SMR	SCR		
Bit 7: GM	**Bit 1: CKE1**	**Bit 0: CKE0**	**Description**
0	0	0	The internal clock/SCK_0 pin functions as an I/O port (
0	0	1	The internal clock/SCK_0 pin functions as the clock outpu
1	0	0	The internal clock/SCK_0 pin is fixed at low-level output
1	0	1	The internal clock/SCK_0 pin functions as the clock outpu
1	1	0	The internal clock/SCK_0 pin is fixed at high-level output
1	1	1	The internal clock/SCK_0 pin functions as the clock outpu

the end of the parity bit and the start of the next frame. (An elementary time unit is
required to transmit one bit.)

- In receiving, if a parity error is detected, a low error signal is output for 1 etu, begi
etu after the start bit.
- In transmitting, if an error signal is received, after at least 2 etu, the same data is a
transmitted again.
- Only asynchronous communication is supported. There is no synchronous commu
function.

14.3.2 Pin Connections

Figure 14.2 shows a pin connection diagram for the smart card interface.

In communication with a smart card, data is transmitted and received over the same si
The TxD_0 and RxD_0 pins should both be connected to this line. The data transmission
be pulled up to V_{CC} through a resistor.

If the smart card uses the clock generated by the smart card interface, connect the SCK
to the card's CLK input. If the card uses its own internal clock, this connection is unn

The reset signal should be output from one of the H8/3048B Group's generic ports.

In addition to these pin connections, power and ground connections will normally also
necessary.

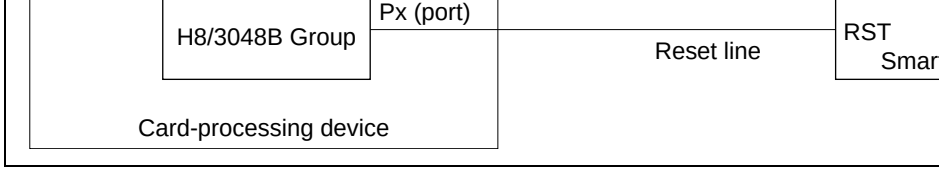


Figure 14.2 Smart Card Interface Connection Diagram

Note: A loop-back test can be performed by setting both RE and TE to 1 without con
smart card.

14.3.3 Data Format

Figure 14.3 shows the data format of the smart card interface. In receive mode, parity i
once per frame. If a parity error is detected, an error signal is returned to the transmittin
request retransmission. In transmit mode, the error signal is sampled and the same data
retransmitted if the error signal is low.

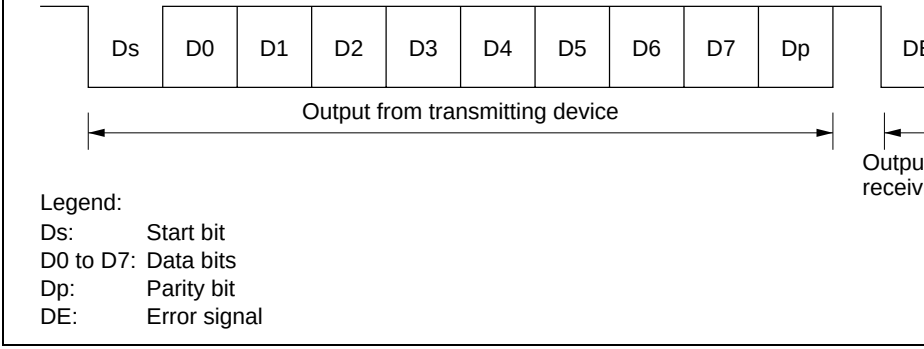


Figure 14.3 Smart Card Interface Data Format

The operating sequence is as follows.

1. When not in use, the data line is in the high-impedance state, and is pulled up to th through a resistor.
2. To start transmitting a frame of data, the transmitting device transmits a low start b followed by eight data bits (D0 to D7) and a parity bit (Dp).
3. Next, in the smart card interface, the transmitting device returns the data line to the impedance state. The data line is pulled up to the high level through a resistor.
4. The receiving device performs a parity check. If there is no parity error, the receiv waits to receive the next data. If a parity error is present, the receiving device outp error signal (DE) to request retransmission of the data. After outputting the error si designated interval, the receiving device returns the signal line to the high-impedan The signal line is pulled back up to the high level through the pull-up resistor.
5. If the transmitting device does not receive an error signal, it proceeds to transmit th If it receives an error signal, it returns to step 2 and transmits the same data again.

SMR	H'FFB0	GM	0	1	O/$\overline{E}$	1	0	CKS1
BRR	H'FFB1	BRR7	BRR6	BRR5	BRR4	BRR3	BRR2	BRR1
SCR	H'FFB2	TIE	RIE	TE	RE	0	0	CKE1
TDR	H'FFB3	TDR7	TDR6	TDR5	TDR4	TDR3	TDR2	TDR1
SSR	H'FFB4	TDRE	RDRF	ORER	ERS	PER	TEND	0
RDR	H'FFB5	RDR7	RDR6	RDR5	RDR4	RDR3	RDR2	RDR1
SCMR	H'FFB6	—	—	—	—	SDIR	SINV	—

Legend: —: Unused bit.

Notes: 1. Lower 16 bits of the address.

2. When the GM of the SMR is set at 0, be sure the CKE1 bit is 0.

Serial Mode Register (SMR) Settings: In regular smart card interface mode, set the G
In regular smart card mode, clear the GM bit to 0. In GSM mode, set the GM bit to 1. C
O/$\overline{E}$ bit to 0 if the smart card uses the direct convention. Set the O/$\overline{E}$ bit to 1 if the smar
the inverse convention. Bits CKS1 and CKS0 select the clock source of the built-in bau
generator. See section 14.3.5, Clock.

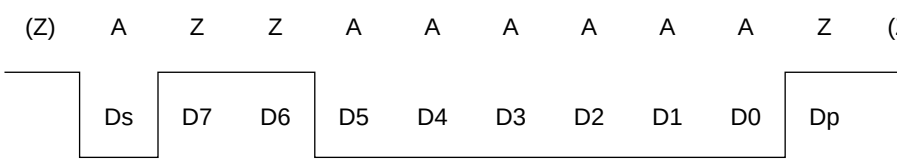
Bit Rate Register (BRR) Settings: This register sets the bit rate. Equations for calcula
setting are given in section 14.3.5, Clock.

Serial Control Register (SCR): The TIE, RIE, TE, and RE bits have their normal seri
communication functions. For details, see section 13, Serial Communication Interface.
and CKE0 bits select clock output. When the GM bit of the SMR is cleared to 0, to disa
output, clear this bit to 00. To enable clock output, set this bit to 01. When the GM bit
is set to 1, clock output is enabled. Clock output is fixed at high or low.

Smart Card Mode Register (SCMR): If the smart card follows the direct convention,
SDIR and SINV bits to 0. If the smart card follows the indirect convention, set the SDI
SINV bits to 1. To use the smart card interface, set the SMIF bit to 1.

In the direct convention, state Z corresponds to logic level 1, and state A to logic l
Characters are transmitted and received LSB-first. In the example above the first c
is H'3B. The parity bit is 1, following the even parity rule designated for smart car

- Inverse convention (SDIR = SINV = O/$\overline{E}$ = 1)



In the inverse convention, state A corresponds to the logic level 1, and state Z to th
0. Characters are transmitted and received MSB-first. In the example above the fir
data is H'3F. Following the even parity rule designated for smart cards, the parity b
level is 0, corresponding to state Z.

In the H8/3048B Group, the SINV bit inverts only the data bits D7 to D0. The parity b
inverted, so the O/$\overline{E}$ bit in SMR must be set to odd parity mode. This applies in both t
and receiving.

$$B = \frac{\phi}{1488 \times 2^{2n-1} \times (N + 1)} \times 10^6$$

where, N: BRR setting ($0 \leq N \leq 255$)
B: Bit rate (bits/s)
ϕ: System clock frequency (MHz)*
n: See table 14.4

Table 14.4 n-Values of CKS1 and CKS0 Settings

n	CKS1	CKS0
0	0	0
1	0	1
2	1	0
3	1	1

Note: * If the gear function is used to divide the system clock frequency, use the div
frequency to calculate the bit rate. The equation above applies directly to 1/
division.

Table 14.5 Bit Rates (bits/s) for Different 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		20.0	
Bit/s	**N**	**Error**	**N**	**Error**	**N**	**Error**	**N**	**Error**	**N**	**Error**	**N**	**Error**	**N**	**Error**	**N**	**Er**
9600	0	0.00	1	30.00	1	25.00	1	8.99	1	0.00	1	12.01	2	15.99	2	6.6

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 calculated from the following equation.

$$\text{Error (\%)} = \left\{ \frac{\phi}{1488 \times 2^{2n-1} \times B \times (N + 1)} \times 10^6 - 1 \right\} \times 100$$

2. Clear the ERS, PER, and ORER error flags to 0 in the serial status register (SSR).
3. Set the parity mode bit (O/$\overline{E}$) and baud rate generator clock source select bits (CKS
CKS0) as required in the serial mode register (SMR). At the same time, clear the C/
and MP bits to 0, and set the STOP and PE bits to 1.
4. Set the SMIF, SDIR, and SINV bits as required in the smart card mode register (SC
When the SMIF bit is set to 1, the TxD_0 and RxD_0 pins switch from their I/O port fu
their serial communication interface functions, and are placed in the high-impedanc
5. Set a value corresponding to the desired bit rate in the bit rate register (BRR).
6. Set clock enable bit 0 (CKE0) as required in the serial control register (SCR). Write
TIE, RIE, TE, RE, MPIE, TEIE, and CKE1 bits. If bit CKE0 is set to 1, a serial cloc
output from the SCK_0 pin.
7. Wait for at least the interval required to transmit or receive one bit, then set the TIE
and RE bits as necessary in SCR. Do not set TE and RE both to 1, except when perf
loop-back test.

3. Check that the TEND flag is set to 1 in SSR. Repeat steps 2 and 3 until this check
4. Write transmit data in TDR and clear the TDRE flag to 0. The data will be transmi
   TEND flag will be cleared to 0.
5. To continue transmitting data, return to step 2.
6. To terminate transmission, clear the TE bit to 0.

This procedure may include interrupt handling and DMA transfer.

If the TIE bit is set to 1 to enable interrupt requests, when transmission is completed a
TEND flag is set to 1, a transmit-data-empty interrupt (TXI) is requested. If the RIE b
to enable interrupt requests, when a transmit error occurs and the ERS flag is set to 1,
transmit/receive-error interrupt (ERI) is requested.

The timing of TEND flag setting depends on the GM bit in SMR. The timing is shown
14.6.

If the TXI interrupt activates the DMAC, the number of bytes designated in the DMA
transmitted automatically, including automatic retransmit.

For details, see Interrupt Operations and Data Transfer by DMAC in this section.

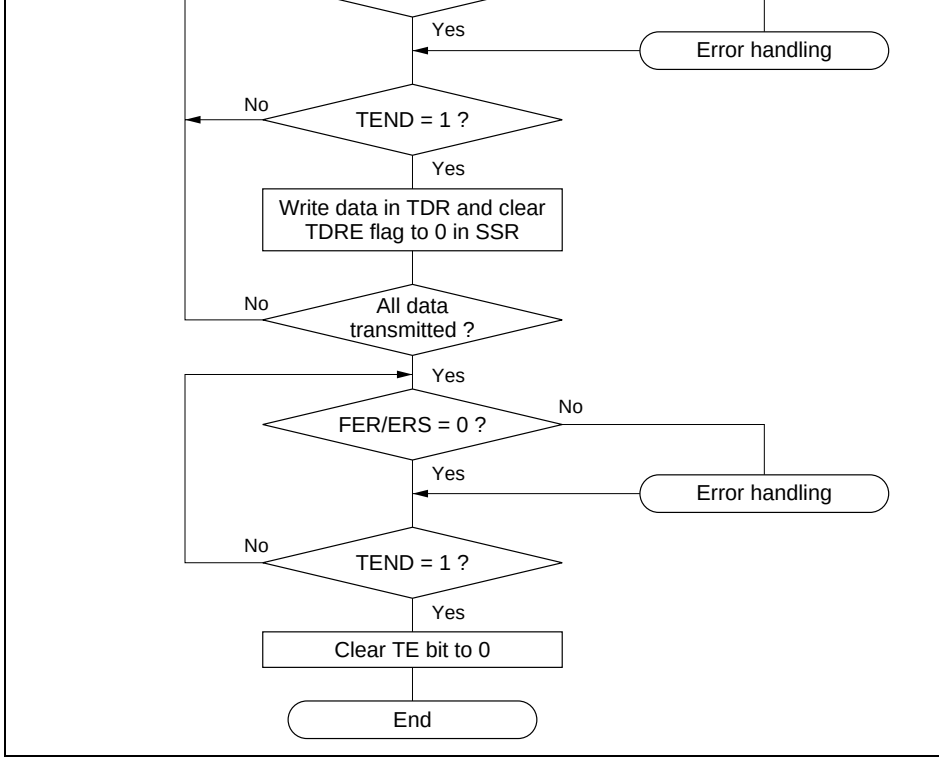


Figure 14.4 Transmit Flowchart (Example)

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 until the TEND

Note: When the ERS flag is set, it should be cleared until transfer of the last bit (D7 in LS transmission, D0 in MSB-first transmission) of the next transfer data to be transmi been completed.

Figure 14.5 Relation Between Transmit Operation and Internal Regist

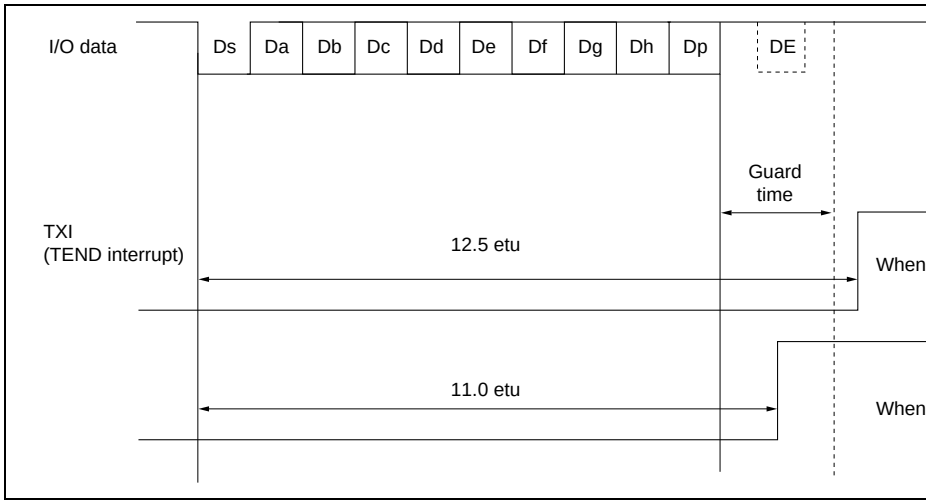


Figure 14.6 TEND Flag Occurrence Timing

3. Check that the RDRF flag is set to 1. Repeat steps 2 and 3 until this check passes.
4. Read receive data from RDR.
5. To continue receiving data, clear the RDRF flag to 0 and return to step 2.
6. To terminate receiving, clear the RE bit to 0.

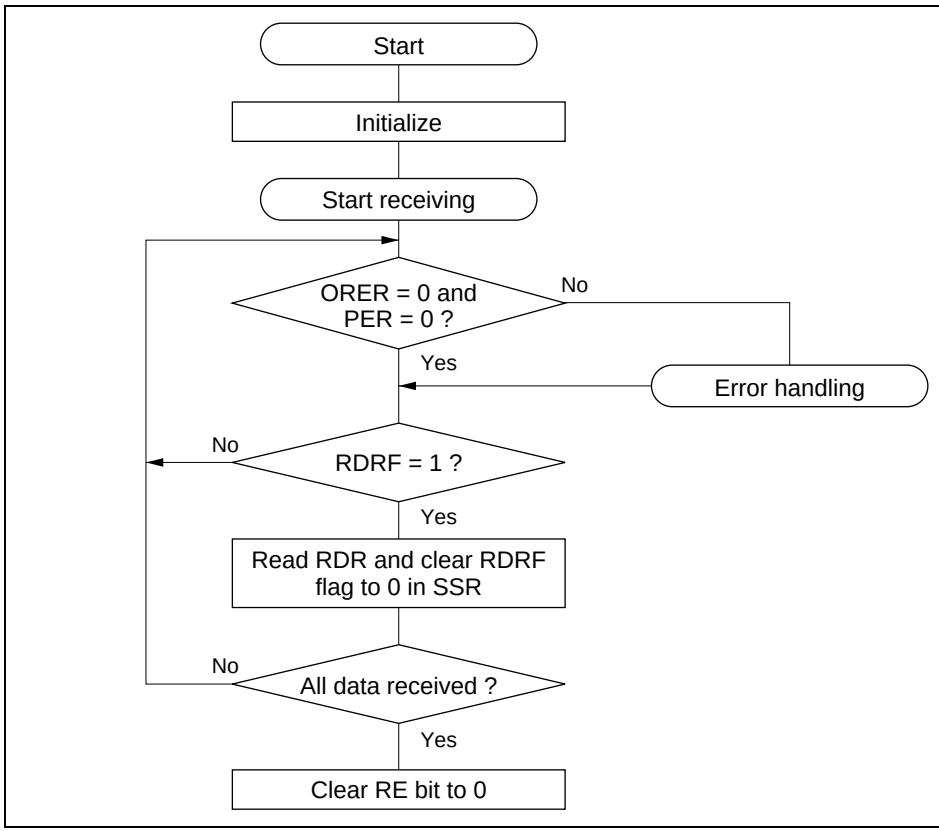


Figure 14.7 Receive Flowchart (Example)

For details, see Interrupt Operations and Data Transfer by DMAC below.

When a parity error occurs and PER is set to 1, the receive data is transferred to RDR, erroneous data can be read.

Switching Modes

To switch from receive mode to transmit mode, check that receiving operations have c then initialize the smart card interface, clearing RE to 0 and setting TE to 1. Completi operations is indicated by the RDRF, PER, or ORER flag.

To switch from transmit mode to receive mode, check that transmitting operations hav completed, then initialize the smart card interface, clearing TE to 0 and setting RE to 1 Completion of transmit operations can be verified from the TEND flag.

Fixing Clock Output

When the GM bit of the SMR is set to 1, clock output is fixed by CKE1 and CKE0 of case, the clock pulse can be set at minimum value.

Figure 14.8 shows clock output fixed timing: CKE0 is restricted with GM = 1 and CK

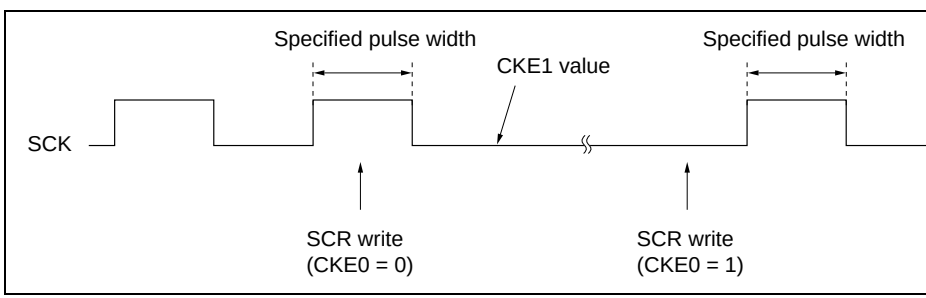


Figure 14.8 Clock Output Fixed Timing

Table 14.8 Smart Card Mode Operating States and Interrupt Sources

Operating State		Flag	Mask Bit	Interrupt Source	DM Ac
Transmit mode	Normal operation	TEND	TIE	TXI	Av
	Error	ERS	RIE	ERI	No
Receive mode	Normal operation	RDRF	RIE	RXI	Av
	Error	PER, ORER	RIE	ERI	No

Data Transfer by DMAC

The DMAC can be used to transmit and receive in smart card mode, as in normal SCI c
In transmit mode, when the TEND flag is set to 1 in SSR, the TDRE flag is set simultar
generating a TXI interrupt. If TXI is designated in advance as a DMAC activation sour
DMAC will be activated by the TXI request and will transfer the next transmit data. Th
transfer by the DMAC automatically clears the TDRE and TEND flags to 0. When an e
the SCI automatically retransmits the same data, keeping TEND cleared to 0 so that the
not activated. The SCI and DMAC will therefore automatically transmit the designated
bytes, including retransmission when an error occurs. When an error occurs the ERS fla
cleared automatically, so the RIE bit should be set to 1 to enable the error to generate a
request, and the ERI interrupt handler 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 8, DMA Controller.

In receive operations, when the RDRF flag is set to 1 in SSR, an RXI interrupt is reque
is designated in advance as a DMAC activation source, the DMAC will be activated by
request and will transfer the received data. This data transfer by the DMAC automatica
the RDRF flag to 0. When an error occurs, the RDRF flag is not set and an error flag is
The DMAC is not activated. The ERI interrupt request is directed to the CPU. The ERI
handler should clear the error flags.

2. Write 0 to the TE and RE bits in the serial control register (SCR) to stop transi operations. At the same time, set the CKE1 bit to the value for the fixed output software standby mode.
3. Write 0 to the CKE0 bit in SCR to stop the clock.
4. Wait for one serial clock cycle. During this period, the duty cycle is preserved output is fixed at the specified level.
5. Write H'00 to the serial mode register (SMR) and smart card mode register (SC
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 so standby (the current $P9_4$ pin state).
  3. Set smart card interface mode and output the clock. Clock signal generation is the normal duty cycle.

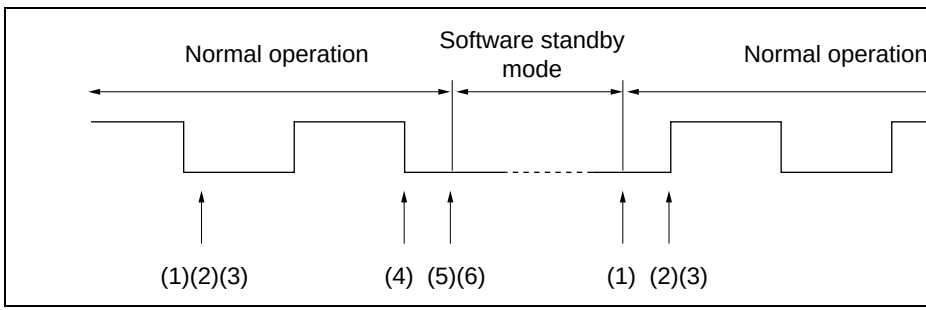


Figure 14.9 Procedure for Stopping and Restarting the Clock

14.4 Usage Notes

When using the SCI as a smart card interface, note the following points.

Receive Data Sampling Timing in Smart Card Mode and Receive Margin

In smart card mode the SCI operates on a base clock with 372 times the bit rate frequen
receiving, the SCI synchronizes internally with the fall of the start bit, which it samples
base clock. Receive data is latched at the rising edge of the 186th base clock pulse. See
14.10.

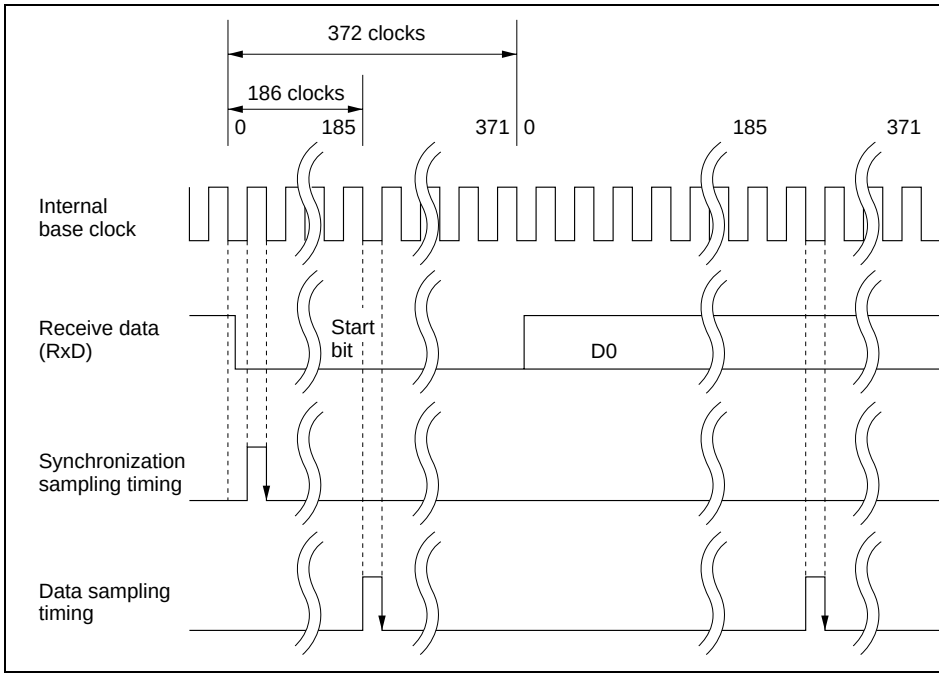


Figure 14.10 Receive Data Sampling Timing in Smart Card Mode

D: Clock duty cycle (D = 0 to 1.0)
L: Frame length (L = 10)
F: Absolute deviation of clock frequency

From this equation, if F = 0 and D = 0.5 the receive margin is as follows.

D = 0.5, F = 0
M = $\{0.5 - 1/(2 \times 372)\} \times 100\%$
= 49.866%

Retransmission

Retransmission is described below for the separate cases of transmit mode and receive

- Retransmission when SCI is in Receive Mode (see figure 14.11)
  (1) The SCI checks the received parity bit. If it detects an error, it automatically se flag to 1. If the RIE bit in SCR is set to the enable state, an ERI interrupt is req PER flag should be cleared to 0 in SSR before the next parity bit sampling timi
  (2) The RDRF bit in SSR is not set to 1 for the error frame.
  (3) If an error is not detected when the parity bit is checked, the PER flag is not se
  (4) If an error is not detected when the parity bit is checked, receiving operations a to have ended normally, and the RDRF bit is automatically set to 1 in SSR. If t 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 t reads the RDR data, it automatically clears RDRF to 0.
  (5) When a normal frame is received, at the error signal transmit timing, the data p the high-impedance state.

Figure 14.11 Retransmission in SCI Receive Mode

- Retransmission when SCI is in Transmit Mode (see figure 14.12)

  (6) After transmitting one frame, if the receiving device returns an error signal, the
  ERS flag to 1 in SSR. If the RIE bit in SCR is set to the enable state, an ERI int
  requested. The ERS flag should be cleared to 0 in SSR before the next parity bit
  timing.

  (7) The TEND bit in SSR is not set for the frame in which the error signal was rece
  indicating an error.

  (8) If no error signal is returned from the receiving device, the ERS flag is not set i

  (9) If no error signal is returned from the receiving device, transmission of the fram
  retransmission, is assumed to be complete, and the TEND bit is set to 1 in SSR.
  bit in SCR is set to the enable state, a TXI interrupt is requested. If TXI is enabl
  DMA transfer activation source, the next data can be written in TDR automatica
  the DMAC writes data in TDR, it automatically clears the TDRE bit to 0.

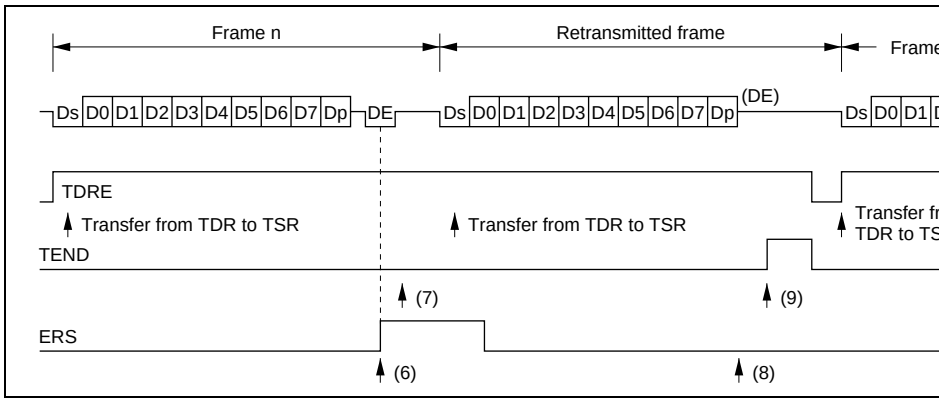


Figure 14.12 Retransmission in SCI Transmit Mode

When the A/D converter is not used, it can be halted independently to conserve power 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

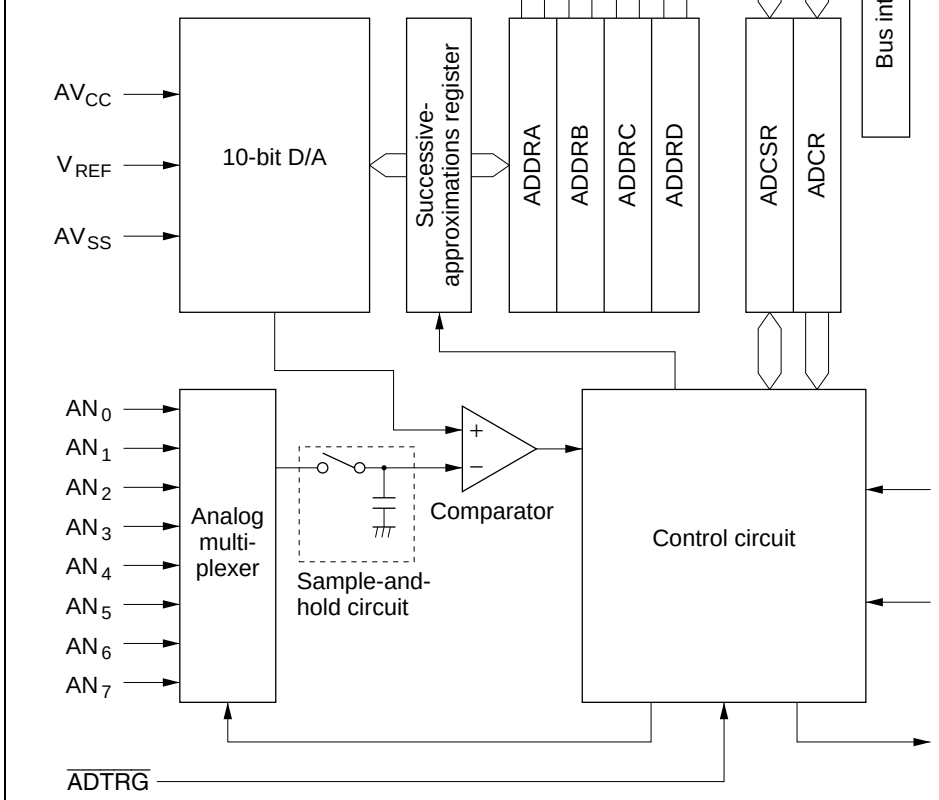
  Conversion time: Minimum 5.36 µs per channel (with 25-MHz system clock)
- Two conversion modes

  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
- A/D conversion can be externally triggered
- A/D interrupt requested at end of conversion

  At the end of A/D conversion, an A/D end interrupt (ADI) can be requested.



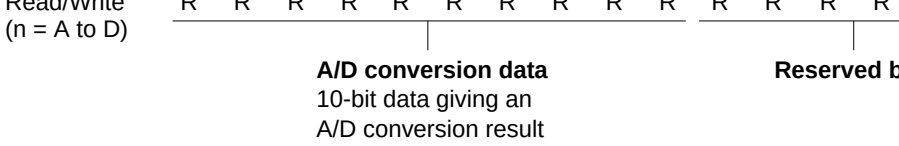
Legend:
ADCR: A/D control register
ADCSR: A/D control/status register
ADDRA: A/D data register A
ADDRB: A/D data register B
ADDRC: A/D data register C
ADDRD: A/D data register D

Figure 15.1 A/D Converter Block Diagram

Pin Name	viation	I/O	Function
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'FFE2	A/D data register B (high)	ADDRBH	R	H'0
H'FFE3	A/D data register B (low)	ADDRBL	R	H'0
H'FFE4	A/D data register C (high)	ADDRCH	R	H'0
H'FFE5	A/D data register C (low)	ADDRCL	R	H'0
H'FFE6	A/D data register D (high)	ADDRDH	R	H'0
H'FFE7	A/D data register D (low)	ADDRDL	R	H'0
H'FFE8	A/D control/status register	ADCSR	R/(W)*2	H'0
H'FFE9	A/D control register	ADCR	R/W	H'7

Notes: 1. Lower 16 bits of the address
2. Only 0 can be written in bit 7, to clear the flag.
3. Initial value is H'7F in mask ROM versions, PROM versions, and dual power flash memory versions.



The four A/D data registers (ADDRA to ADDRD) are 16-bit read-only registers that s
results of A/D conversion.

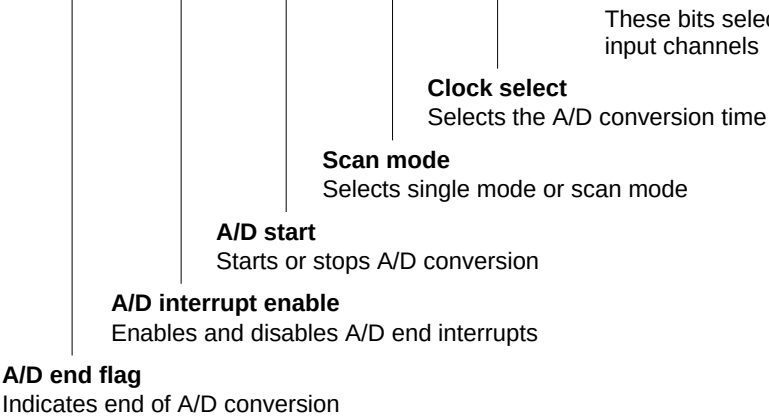
An A/D conversion produces 10-bit data, which is transferred for storage into the A/D
register corresponding to the selected channel. The upper 8 bits of the result are stored
byte of the A/D data register. The lower 2 bits are stored in the lower byte. Bits 5 to 0
data register are reserved bits that 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 condition]	(In
	Cleared by reading ADF while ADF = 1, then writing 0 in ADF	
1	[Setting conditions]	
	Single mode: A/D conversion ends	
	Scan mode: A/D conversion ends in all selected channels	

A/D conversion. It can also be set to 1 by external trigger input at the $\overline{\text{ADTRG}}$ pin.

Bit 5: ADST	Description
0	A/D conversion is stopped
1	Single mode: A/D conversion starts; ADST is automatically cleared conversion ends. Scan mode: A/D conversion starts and continues, cycling among t channels, until ADST is cleared to 0 by software, by a reset, or by to 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

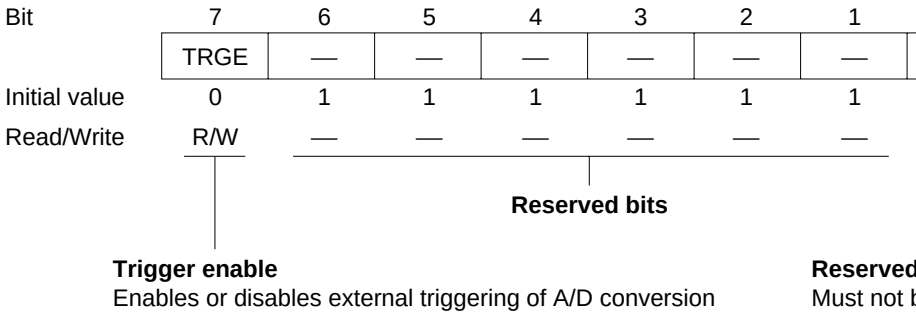
Bit 3—Clock Select (CKS): Selects the A/D conversion time. Clear the ADST bit to switching the conversion time.

Bit 3: CKS	Description
0	Conversion time = 266 states (maximum)
1	Conversion time = 134 states (maximum)

Bits 2 to 0—Channel Select 2 to 0 (CH2 to CH0): These bits and the SCAN bit sele input channels. Clear the ADST bit to 0 before changing the channel selection.

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

15.2.3 A/D Control Register (ADCR)



ADCR is an 8-bit readable/writable register that enables or disables external triggering conversion. ADCR is initialized to H'7E by a reset and in standby mode.

Bit 7—Trigger Enable (TRGE): Enables or disables external triggering of A/D conve

Bit 7: TRGE	Description
0	A/D conversion cannot be externally triggered (I
1	A/D conversion starts at the falling edge of the external trigger signa

Bits 6 to 1—Reserved: Read-only bits, always read as 1.

Bit 0—Reserved: Do not set to 1.

When reading an A/D data register, always read the upper byte before the lower byte.
to read only the upper byte, but if only the lower byte is read, incorrect data may be ob

Figure 15.2 shows the data flow for access to an A/D data register.

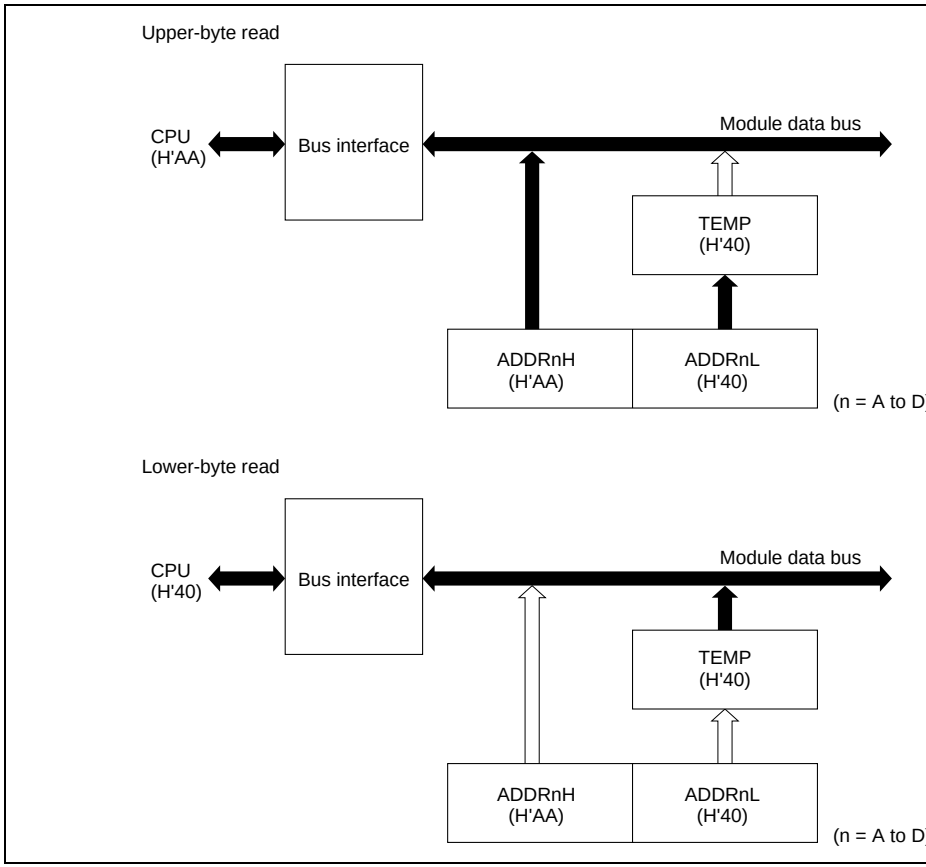


Figure 15.2 A/D Data Register Access Operation (Reading H'AA40)

conversion starts when the ADST bit is set to 1 by software, or by external trigger inpu
ADST bit remains set to 1 during A/D conversion and is automatically cleared to 0 whe
conversion ends.

When conversion ends the ADF bit is set to 1. If the ADIE bit is also set to 1, an ADI i
requested at this time. To clear the ADF flag to 0, first read ADCSR, then write 0 in AI

When the mode or analog input channel must be switched during analog conversion, to
incorrect operation, first clear the ADST bit to 0 in ADCSR to halt A/D conversion. Af
the necessary changes, set the ADST bit to 1 to start A/D conversion again. The ADST
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 next.

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 (
2. When A/D conversion is completed, the result is transferred into ADDRB. At the sa
   the ADF flag is set to 1, the ADST bit is cleared to 0, and the A/D converter becom
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 is
   A/D conversion starts again and steps 2 to 7 are repeated.

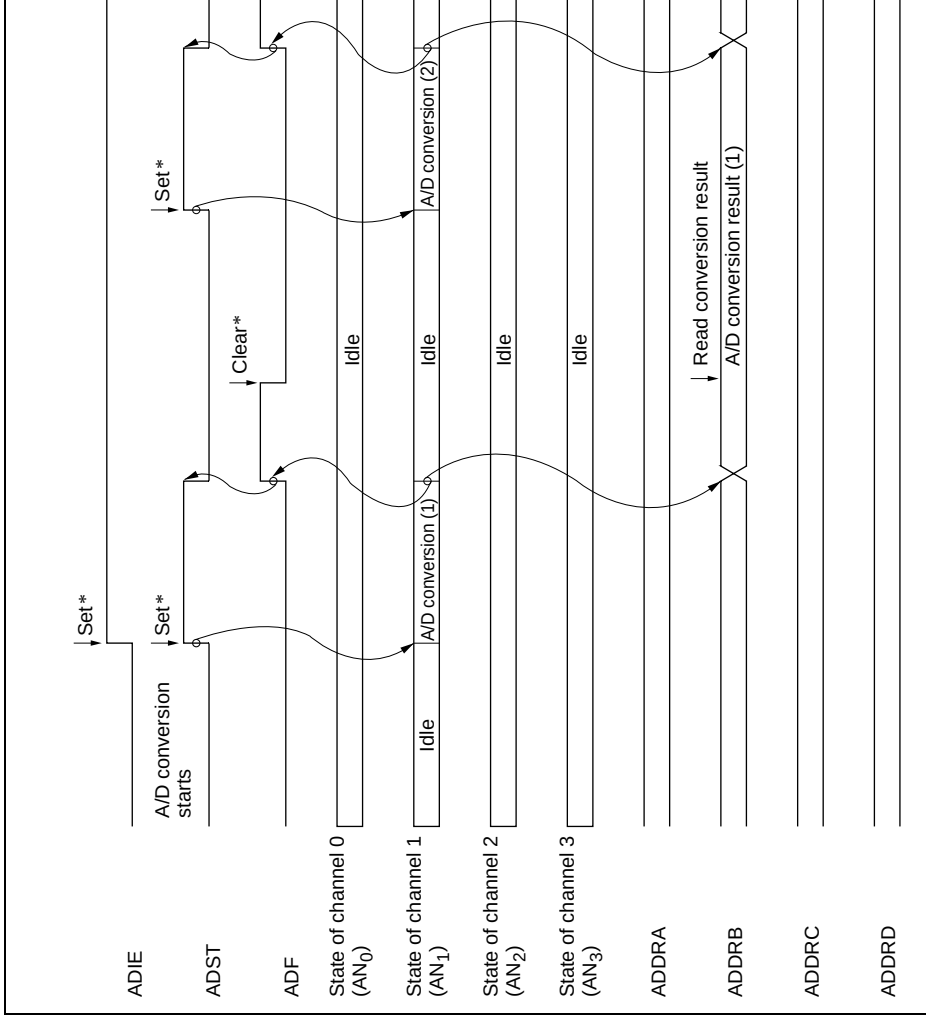


Figure 15.3 Example of A/D Converter Operation (Single Mode, Channel 1 S

corresponding to the channels.

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 conver
making the necessary changes, set the ADST bit to 1. A/D conversion will start again f
first channel in the group. The ADST bit can be set at the same time as the mode or cha
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 inpu
   AN_0 to AN_2 are selected (CH1 = 1, CH0 = 0), and A/D conversion is started (ADST
2. When A/D conversion of the first channel (AN_0) is completed, the result is transfer
   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 flag
   and conversion of the first channel (AN_0) starts again. If the ADIE bit is set to 1, an
   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 conv
   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 second subsequent conversions the conversion time is fixed at 256 states when CKS = 0 or 128 when CKS = 1.

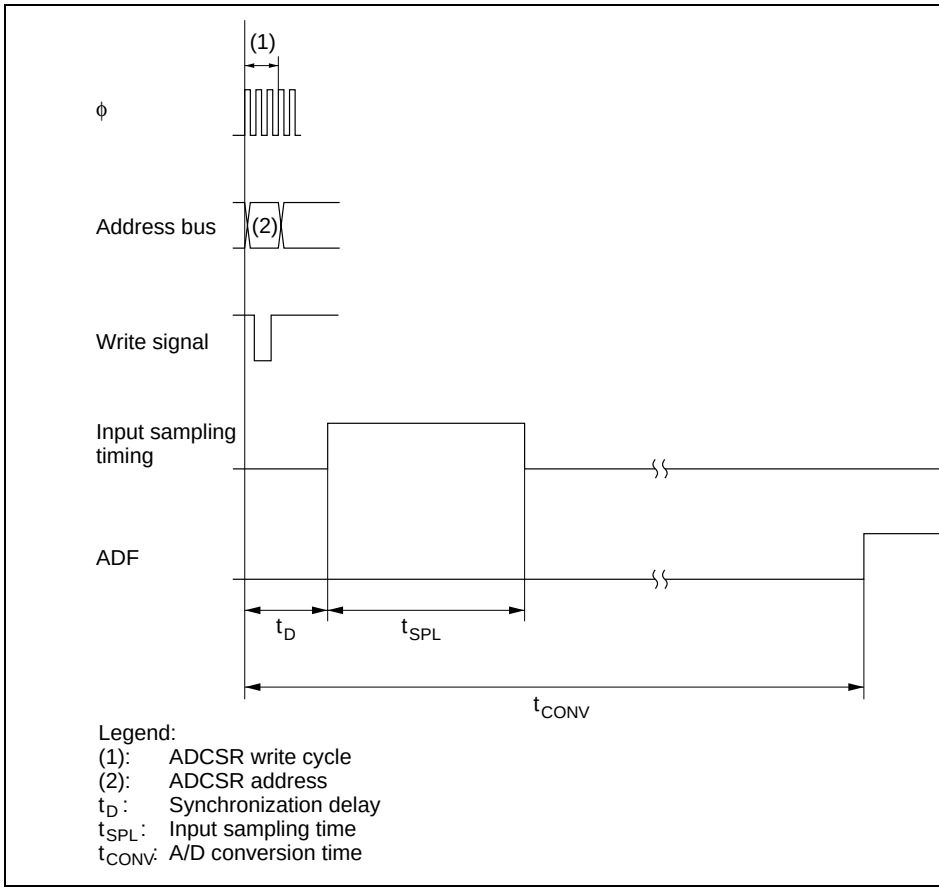


Figure 15.5 A/D Conversion Timing

15.4.4 External Trigger Input Timing

A/D conversion can be externally triggered. When the TRGE bit is set to 1 in ADCR, trigger input is enabled at the $\overline{\text{ADTRG}}$ pin. A high-to-low transition at the $\overline{\text{ADTRG}}$ pi ADST bit to 1 in ADCSR, starting A/D conversion. Other operations, in both single a modes, are the same as if the ADST bit had been set to 1 by software. Figure 15.6 sho timing.

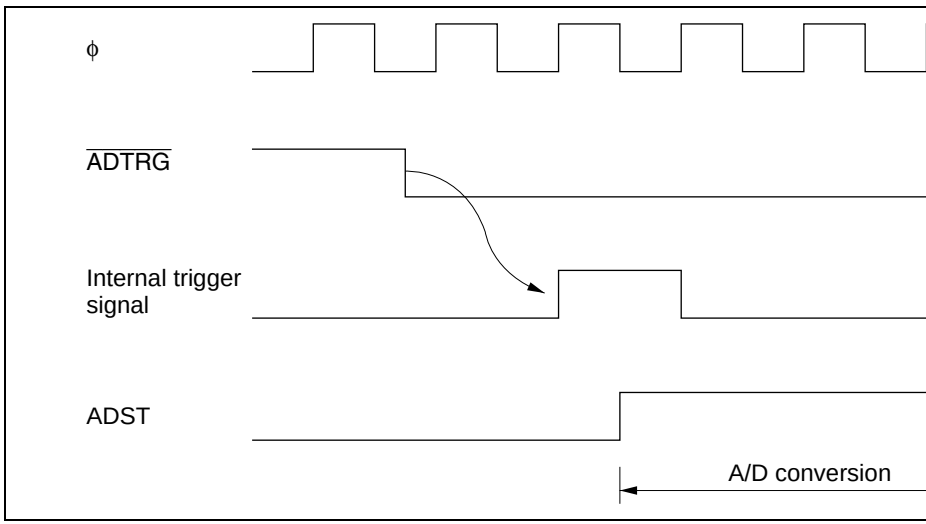


Figure 15.6 External Trigger Input Timing

1. Analog Input Voltage Range: During A/D conversion, the voltages input to the anal 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} should be follows: $AV_{SS} = V_{SS}$. AV_{CC} and AV_{SS} must not be left open, even if the A/D converte used.

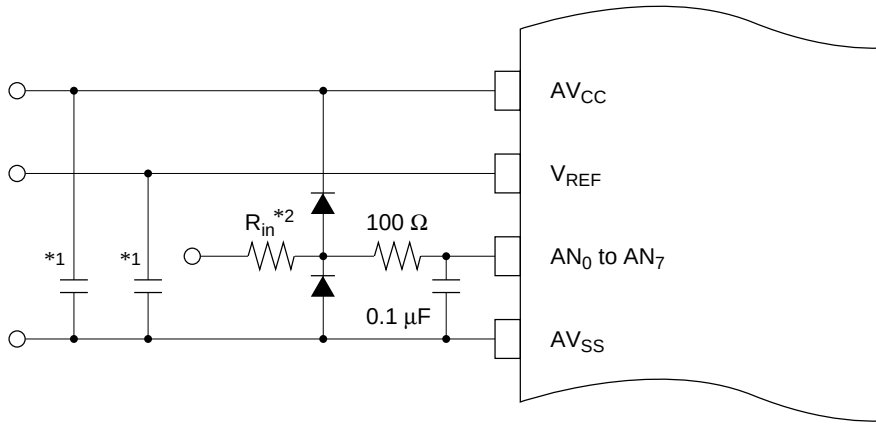
3. V_{REF} Programming Range: The reference voltage input at the V_{REF} pin should be in t $V_{REF} \leq AV_{CC}$.

Failure to observe points 1, 2, and 3 above may degrade chip reliability.

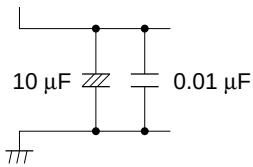
4. Note on Board Design: In board layout, separate the digital circuits from the analog much as possible. Particularly avoid layouts in which the signal lines of digital circu closely approach the signal lines of analog circuits. Induction and other effects may analog circuits to operate incorrectly, or may adversely affect the accuracy of A/D c

   The analog input signals (AN_0 to AN_7), analog reference voltage (V_{REF}), and analog voltage (AV_{CC}) must be separated from digital circuits by the analog ground (AV_{SS}). analog ground (AV_{SS}) should be connected to a stable digital ground (V_{SS}) at one po board.

than that input to the analog input pins via input impedance R_{in}. The circuit constan
therefore be selected carefully.



Notes: 1. Numeric values are approximate.



2. R_{in}: input impedance

Figure 15.7 Example of Analog Input Protection Circuit

Figure 15.8 Analog Input Pin Equivalent Circuit

Table 15.5 Analog Input Pin Ratings

Item		Min	Max	U
Analog input capacitance		—	20	p
Allowable signal-source impedance	$\phi \leq 13$ MHz	—	10	k
	$\phi > 13$ MHz	—	5	k

6. A/D Conversion Accuracy Definitions: A/D conversion accuracy in the H8/3048B G defined as follows:
   - Resolution

     Digital output code length of A/D converter
   - Offset error

     Deviation from ideal A/D conversion characteristic of analog input voltage requ raise digital output from minimum voltage value B'0000000000 to B'000000000 15.10)
   - Full-scale error

     Deviation from ideal A/D conversion characteristic of analog input voltage requ raise digital output from B'1111111110 to B'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 range from zero volts to f exclusive of offset error, full-scale error, and quantization error.
   - Absolute accuracy

     Deviation of digital value from analog input value, including offset error, full-sc quantization error, and nonlinearity error.

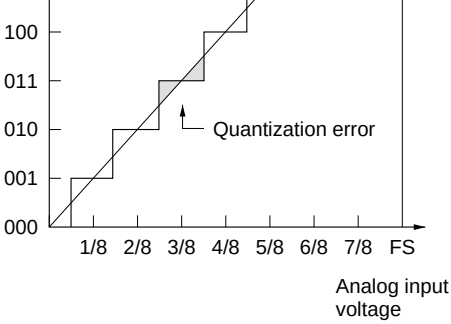


Figure 15.9 A/D Converter Accuracy Definitions (1)

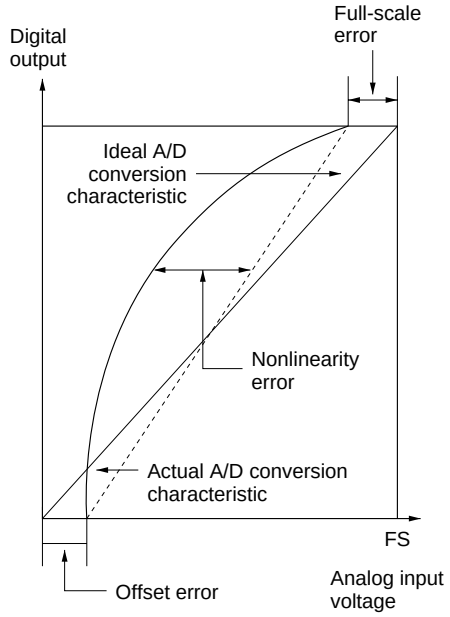


Figure 15.10 A/D Converter Accuracy Definitions (2)

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.

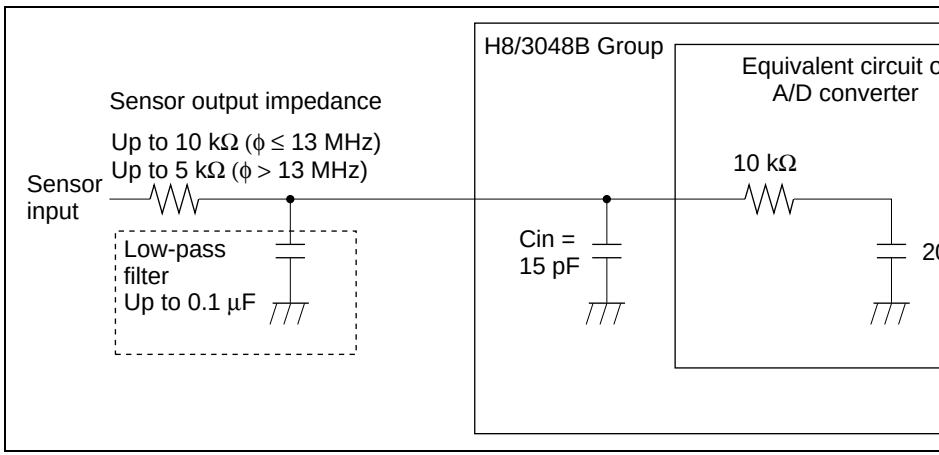


Figure 15.11 Analog Input Circuit (Example)

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 $255/256 \times V_{REF}$
- D/A outputs can be sustained in software standby mode

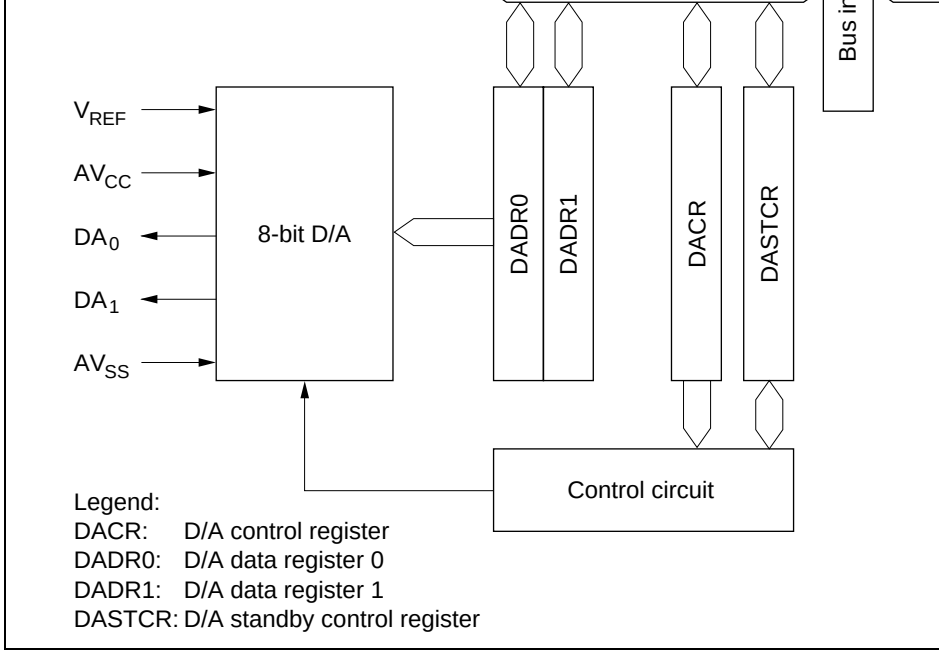


Figure 16.1 D/A Converter Block Diagram

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	I
H'FFDC	D/A data register 0	DADR0	R/W	H
H'FFDD	D/A data register 1	DADR1	R/W	H
H'FFDE	D/A control register	DACR	R/W	H
H'FF5C	D/A standby control register	DASTCR	R/W	H

Note: * Lower 16 bits of the address

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

The D/A data registers (DADR0 and DADR1) are 8-bit readable/writable registers that data to be converted. When analog output is enabled, the D/A data register values are c 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.

16.2.2 D/A Control Register (DACR)

Bit	7	6	5	4	3	2	1
	DAOE1	DAOE0	DAE	—	—	—	—
Initial value	0	0	0	1	1	1	1
Read/Write	R/W	R/W	R/W	—	—	—	—

D/A enable
Controls D/A conversion

D/A output enable 0
Controls D/A conversion and analog output

D/A output enable 1
Controls D/A conversion and analog output

DACR is an 8-bit readable/writable register that controls the operation of the D/A conv DACR is initialized to H'1F by a reset and in standby mode.

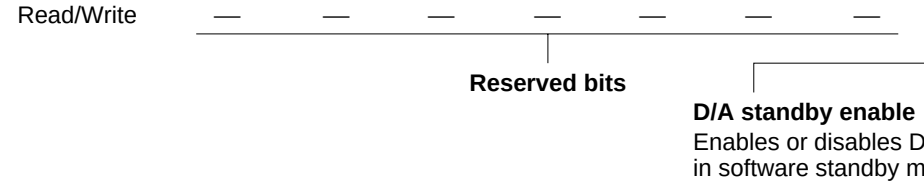
Bit 6: DAOE0	Description
0	DA_0 analog output is disabled
1	Channel-0 D/A conversion and DA_0 analog output are enabled

Bit 5—D/A Enable (DAE): Controls D/A conversion, together with bits DAOE0 and
When the DAE bit is cleared to 0, analog conversion is controlled independently in ch
1. When the DAE bit is set to 1, analog conversion is controlled together in channels 0
Output of the conversion results is always controlled independently by DAOE0 and D

Bit 7: DAOE1	Bit 6: DAOE0	Bit 5: DAE	Description
0	0	—	D/A conversion is disabled in channels 0
	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 a
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 a
	1	—	D/A conversion is enabled in channels 0 a

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 d
and D/A conversion.

Bits 4 to 0—Reserved: Read-only bits, always read as 1.



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

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

Bit 0—D/A Standby Enable (DASTE): Enables or disables D/A output in software st
mode.

Bit 0: DASTE	Description	
0	D/A output is disabled in software standby mode	(In
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 figu

1. Data to be converted is written in DADR0.
2. Bit DAOE0 is set to 1 in DACR. D/A conversion starts and DA_0 becomes an outpu converted result is output after the conversion time. The output value is (DADR0 c $\times V_{REF}$. Output of this conversion result continues until the value in DADR0 is mod DAOE0 bit is cleared to 0.
3. If the DADR0 value is modified, conversion starts immediately, and the result is o the conversion time.
4. When the DAOE0 bit is cleared to 0, DA_0 becomes an input pin.

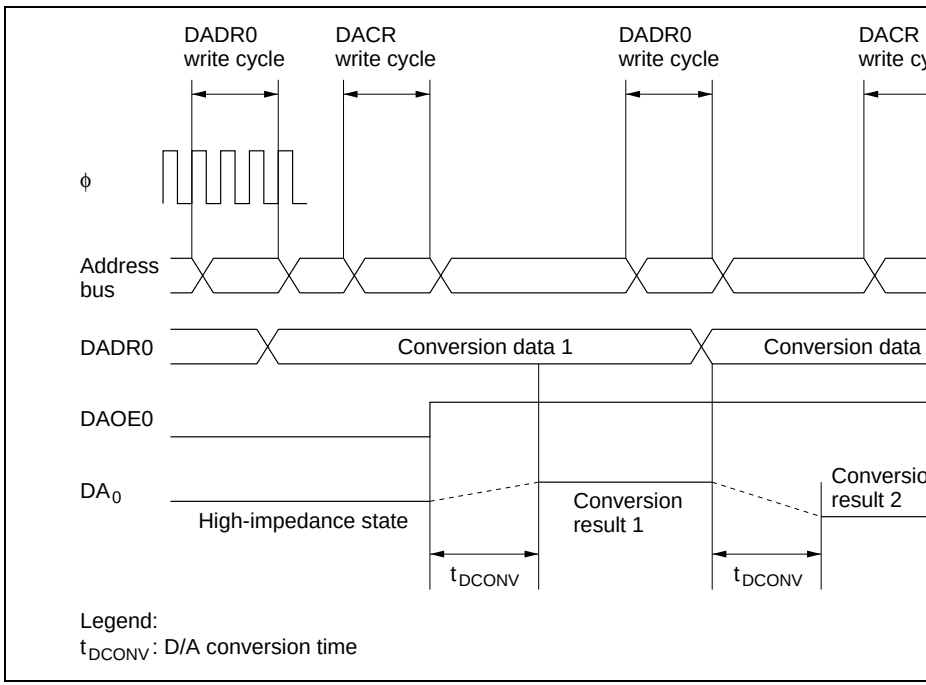
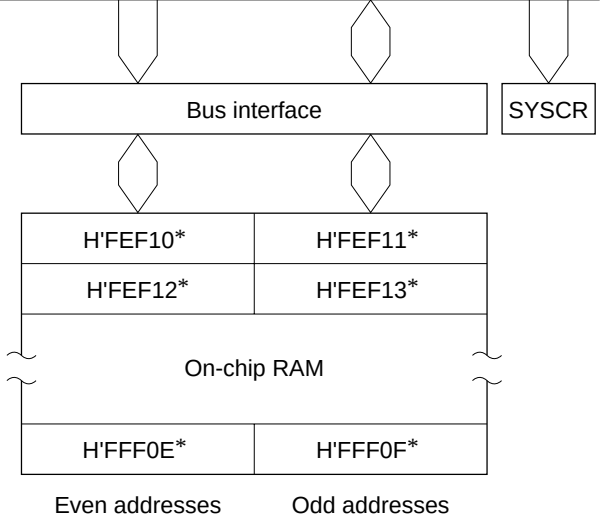


Figure 16.2 Example of D/A Converter Operation

When D/A output is enabled in software standby mode, the reference supply current is during normal operation.

The on-chip RAM of the H8/3048B Group is assigned to addresses H'FEF10 to H'FF
1, 2, 5, and 7, and to addresses H'FFEF10 to H'FFFF0F in modes 3, 4, and 6. The RA
(RAME) in the system control register (SYSCR) can enable or disable the on-chip RA



Legend:
SYSCR: System control register

Note: * This example is of the operating in mode 7. The lower 20 bits of the address ar

Figure 17.1 RAM Block Diagram

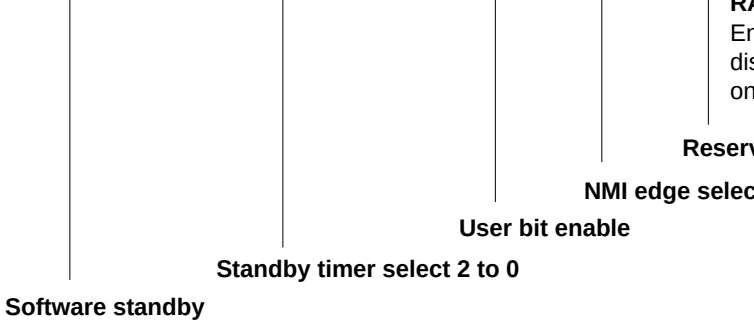
17.1.2 Register Configuration

The on-chip RAM is controlled by SYSCR. Table 17.1 gives the address and initial val
SYSCR.

Table 17.1 System Control Register

Address*	Name	Abbreviation	R/W	Init
H'FFF2	System control register	SYSCR	R/W	H'0

Note: * Lower 16 bits of the address.



One function of SYSCR is to enable or disable access to the on-chip RAM. The on-ch
enabled or disabled by the RAME bit in SYSCR. For details about the other bits, see s
System Control Register (SYSCR).

Bit 0—RAM Enable (RAME): Enables or disables the on-chip RAM. The RAME bi
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	(

Since the on-chip RAM is connected to the CPU by an internal 16-bit data bus, it can b
and read by word access. It can also be written and read by byte access. Byte data is ac
two states using the upper 8 bits of the data bus. Word data starting at an even address i
in two states using all 16 bits of the data bus.

18.1.1 Notes on H8/3048F-ONE (Single Power Supply)

There are two models of the H8/3048F-ZTAT with on-chip flash memory: a dual pow
model (H8/3048F) and single power supply model (H8/3048F-ONE). Points to be not
using the H8/3048F-ONE single power supply is given below.

For the differences between the dual power supply model and single power supply mo
(H8/3048F-ONE), see section 1.4.3, Differences between H8/3048F and H8/3048F-O

(1) Voltage Application

**12 V must not be applied to the H8/3048F-ONE (single power supply), as this wil
permanently damage the device.**

The flash memory programming power supply for the H8/3048F-ONE (single power s
V_{CC}.
The programming power supply for the dual power supply model is the V_{PP} pin (12 V)
no V_{PP} pin in the single power supply model. In the H8/3048F-ONE (single power sup
the FWE pin is provided at the same pin position as the V_{PP} pin in the dual power sour
but FWE is not a power supply pin—it is used to control flash memory write enabling
Also, in boot mode, 12 V must be applied to the MD_2 pin in the dual power supply mo
is not necessary in the H8/3048F-ONE (single power supply).
**The maximum rating of the FWE and MD_2 pins in the H8/3048F-ONE (single po
is V_{CC} +0.3 V. Applying a voltage in excess of the maximum rating will permanen
the device.**

**Do not select the HN28F101 programmer setting for the H8/3048F-ONE (single p
supply). If this setting is made by mistake, 12.0 V will be applied to the FWE pin,
permanently damaging the device.**

**When using a PROM programmer to program the on-chip flash memory in the H
ONE (single power supply), use a PROM programmer that supports Renesas Tec
microcomputer device types with 128-kbyte on-chip flash memory.**

Table 18.1 Operating Mode and ROM

	Pin Name					
Mode	**MD_2**	**MD_1**	**MD_0**	**FWE**	**RXD1**	**On-Chip**
Mode 1 (1-Mbyte expanded mode with on-chip ROM disabled)	0	0	1	0	0/1	Disabled (address a
Mode 2 (1-Mbyte expanded mode with on-chip ROM disabled)	0	1	0	0	0/1	
Mode 3 (16-Mbyte expanded mode with on-chip ROM disabled)	0	1	1	0	0/1	
Mode 4 (16-Mbyte expanded mode with on-chip ROM disabled)	1	0	0	0	0/1	
Mode 5 (1-Mbyte expanded mode with on-chip ROM enabled)	1	0	1	0	0/1	Enabled
Mode 6 (16-Mbyte expanded mode with on-chip ROM enabled)	1	1	0	0	0/1	
Mode 7 (single-chip mode)	1	1	1	0	0/1	

The H8/3048F-ONE can be set to PROM mode and programmed with a general-purpos
programmer.

— Program-verify mode

— Erase-verify mode

- Programming/erase methods

  The flash memory is programmed 128 bytes at a time. Block erase (in single-block be performed. To erase the entire flash memory, each block must be erased in turn erasing can be performed as required on 1 kbyte, 28 kbytes, and 32 kbytes blocks.

- Programming/erase times

  The flash memory programming time is 10 ms (typ.) for simultaneous 128-byte pr equivalent approximately to 80 µs (typ.) per byte, and the erase time is 100 ms (ty

- 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 c

  — Boot mode

  — User program mode

  In the boot mode, the transferred program from the host can be recognized.

- Automatic bit rate adjustment

  With data transfer in boot mode, the LSI's bit rate can be automatically adjusted to transfer bit rate of the host.

- Flash memory emulation in RAM

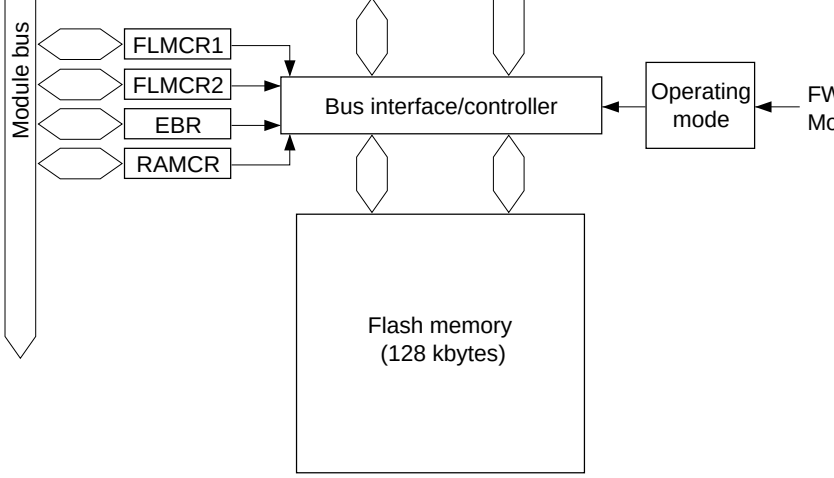
  Flash memory programming can be emulated in real time by overlapping a part of flash memory.

- Protect modes

  There are three protect modes, hardware, software, and error which allow protecte designated for flash memory program/erase/verify operations.

- PROM mode

  Flash memory can be programmed/erased in PROM mode, using a PROM progra well as in on-board programming mode.



Legend:
FLMCR1: Flash memory control register 1
FLMCR2: Flash memory control register 2
EBR: Erase block register
RAMCR: RAM control register

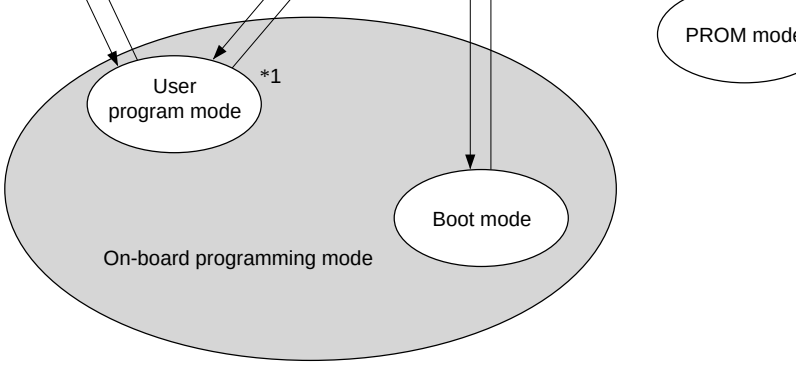
Note: Never apply 12 V to the H8/3048F-ONE (single power supply).
Otherwise, the LSI will be permanently damaged.

Figure 18.1 Block Diagram of Flash Memory

18.2.2 Mode Transitions

When the mode pins and the FWE pin are set in the reset state and a reset-start is execu microcomputer enters an operating mode as shown in figure 18.2. In user mode, flash m be read but not programmed or erased.

The boot, user program and PROM modes are provided as modes to write and erase the memory.



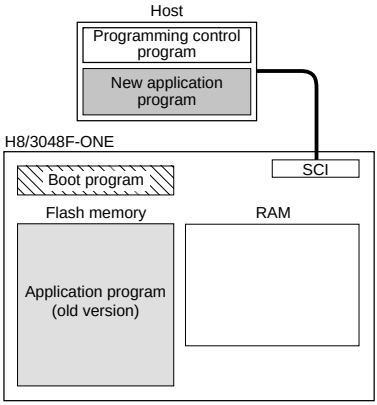
Notes: Only make a transition between user mode and user program mode when the CPU is not the flash memory.

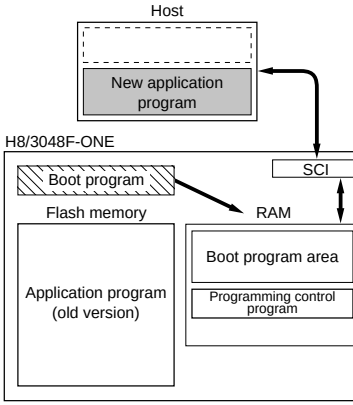
1. RAM emulation possible
2. The H8/3048F-ONE is placed in PROM mode by means of a dedicated PROM writer.
3. Mode settings are shown in the following table.
4. For pins RXD1 and TXD1, use on-board pull-up in boot mode.

Mode	Pins				
	FWE	MD_2	MD_1	MD_0	RXD1
Mode 1	0	0	0	1	0, 1
Mode 2		0	1	0	0, 1
Mode 3		0	1	1	0, 1
Mode 4		1	0	0	0, 1
Mode 5		1	0	1	0, 1
Mode 6		1	1	0	0, 1
Mode 7		1	1	1	0, 1
Boot mode 5	1	0	0	1	1*4
Boot mode 6		0	1	0	1*4
Boot mode 7		0	1	1	1*4
Setting prohibited		1	0	0	1
User program mode 5		1	0	1	0, 1
User program mode 6		1	1	0	0, 1
User program mode 7		1	1	1	0, 1

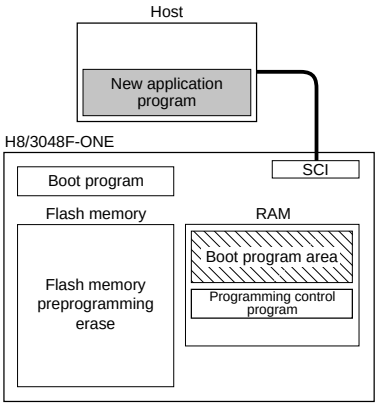
Figure 18.2 Flash Memory State Transitions

to the RAM boot program area.

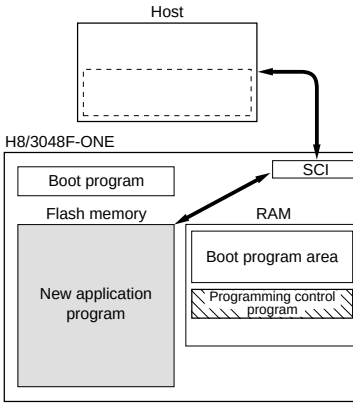




3. Flash memory initialization
   The erase program in the boot program area (in RAM) is executed, and the flash memory is initialized (to H'FF). In boot mode, total flash memory erasure is performed, without regard to blocks.

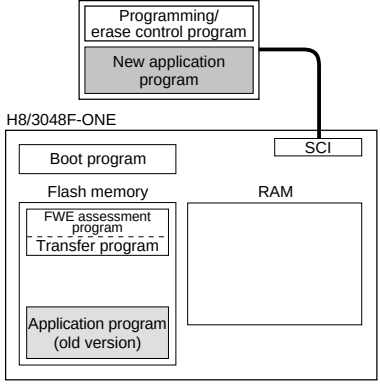


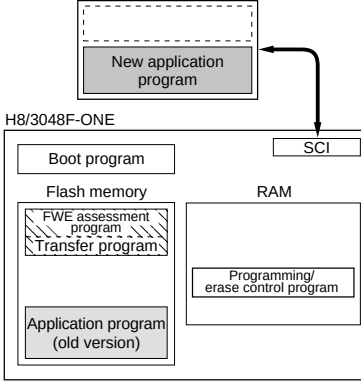
4. Writing new application program
   The programming control program is recog
   if it corresponds to the H8/3048F-ONE.
   The programming control program is then
   transferred from the host to RAM is execut
   and the new application program in the hos
   written into the flash memory.



Program executio

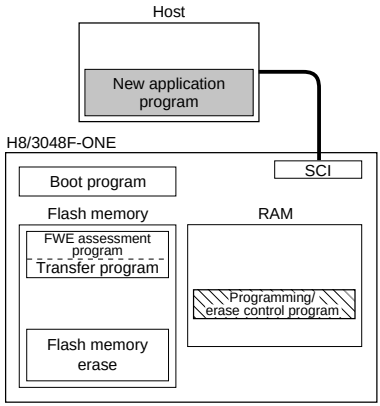
Note: Never apply 12 V to the H8/3048F-ONE (single power supply).
Otherwise, the LSI will be permanently damaged.

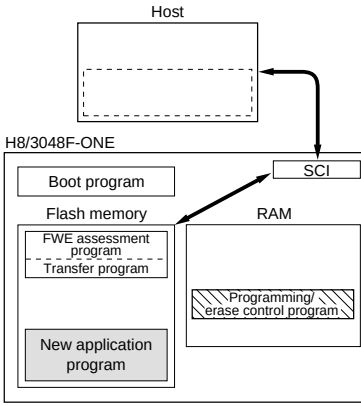




3. Flash memory initialization
   The programming/erase program in RAM is executed, and the flash memory is initialized (to H'FF). Erasing can be performed in block units, but not in byte units.

4. Writing new application program
   Next, the new application program in the ho written into the erased flash memory blocks. not write to unerased blocks.





Program executio

Note: Never apply 12 V to the H8/3048F-ONE (single power supply).
Otherwise, the LSI will be permanently damaged.

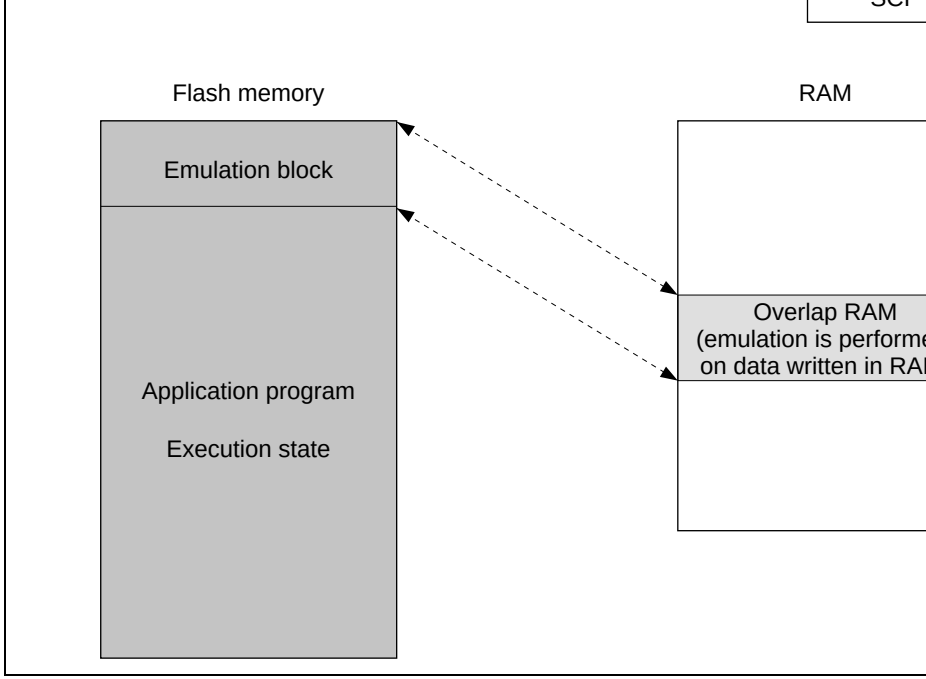


Figure 18.3 Reading Overlap RAM Data in User Mode or User Program

When overlap RAM data is confirmed, clear the RAMS bit to release RAM overlap, a
perform writes to the flash memory.

When the programming control program is transferred to RAM in on-board programm
ensure that the transfer destination and the overlap RAM do not overlap, as this will c
the overlap RAM to be rewritten.

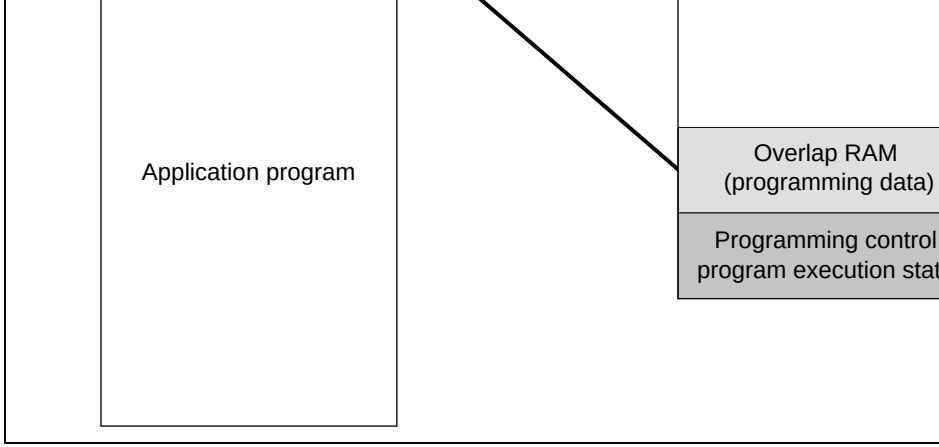


Figure 18.4 Writing Overlap RAM Data in User Program Mode

18.2.5 Differences between Boot Mode and User Program Mode

Item	Boot Mode	User Program Mode
Total erase	Yes	No
Block erase	No	Yes
Programming control program*	Boot program is initiated, and programming control program is transferred from host to on-chip RAM, and executed there.	Program that controls programming progra memory is executed. should be written bef PROM mode and boo

Note: * To be provided by the user, in accordance with the recommended algorithm.

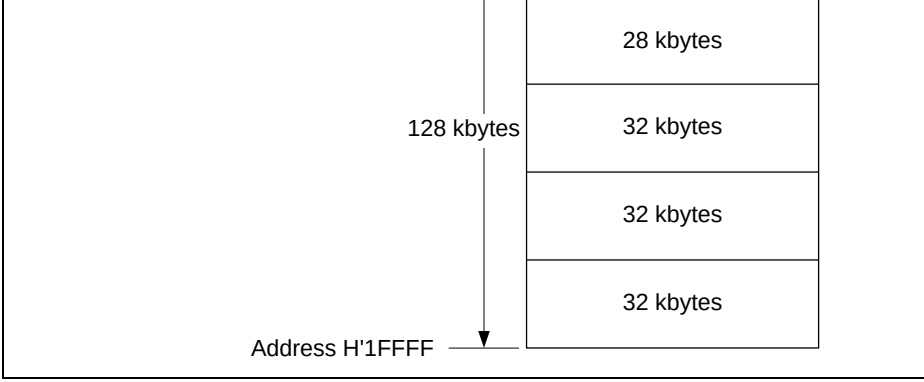


Figure 18.5 Erase Area Block Divisions

18.3 Flash Memory Pin Configuration

The flash memory is controlled by means of the pins shown in table 18.2.

Table 18.2 Pin Configuration

Pin Name	Abbreviation	I/O	Function
Reset	$\overline{\text{RES}}$	Input	Reset
Flash write enable	FWE*1	Input	Flash program/erase protection
Mode 2	MD2*1	Input	Sets LSI operating mode
Mode 1	MD1	Input	Sets LSI operating mode
Mode 0	MD0	Input	Sets LSI operating mode
Transmit data	TxD1*2	Output	Serial transmit data output
Receive data	RxD1*2	Input	Serial receive data input

Notes: 1. Never apply 12 V to the H8/3048F-ONE (single power supply). Otherwise, the LSI will be permanently damaged.
2. In boot mode, use on-board pull-up.

Table 18.3 Register Configuration

Register Name	Abbreviation	R/W	Initial Value	Ad
Flash memory control register 1	FLMCR1*5	R/W*2	H'00*3	H'F
Flash memory control register 2	FLMCR2*5	R/W*2 *6	H'00	H'F
Erase block register	EBR*5	R/W*2	H'00*4	H'F
RAM control register	RAMCR*5	R/W	H'F0	H'F

Notes: 1. Lower 16 bits of the address.
2. If the chip is in a mode in which the on-chip flash memory is disabled, a read H'00 and writes are invalid. Writes are also invalid when the FWE bit in FLM set to 1.
3. When a high level is input to the FWE pin, the initial value is H'80.
4. When a low level is input to the FWE pin, or if a high level is input and the SW FLMCR1 is 0, these registers are initialized to H'00.
5. FLMCR1, FLMCR2, EBR, and RAMCR are 8-bit registers.
Byte access must be used on these registers (do not use word or longword a
6. Bits 6 to 0 are reserved bits but are readable/writable.

18.5 Flash Memory Register Descriptions

18.5.1 Flash Memory Control Register 1 (FLMCR1)

Bit	7	6	5	4	3	2	1
	FWE	SWE	ESU	PSU	EV	PV	E
Initial value	—*	0	0	0	0	0	0
Read/Write	R	R/W	R/W	R/W	R/W	R/W	R/W

Note: * Determined by the state of the FWE pin.

a read will return H'00, and writes are invalid. Set 1 to bits 6 to 0 by each bit in this re

Writes are enabled only in the following cases: Writes to bit SWE of FLMCR1 enable FWE = 1, to bits ESU, PSU, EV, and PV when FWE = 1 and SWE = 1, to bit E when SWE = 1 and ESU = 1, and to bit P 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 n normal on-chip ROM, bits 6 to 0 in this register must be cleared.

Bit 7—Flash Write Enable Bit (FWE): Sets hardware protection against flash memo 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

Bit 6—Software Write Enable Bit (SWE): Enables or disables flash memory progra erasing (applicable addresses: H'00000 to H'1FFFF). Set this bit when setting bits 5 to 0 of EBR.

Bit 6: SWE	Description
0	Writes disabled
1	Writes enabled* [Setting condition] When FWE = 1

Note: * Do not execute a SLEEP instruction while the SWE bit is set to 1.

	When FWE = 1 and SWE = 1

Bit 4—Program Setup Bit (PSU): Prepares for a transition to program mode (applicat addresses: H'00000 to H'1FFFF). Do not set the SWE, ESU, EV, PV, E, or P bit at the s Set this bit to 1 before setting bit P to 1 in FLMCR1.

Bit 4: PSU	Description	
0	Program setup cleared	(In
1	Program setup [Setting condition] When FWE = 1 and SWE = 1	

Bit 3—Erase-Verify Bit (EV): Selects erase-verify mode transition or clearing (applic addresses: H'00000 to H'1FFFF). Do not set the SWE, ESU, PSU, PV, E, or P bit at the

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 Bit (E): Selects erase mode transition or clearing (applicable addresses: H'1FFFF). Do not set the SWE, ESU, 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 flash memory while the E bit is set to 1.

Bit 0—Program Bit (P): Selects program mode transition or clearing (applicable add H'00000 to H'1FFFF). Do not set the SWE, PSU, ESU, EV, PV, or E bit at the same ti

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 flash memory while the P bit is set.

FLMCR2 is an 8-bit register used for flash memory operating mode control. FLMCR2 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: Bits 6 to 0 are reserved bits but are readable/writable.

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 readable/writable.

Each bit in EBR cannot be set until the SWE bit in FLMCR1 is set. The flash memory configuration is shown in table 18.4. To erase all the blocks, erase each block sequent

Bit		7	6	5	4	3	2	1
		EB7	EB6	EB5	EB4	EB3	EB2	EB1
Modes 1 to 4	Initial value	0	0	0	0	0	0	0
	Read/Write	R	R	R	R	R	R	R
Modes 5 to 7	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

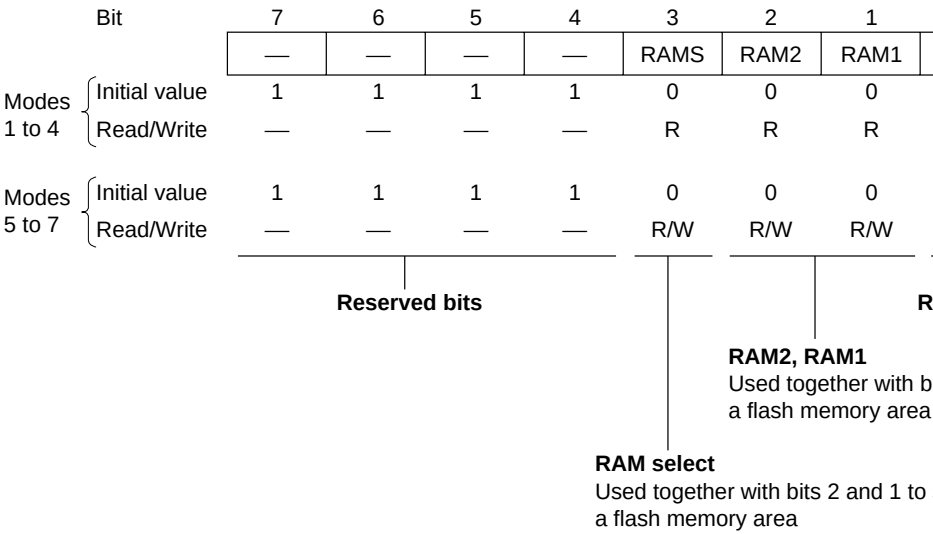
Bits 7 to 0—Block 7 to Block 0 (EB7 to EB0): Setting one of these bits specifies the corresponding block (EB7 to EB0) for erasure.

Bits 7–0: EB7–EB0	**Description**
0	Corresponding block (EB7 to EB0) not selected
1	Corresponding block (EB7 to EB0) selected

Note: When not performing an erase, clear EBR to H'00.

EB5 (32 kbytes)	H'008000–H'00FFFF
EB6 (32 kbytes)	H'010000–H'017FFF
EB7 (32 kbytes)	H'018000–H'01FFFF

18.5.4 RAM Control Register (RAMCR)



RAMCR selects the RAM area to be used when emulating real-time flash memory prog
RAMCR initialized to H'F0 by a reset and in hardware standby mode. It is not initialize
software standby mode. RAMCR settings should be made in user mode or user progra

Note: * When performing flash memory emulation by RAM, the RAME bit in SYS
set to 1.

RAM area settings are shown in table 18.5. To ensure correct operation of the emulatio
the ROM for which RAM emulation is performed should not be accessed immediately

0	Emulation not selected Program/erase-protection of all flash memory blocks is disabled	(
1	Emulation selected Program/erase-protection of all flash memory blocks is enabled	

Bits 2 and 1—RAM2 and RAM1: These bits are used with bit 3 to reassign an area t
table 18.5).

Bit 0—Reserved: This bit is readable/writable.

Table 18.5 RAM Area Setting

	Bit 3	Bit 2	Bit 1	
RAM Area	**RAMS**	**RAM2**	**RAM1**	**RAM Emulation S**
H'FFF000–H'FFF3FF	0	0/1	0/1	No emulation
H'000000–H'0003FF	1	0	0	Mapping RAM
H'000400–H'0007FF	1	0	1	
H'000800–H'000BFF	1	1	0	
H'000C00–H'000FFF	1	1	1	

Figure 18.6 Example of ROM Area/RAM Area Overlap

18.6 Flash Memory On-Board Programming Modes

When pins are set to on-board programming mode and a reset-start is executed, a transi
made to the on-board programming state in which program/erase/verify operations can
performed on the on-chip flash memory. There are two on-board programming modes:
and user program mode. The pin settings for transition to each of these modes are show
18.6. For a diagram of the transitions to the various flash memory modes, see figure 18

Table 18.6 Setting On-Board Programming Modes

Mode		FWE*2	MD2*2	MD1	MD0	RxD1
Boot mode	Mode 5	1*1	0*1	0	1	1*3
	Mode 6		0*1	1	0	1*3
	Mode 7		0*1	1	1	1*3
User program mode	Mode 5		1*1	0	1	0/1
	Mode 6		1*1	1	0	0/1
	Mode 7		1*1	1	1	0/1

Notes: 1. (1) For the high-level application timing, see Notes on Use of Boot Mode.
(2) In boot mode, the inverse of the MD_2 setting should be input.
(3) In boot mode, the mode control register (MDCR) can be used to monitor
of modes 5, 6, and 7, in the same way as in normal mode.
2. Never apply 12 V to the H8/3048F-ONE (single power supply). If do so, the
permanently damaged.
3. For pins RXD1 and TXD1, use on-board pull-up.

transfer is completed, the programming control program is recognized (the ID code is
it corresponds to the H8/3048F-ONE. When the ID code is matched, control branches
address of the programming control program area and the programming control progra
state is entered (flash memory programming is performed).

The transferred programming control program must therefore include coding that follo
programming algorithm given later.

The system configuration in boot mode is shown in figure 18.7, and the boot mode ex
procedure in figure 18.8.

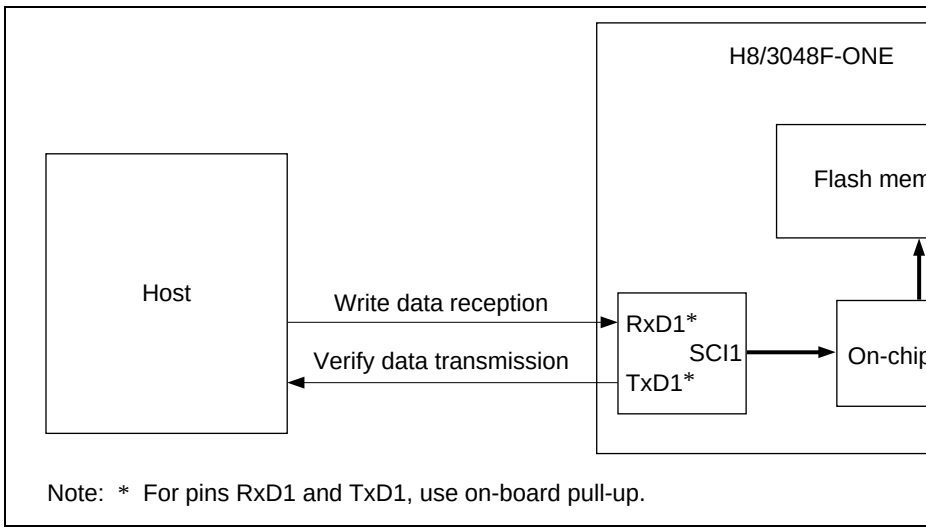
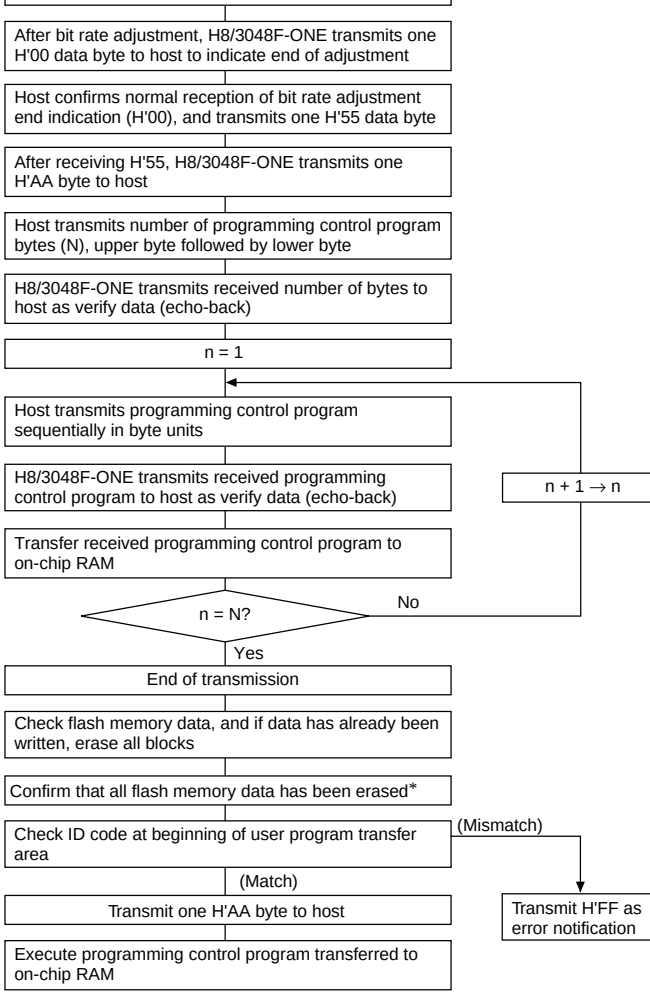


Figure 18.7 System Configuration in Boot Mode



Note: * If a memory cell does not operate normally and cannot be erased, one H'FF byte is transmitted as an erase indication, and the erase operation and subsequent operations are halted.

Figure 18.8 Boot Mode Execution Procedure

When boot mode is initiated, H8/3048F-ONE measures the low period of the asynchro communication data (H'00) transmitted continuously from the host. The SCI transmit/ format should be set as follows: 8-bit data, 1 stop bit, no parity. H8/3048F-ONE calcu rate of the transmission from the host from the measured low period, and transmits on to the host to indicate the end of bit rate adjustment. The host should confirm that this end indication (H'00) has been received normally, and transmit one H'55 byte to H8/3 If reception cannot be performed normally, initiate boot mode again (reset), and repea operations. Depending on the host's transmission bit rate and H8/3048F-ONE's syster frequency, there will be a discrepancy between the bit rates of the host and H8/3048F- the host transfer bit rate at 4,800, 9,600 or 19,200 bps* to operate the SCI properly.

Table 18.7 shows host transfer bit rates and system clock frequencies for which autom adjustment of H8/3048F-ONE bit rate is possible. The boot program should be execut this system clock range.

Table 18.7 System Clock Frequencies for which Automatic Adjustment of H8/3 Bit Rate is Possible

Host Bit Rate	System Clock Frequency for Which Automatic Adjustmer of LSI Bit Rate is Possible (MHz)
4800 bps	4 to 25
9,600 bps	8 to 25
19,200 bps	16 to 25

Note: * Use a host bit rate setting of 4800, 9600, or 19200 bps only. No other setti used.
Although the H8/3048F-ONE may also perform automatic bit rate adjustm rate and system clock combinations other than those shown in table 18.7, a error will arise between the bit rates of the host and the H8/3048F-ONE, ar subsequent transfer will not be performed normally. Therefore, only combi bit rate and system clock within the ranges shown in table 18.7 can be usec mode execution.

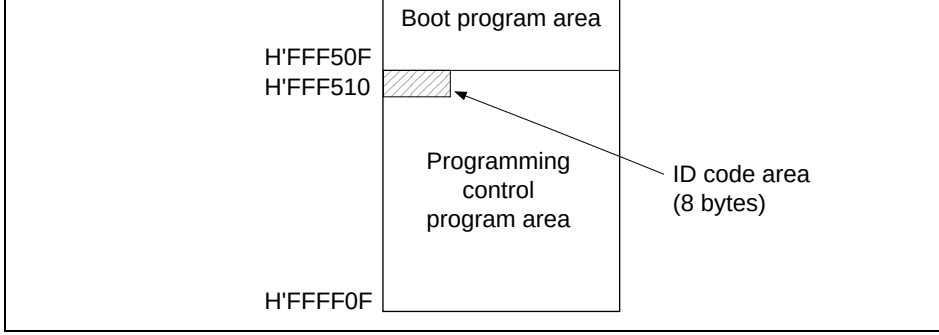
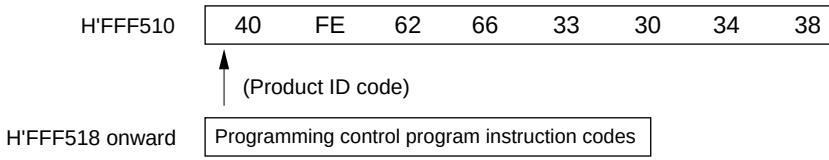


Figure 18.9 RAM Areas in Boot Mode

Note: The boot program area cannot be used until a transition is made to the executio the programming control program transferred to RAM. Note also that the boot remains in this area of the on-chip RAM even after control branches to the prog control program.

In boot mode in the H8/3048F-ONE, the contents of the 8-byte ID code area shown bel checked to determine whether the program is a programming control program compatib H8/3048F-ONE.



If an original programming control program is used in boot mode, the 8-byte ID code s above should be added at the beginning of the program.

accidentally erased and user program mode cannot be executed, for example.

3. Interrupts cannot be used during programming or erasing of flash memory.

4. The RXD_1 and TXD_1 pins should be pulled up on the board.

5. This LSI terminates transmit and receive operations by the on-chip SCI(channel 1) the RE and TE bits in serial control register (SCR)) before branching to the transm output pin. However, the adjusted bit rate is held in the bit rate register (BRR). At TXD_1 is in the high level output state (P9DDR $P9_1$DDR=1, P9DR $P9_1$DR=1).

   Before branching to the programming control program the value of the general reg CPU are also undefined. Therefore, the general registers must be initialized immed control branches to the programming control program. Since the stack pointer (SP) implicitly used during subroutine call, etc., a stack area must be specified for use b programming control program.

   There are no other internal I/O registers in which the initial value is changed.

6. Transition to the boot mode executes a reset-start of this LSI after setting the MD_0 FWE, and RXD1 pins according to the mode setting conditions shown in table 18.

   At this time, this LSI latches the status of the mode pin inside the microcomputer t the boot mode status at the reset clear (startup from Low level to High level) timin

   To clear boot mode, it is necessary to drive the FWE pin low during the reset, and reset release*1. The following points must be noted:

   - Before making a transition from the boot mode to the regular mode, the microc boot mode must be reset by reset input via the $\overline{RES}$ pin*1. At this time, the $\overline{RES}$ hold at low level for at least 20 system clock*2.
   - Do not change the input levels at the mode pins (MD_2 to MD_0) or the FWE pin boot mode. When making a mode transition, first enter the reset state by inputt level to the $\overline{RES}$ pin. When a watchdog timer reset was generated in the boot m microcomputer mode is not reset and the on-chip boot program is restarted reg the state of the mode pin.

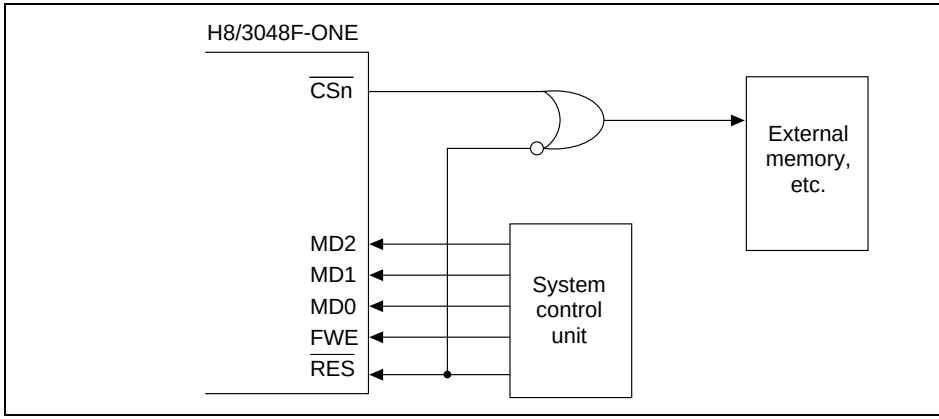


Figure 18.10 Recommended System Block Diagram

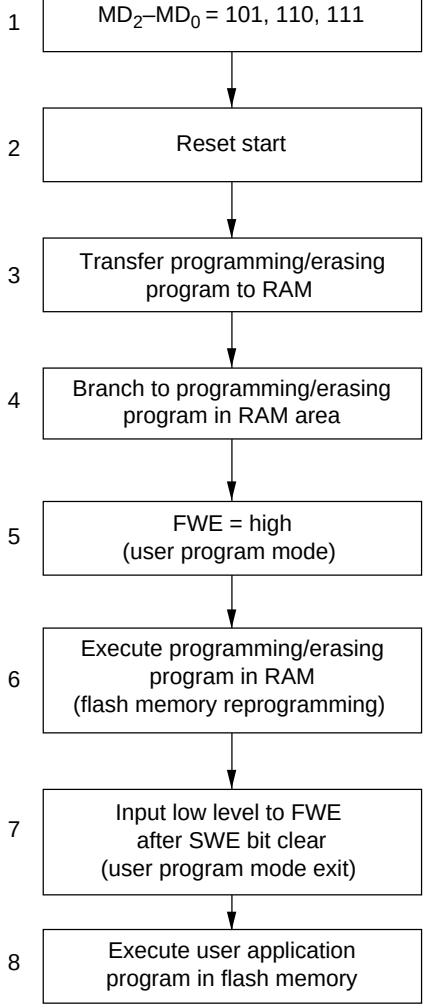
Notes: 1. The mode pin, FWE pin, and RXD1 pin input must satisfy the mode progra setup time (t_{MDS}) relative to the reset clear timing.

2. See section 4.2.2, Reset Sequence and 18.11, Notes on Flash Memory Programming/Erasing. The H8/3048F-ONE requires a minimum of 20 syste
3. For notes on FWE pin High/Low, see section 18.11, Notes on Flash Memor Programming/Erasing.

18.6.2 User Program Mode

When set to the user program mode, user's programming/erasing control program can e program the flash memory. Therefore, on-chip flash memory on-board programming ca performed by providing a means of controlling FWE and supplying the write data on th and providing a programming/erasing program in a part of the program area.

To select this mode, activate H8/3048F-ONE to on-chip flash memory enable modes 5, and apply a high level to the FWE pin. In this state, the peripheral functions, other than memory, are performed the same as in modes 5, 6, and 7.



The user writes a program that execut
3 to 8 in advance as shown below.

1. Sets the mode pin to an on-chip R
enable mode (mode 5, 6, or 7).
2. Starts the CPU via reset.
(The CPU can also be started from
program mode by setting the FWE
High level during reset; that is, dur
period the $\overline{\text{RES}}$ pin is a low level.)
3. Transfers the programming/erasing
to RAM.
4. Branches to the program in RAM a
5. Sets the FWE pin to a high level.*
(Switches to user program mode.)
6. After confirming that the FWE pin i
level, executes the programming/e
program in RAM. This reprograms
application program in flash memo
7. At the end of reprograming, clears
bit, and exits the user program mo
switching the FWE pin from a high
low level.*
8. Branches to, and executes, the us
application program reprogrammed
memory.

Note: * For notes on FWE pin High/L
section 18.11, Notes on Flash
Programming/Erasing.

Figure 18.11 User Program Mode Execution Procedure (Example)

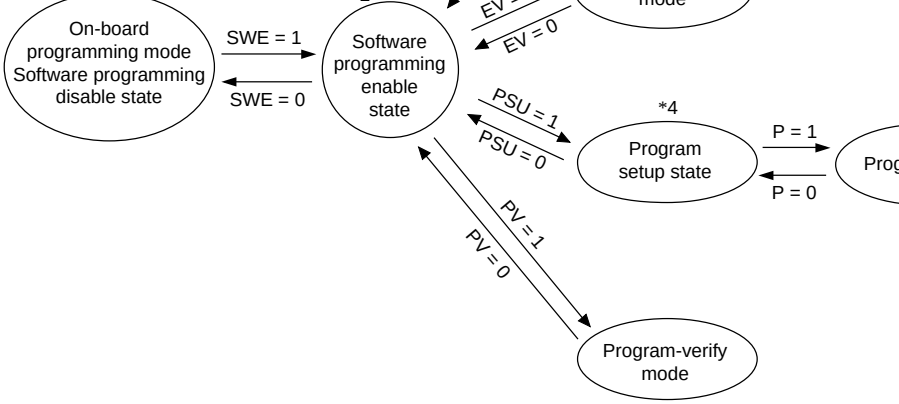
18.7 Programming/Erasing Flash Memory

A software method, using the CPU, is employed to program and erase flash memory in board programming modes. There are four flash memory operating modes: program mo mode, program-verify mode, and erase-verify mode. Transitions to these modes are ma setting the PSU, ESU, P, E, PV, and EV bits in FLMCR1 for addresses H'000000 to H'

The flash memory cannot be read while it is being written or erased. Install the progran flash memory programming and erasing (programming control program) in the on-chip external memory, and execute the program from there.

See section 18.11, Notes on Flash Memory Programming/Erasing, for points to be note programming or erasing the flash memory. In the following operation descriptions, wai after setting or clearing individual bits in FLMCR1 are given as parameters; for details times, see section 21.1.6, Flash Memory Characteristics.

Notes: 1. Operation is not guaranteed if bits SWE, ESU, PSU, EV, PV, E, and P of FI set/reset by a program in flash memory in the corresponding address areas.
2. When programming or erasing, set FWE to 1 (programming/erasing will no executed if FWE = 0).
3. Programming should be performed in the erased state. Do not perform addit programming on previously programmed addresses.



Notes: In order to perform a normal read of flash memory, SWE must be cleared to 0.
Also note that verify-reads 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 18.12 State Transitions Caused by FLMCR1 Bit Settings

the maximum number of programming operations (N) are shown in table 21.11 in secti
Flash Memory Characteristics.

Following the elapse of (t_{sswe}) µs or more after the SWE bit is set to 1 in FLMCR1, 128-
written consecutively to the write addresses. The lower 8 bits of the first address writte
be H'00 and H'80, 128 consecutive byte data transfers are performed. The program addr
program data are latched in the flash memory. A 128-byte data transfer must be perform
writing fewer than 128 bytes; in this case, H'FF data must be written to the extra addres

Next, the watchdog timer (WDT) is set to prevent overprogramming due to program ru
Set a value greater than ($t_{spsu} + t_{sp} + t_{cp} + t_{cpsu}$) µs as the WDT overflow period. Preparatio
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 FLMCR1
elapse of at least (t_{spsu}) µs. The time during which the P bit is set is the flash memory pr
time. Make a program setting so that the time for one programming operation is within
of (t_{sp}) µs.

The wait time after P bit setting must be changed according to the number of reprogram
loops. For details, see Notes on Program/Program-Verify Procedure.

of H'FF data should be made to the addresses to be read. The dummy write should be 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 d before performing this read operation. Next, the originally written data is compared w data, and reprogram data is computed (see figure 18.13) and transferred to RAM. Afte verification of 128 bytes of data has been completed, exit program-verify mode, wait f (t_{cpv}) µs, then determine whether 128-byte programming has finished. If reprogrammin necessary, set program mode again, and repeat the program/program-verify sequence The maximum value for repetition of the program/program-verify sequence is indicate maximum programming count (N). Leave a wait time of at least (t_{cswe}) µs after clearing

Notes on Program/Program-Verify Procedure

1. The program/program-verify procedure for the H8/3048F-ONE is a 128-byte-unit programming algorithm.

   In order to perform 128-byte-unit programming, the lower 8 bits of the write start be H'00 or H'80.

2. When performing continuous writing of 128-byte data to flash memory, byte-unit should be used.

   128-byte data transfer is necessary even when writing fewer than 128 bytes of data must be written 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 FL set. In the H8/3048F-ONE, write pulses should be applied as follows in the progra verify procedure to prevent voltage stress on the device and loss of write data relia

   a. After write pulse application, perform a verify-read in program-verify mode an write pulse again for any bits read as 1 (reprogramming processing). When all 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 H8/3048F-ONE, the nu

additional programming should be performed on the relevant bits. Additio programming should only be performed on bits which first return 0 in a ve in certain reprogramming 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 loop, programming is not necessary for the relevant bits.

c. If programming of other bits is incomplete in the 128 bytes, reprogramming pro be executed. If a bit for which programming has been judged to be completed is in a subsequent verify-read, a write pulse should again be applied to that bit.

5. **The period for which the P bit in FLMCR1 is set (the write pulse width) should changed according to the degree of progress through the program/program-ve procedure.** For detailed wait time specifications, see section 21.1.6, Flash Memory Characteristics.

Table 18.8 Wait Time after P Bit Setting

Item	Symbol	Conditions
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 reprogramm count (n) is 1 to 6.

6. The program/program-verify flowchart for the H8/3048F-ONE is shown in figure 1

To cover the points noted above, bits on which reprogramming processing is to be e and bits on which additional programming is to be executed, must be determined as below.

Since reprogram data and additional-programming data vary according to the progr programming procedure, it is recommended that the following data storage areas (1 each) be provided in RAM.

1	0	1	—
1	1	1	Still in erased state: no action

Legend:

Source data of bits on which programming is executed: (D)

Data of bits on which reprogramming is executed: (X)

Table 18.10 Additional-Programming Data Computation Table

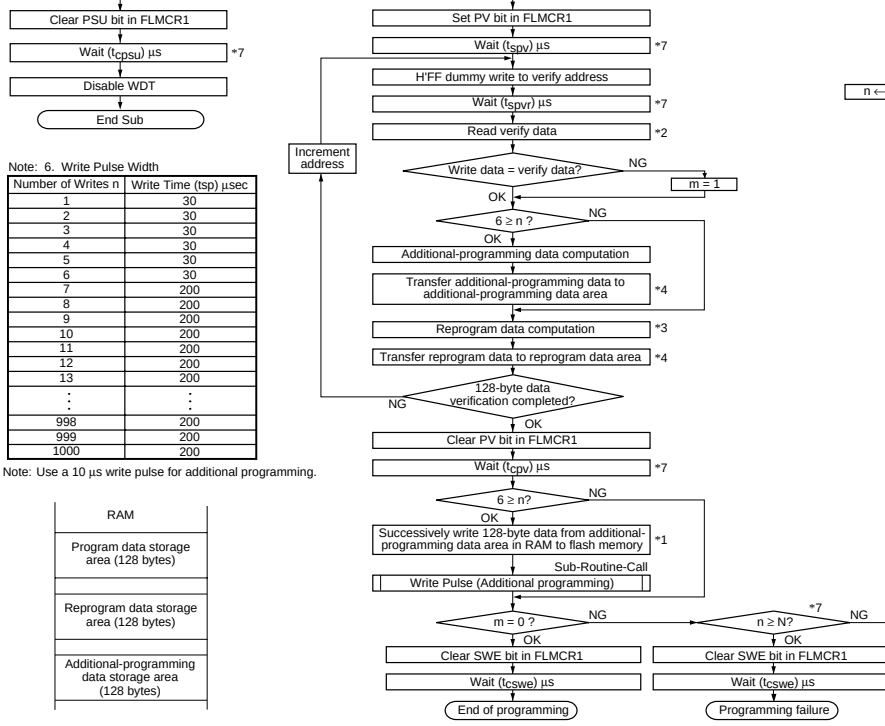
X	Result of Verify-Read after Write Pulse Application (V)	(Y) Result of Operation	Comments
0	0	0	Programming by write pulse app judged to be completed: addition programming processing to be e
0	1	1	Programming by write pulse app incomplete: additional programm processing not to be executed
1	0	1	Programming already completed programming processing not to b
1	1	1	Still in erased state: no action

Legend:

Data of bits on which additional programming is executed: (Y)

Data of bits on which reprogramming is executed in a certain reprogramming loop: (X')

7. It is necessary to execute additional programming processing during the course of H8/3048F-ONE program/program-verify procedure. However, once 128-byte-unit programming is finished, additional programming should not be carried out on the address area. When executing reprogramming, an erase must be executed first. No normal operation of reads, etc., is not guaranteed if additional programming is perf addresses for which a program/program-verify operation has finished.



Note: 6. Write Pulse Width

Number of Writes n	Write Time (tsp) µsec
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 µs 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 verif
Bits for which the reprogram data is 0 are programmed in the next reprogramming loop. Therefore, even bits for which programming has been comple
subjected to 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 data must be provide
The contents of the reprogram data area and additional data area are modified as programming proceeds.
5. A write pulse of 30 µs or 200 µs is applied according to the progress of the programming operation. See Note 6 for details of the pulse widths. When w
additional-programming data is executed, a 10 µs write pulse should be applied. Reprogram data X' means reprogram data when the write pulse is ap
7. The wait times and value of N are shown in section 21.1.6, 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)	Comments
0	0	0	Additional programmin to be executed
0	1	1	Additional programmin not to be executed
1	0	1	Additional programmin not to be executed
1	1	1	Additional programmin not to be executed

Figure 18.13 Program/Program-Verify Flowchart (128-Byte Programmi

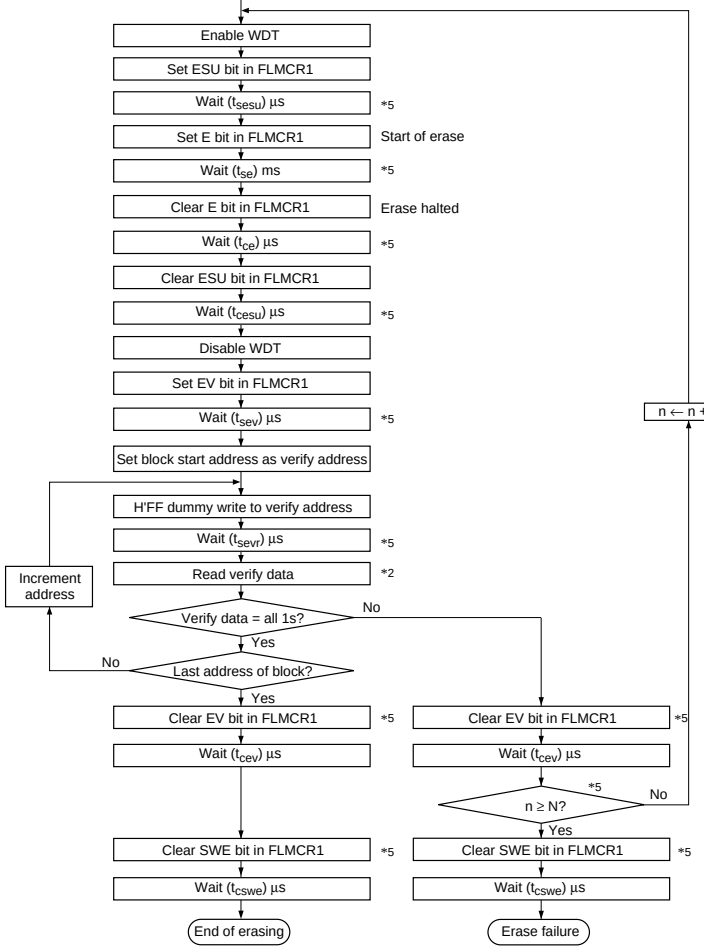
To erase flash memory contents, make a 1-bit setting for the flash memory area to be e
erase block register (EBR1) at least (t_{sswe}) µs after setting the SWE bit to 1 in FLMCR1
watchdog timer (WDT) is set to prevent overerasing due to program runaway, etc. Set
greater than (t_{se}) ms + (t_{sesu} + t_{ce} + t_{cesu}) µs as the WDT overflow period. Preparation for
erase mode (erase setup) is performed next by setting the ESU bit in FLMCR1. The op
mode is then switched to erase mode by setting the E bit in FLMCR1 after the elapse
(t_{sesu}) µs. The time during which the E bit is set is the flash memory erase time. Ensure
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.

18.7.4 Erase-Verify Mode

In erase-verify mode, data is read after memory has been erased to check whether it ha
correctly erased.

After the elapse of the fixed erase time, clear the E bit in FLMCR1, then wait for at le
before clearing the ESU bit to exit erase mode. After exiting erase mode, the watchdo
setting is also cleared. The operating mode is then switched to erase-verify mode by se
bit in FLMCR1. Before reading in erase-verify mode, a dummy write of H'FF data sho
to the addresses to be read. The dummy write should be executed after the elapse of (t
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
operation. If the read data has been erased (all 1), a dummy write is performed to the n
and erase-verify is performed. If the read data is unerased, set erase mode again, and r
erase/erase-verify sequence as before. The maximum value for repetition of the erase/
sequence is indicated by the maximum erase count (N). When verification is complete
erase-verify mode, and wait for at least (t_{cev}) µs. If erasure has been completed on all th
blocks, clear bit SWE1 in FLMCR1, and leave a wait time of at least (t_{cswe}) µs.



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 register (EBR). Two or more bits must not be set simultaneously.
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 21.1.6, Flash Memory Characteristics.

Figure 18.14 Erase/Erase-Verify Flowchart (Single-Block Erasing)

disabled or aborted. Hardware protection is reset by settings in flash memory control r
(FLMCR1), and erase block register (EBR). In the error-protected state, the FLMCR1
and EBR settings are retained; the P and E bits can be set, but a transition is not made
mode or erase mode. (See table 18.11.)

standby protection	in standby mode, FLMCR1, FLMCR2, and EBR 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.		
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 and EBR 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.	No	No*3

Notes: 1. Excluding a RAM area overlapping flash memory.
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.
3. All blocks are unerasable and block-by-block specification is not possible.
4. See section 4.2.2, Reset Sequence, and section 18.11, Notes on Flash Mem Programming/Erasing. The H8/3048F-ONE requires at least 20 system clock reset period during operation.

Item	Description	Functio Program	Erase
Block specification protection	• Erase protection can be set for individual blocks by settings in erase block register (EBR)*2. However, programming protection is disabled. • Setting EBR to H'00 places all blocks in the erase-protected state.	—	No
Emulation protection	• Setting the RAMS bit to 1 in the RAM control register (RAMCR) places all blocks in the program/erase-protected state.	No*1	No*3

Notes: 1. A RAM area overlapping flash memory can be written to.
2. When not erasing, clear all EBR bits to 0.
3. All blocks are unerasable and block-by-block specification is not possible.

settings*3 are retained, but program mode or erase mode is aborted at the point at which occurred. Program mode or erase mode cannot be re-entered by re-setting the P, E bit. I PV, EV bit setting is enabled, and a transition can be made to verify mode*2.

FLER bit setting conditions are as follows:

1. When the flash memory of the relevant address area is read during programming/er (including vector read and instruction fetch)
2. Immediately after exception handling (excluding an illegal reset or trap instruction exception handling at zero division) during programming/erasing
3. When a SLEEP instruction (including software standby) is executed during programming/erasing
4. When the CPU releases the bus to the DMAC, refresh controller, and external bus during programming/erasing

Error protection is released only by a reset ($\overline{\text{RES}}$ pin or WDT reset) and in hardware st mode.

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 and EBR can be written to. However, the registers are initialized transition is made to software standby mode while in the error-protected st

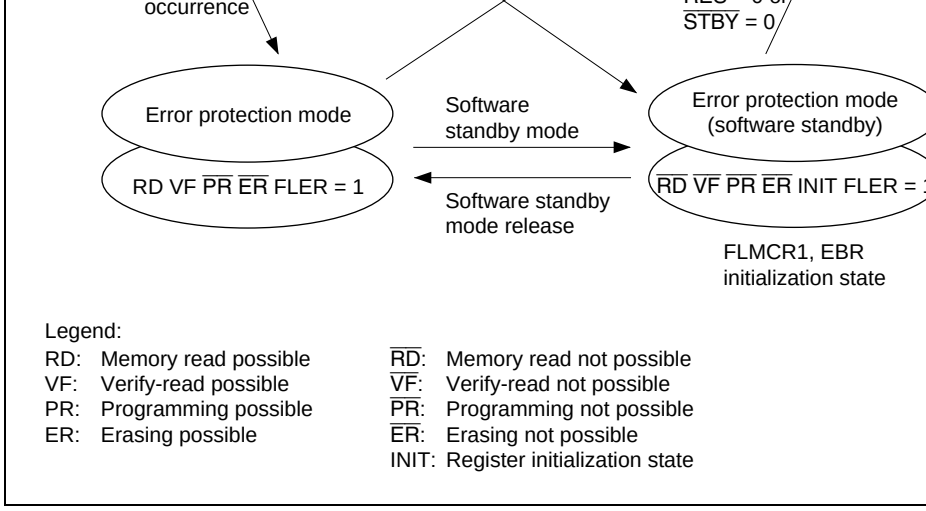


Figure 18.15 Flash Memory State Transitions
(When High Level Is Applied to FWE Pin in Mode 5, 6, and 7 (On-Chip ROM

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

and normal operation can not longer be assured.

2. Vector-read cannot be carried out normally*[2] during interrupt exception handling du programming/erasing and the microcomputer runs away as a result.
3. If an interrupt is generated during boot program execution, the normal boot mode se cannot be executed.

With above reasons, there are conditions that exceptionally disable NMI inputs only in board programming mode. However, this does not assure normal programming/erasing microcomputer operation.

Thus, when the flash memory is programmed/erased, all interrupt requests (exception h and bus release), including NMI, inside and outside the microcomputer, must be disabl interrupt is also disabled in the error-protected state and when the P bit or E bit in FLM retained during flash memory emulation by RAM.

Notes: 1. Indicates the period up to branching to the on-chip RAM boot program area (H'FFEF10). (This branch occurs immediately after programming control pr transfer was completed.)

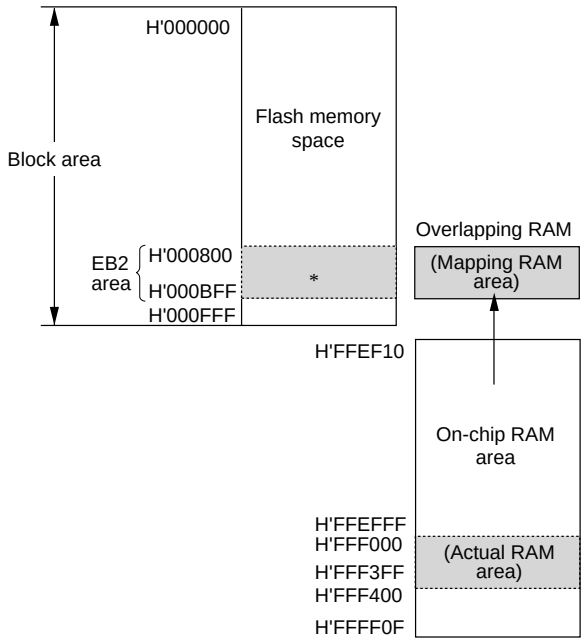
Therefore, after branching to RAM area, NMI input is enabled in states othe program/erase state. Thus, interrupt requests inside and outside the microco must be disabled until initial writing by programming control program (writ vector table and NMI processing program, etc.) is completed.

2. In this case, vector read is not performed normally for the following two rea
   - The correct value cannot be read even by reading the flash memory duri programming/erasing (P bit or E bit in FLMCR1 is set). (Value is undefi
   - If a value has not yet been written to the interrupt vector table, interrupt handling will not be performed correctly.

RAMCR and the RAM area setting method, see section 18.3.4, RAM Control Registe

Example of Emulation of Real-Time Flash Memory Programming

In the following example, RAM area H'FFF000–H'FFF3FF is overlapped onto flash m
EB2 (H'000800–H'000BFF).



Procedure:

1. Part of RAM (H'FFF000–H'FFF3FF) is overlapped o area (EB2) requiring real-ti programming.

   (RAMCR bits 3–1 are set to and the flash memory area overlapped (EB2) is selecte
2. Real-time programming is performed using the overla RAM.
3. The programmed data is ch then RAM overlapping is cl (RAMS bit is cleared.)
4. The data written in RAM ar H'FFF000–H'FFF3FF is wri flash memory space.

Note: * When part of RAM (H'FFF000–H'FFF3FF) is overlapped onto a flash memory small block a memory in the overlapped area cannot be accessed. It can be accessed when the overlap cleared.

Figure 18.16 Example of RAM Overlap Operation

while flash memory emulation in RAM is being used.

3. Block area EB0 includes the vector table. When performing RAM emulation, the ve is needed by the overlap RAM.

4. Flash write enable (FWE) application and releasing

   As in on-board programming mode, care is required when applying and releasing F prevent erroneous programming or erasing. To prevent erroneous programming and due to program runaway during FWE application, in particular, the watchdog timer set when the PSU, P, ESU, or E bit in FLMCR1 is set to 1, even while the emulatio is being used. For details, see section 18.11, Notes on Flash Memory Programming/

5. Prohibited conditions of NMI input

   When the emulation function is used, NMI input is prohibited when the P bit or E b FLMCR1 is set to 1, in the same way as with normal programming and erasing. The bits are cleared by a reset (including a watchdog timer reset), in standby mode, whe level is not being input to the FWE pin, or when the SWE bit in FLMCR1 is 0, whil level is being input to the FWE pin.

18.10 Flash Memory PROM Mode

The H8/3048F-ONE has a PROM mode as well as the on-board programming modes fo programming and erasing flash memory. In PROM mode, the on-chip ROM can be free programmed using a general-purpose PROM writer that supports the Renesas Technolo microcomputer device type with 128-kbyte on-chip flash memory.

Product Code	Package	Socket Adapter Product Code*	Manufac
HD64F3048BF	100-pin QFP (FP-100B)	ME3064ESHF1H	Minato E
HD64F3048BVF		ME3024ESHF1H	
HD64F3048BTE	100-pin TQFP (TFP-100B)	ME3064ESNF1H	
HD64F3048BVTE		ME3024ESNF1H	
HD64F3048BF	100-pin QFP (FP-100B)	HF306BQ100D4001	Data IO J
HD64F3048BVF		HF302BQ100D4001	
HD64F3048BTE	100-pin TQFP (TFP-100B)	HF306BT100D4001	
HD64F3048BVTE		HF302BT100D4001	

Note: * Use of the wrong socket adapter may destroy the chip.

Figure 18.17 shows the memory map in PROM mode.

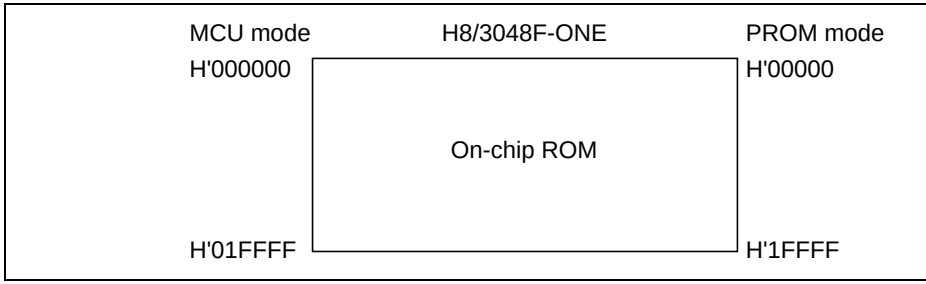


Figure 18.17 Memory Map in PROM Mode

3. The memory is initially in the erased state when the device is shipped by Renesas. F for which the erasure history is unknown, it is recommended that erasing be execute and correct the initialization (erase) level.
4. The H8/3048F-ONE does not support a product identification mode as used with ge 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 H8/3048F-ONE.
6. Select a Renesas Technology 128 kbytes flash memory on-board microcomputer de If HN28F101 is selected, the LSI may be permanently damaged.

H8/3048F-ONE (single power supply) and H8/3048F (dual power supply) models

Use a PROM writer that supports the Renesas Technology 128 kbytes flash memo microcomputer device type.

Do not select the HN28F101 as the PROM writer. Otherwise, 12 V will be applied pin and this will permanently damage H8/3048F-ONE.

2. Notes on powering on/powering off (see figures 18.18 to 18.20)

Input a high level to the FWE pin after verifying Vcc. Before turning off Vcc, set t to a low level.

When powering on and powering off the Vcc power supply, fix the FWE pin low a flash memory to the hardware protection mode.

Be sure that the powering on and powering off timing is satisfied even when the p turned off and back on in the event of a power interruption, etc. If this timing is no microcomputer runaway, etc., may cause overprogramming or overerasing and the cells may not operate normally.

3. Notes on FWE pin High/Low switching (see figures 18.18 to 18.20)

Input FWE in the state microcomputer operation is verified. If the microcomputer satisfy the operation confirmation state, fix the FWE pin low to set the protection m

To prevent erroneous programming/erasing of flash memory, note the following in High/Low switching:

a. Apply an input to the FWE pin after the Vcc voltage has stabilized within the r

   If an input is applied to the FWE pin when the microcomputer Vcc voltage doe the rated voltage, flash memory may be erroneously programmed or erased bec microcomputer is in the unconfirmed state.

b. Apply an input to the FWE pin when the oscillation has stabilized (after the osc stabilization time).

   When turning on the Vcc power, apply an input to the FWE pin after holding th at a low level during the oscillation stabilization time. Do not apply an input to pin when oscillation is stopped or unstable.

d. In the user program mode, FWE = High/Low switching is possible regardless of input.

   FWE input switching is also possible during program execution on flash memor

e. Apply an input to FWE when the program is not running away.

   When applying an input to the FWE pin, the program execution state must be su using a watchdog timer, etc.

f. Release FWE pin input only when the SWE, ESU, PSU, EV, PV, E, and P bits i are cleared.

   Do not erroneously set any of bits SWE, ESU, PSU, EV, PV, E, or P when appl releasing FWE.

4. Do not input a constant high level to the FWE pin.

To prevent erroneous programming/erasing in the event of program runaway, etc., i level to the FWE pin only when programming/erasing flash memory (including flas emulation by RAM). Avoid system configurations that constantly input a high level FWE pin. Handle program runaway, etc. by starting the watchdog timer so that flas is not overprogrammed/overerased even while a high level is input to the FWE pin.

5. Program/erase the flash memory in accordance with the recommended algorit

The recommended algorithms can program/erase the flash memory without applyin stress to the device or sacrificing the reliability of the program data.

When setting the P and E bits in FLMCR1, set the watchdog timer for program runa

Accesses to flash memory by means of an MOV instruction, etc., are prohibited P or bit E is set.

6. Do not set/clear the SWE bit while a program is executing on flash memory.

Before performing flash memory program execution or data read, clear the SWE bit

If the SWE bit is set, the flash data can be reprogrammed, but flash memory cannot accessed for purposes other than verify (verify during programming/erase).

Since programming/erase operations (including emulation by RAM) have priority level is input to the FWE pin, disable all interrupt requests, including NMI. The bu should also be disabled.

8. **Do not perform additional programming. Reprogram flash memory after era**

   With on-board programming, program to 128-byte programming unit blocks one t Erase all the programming unit blocks before reprogramming.

9. **Before programming, check that the chip is correctly mounted in the PROM programmer.**

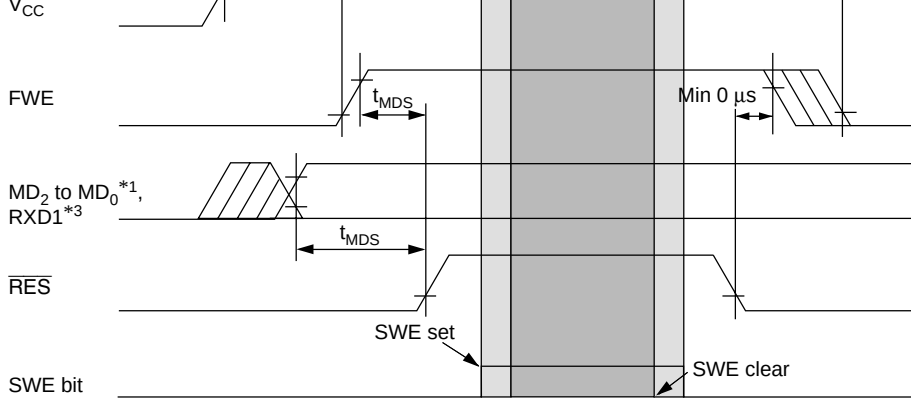
   Overcurrent damage to the device can result if the index marks on the PROM prog socket, socket adapter, and chip are not correctly aligned.

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, FL EBR, and RAMCR).**

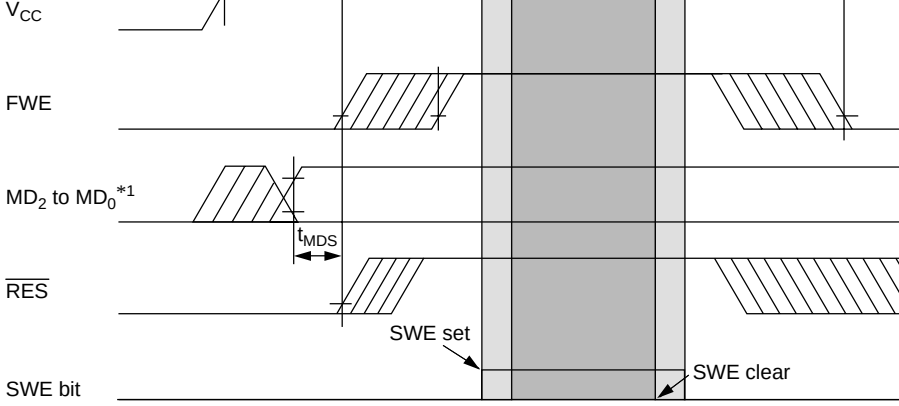


Flash memory access disabled period
(x: Wait time after SWE setting, y: wait time after SWE clearing)*2

Flash memory reprogrammable period
(Flash memory program execution and data read, other than verify, are disabled.)

Notes: 1. Always fix the level by pulling down or pulling up the mode pins (MD_2 to MD_0 until powering off, except for mode switching.
2. See section 21.1.6, Flash Memory Characteristics.
3. For pins RXD1 and TXD1, use on-board pull-up.

Figure 18.18 Powering On/Off Timing (Boot Mode)

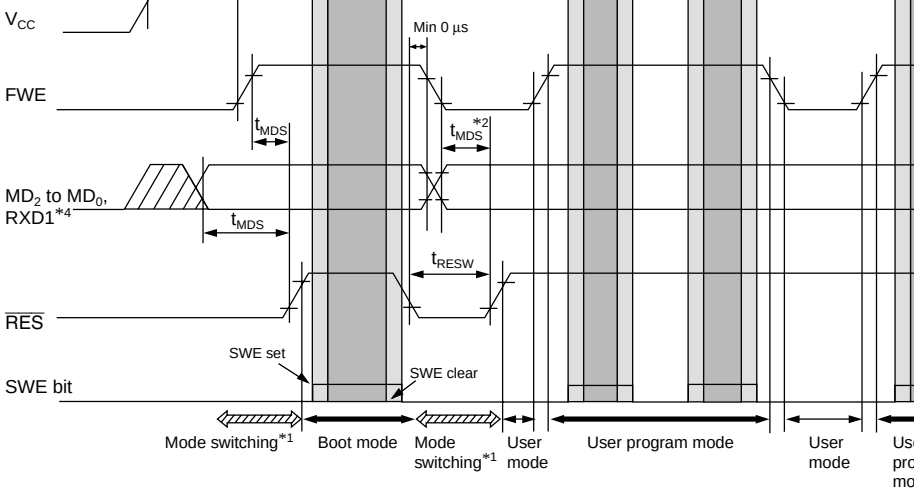


Flash memory access disabled period
(x: Wait time after SWE setting, y: wait time after SWE clearing)*2

Flash memory reprogrammable period
(Flash memory program execution and data read, other than verify, are disabled.)

Notes: 1. Always fix the level by pulling down or pulling up the mode pins (MD_2 to MD_0) up to powering off, except for mode switching.
2. See section 21.1.6, Flash Memory Characteristics.

Figure 18.19 Powering On/Off Timing (User Program Mode)



Flash memory access disabled time
(x: Wait time after SWE setting, y: wait time after SWE clearing)*3

Flash memory reprogammable period
(Flash memory program execution and data read, other than verify, are disabled.)

Notes: 1. In transition to the boot mode and transition from the boot mode to another mode, mode switching v
input is necessary.
During this switching period (period during which a low level is input to the $\overline{RES}$ pin),the state of the
dual port and bus control output signals ($\overline{CS}$n, $\overline{AS}$, $\overline{RD}$, $\overline{WR}$) changes.
Therefore, do not use these pins as output signals during this switching period.

2. When making a transition from the boot mode to another mode, the mode programming setup time
to the $\overline{RES}$ clear timing is necessary.

3. See section 21.1.6, Flash Memory Characteristics.

4. For pin RXD1, use on-board pull-up.

Figure 18.20 Mode Transition Timing
(Example: Boot mode → User mode ↔ User program mode)

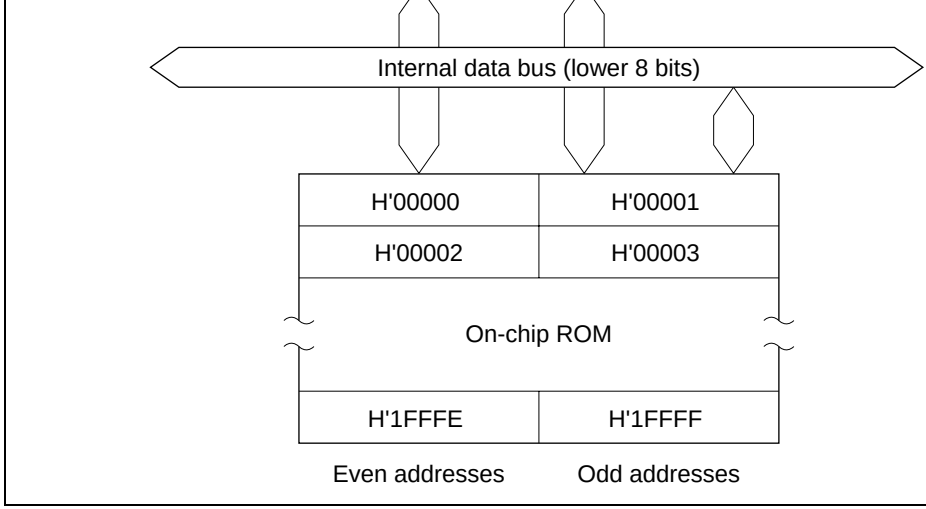


Figure 18.21 ROM Block Diagram (H8/3048B Mask ROM Version)

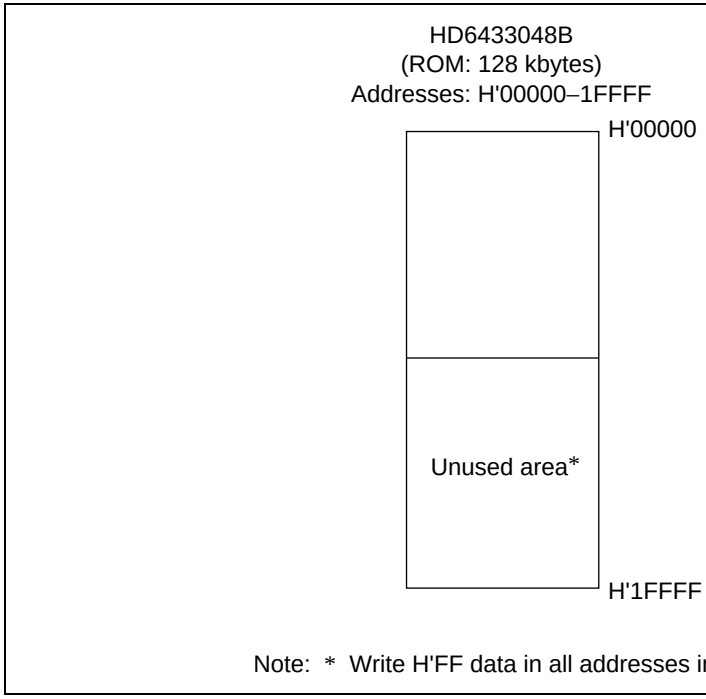


Figure 18.22 Mask ROM Addresses and Data

3. The flash memory control registers (FLMCR, EBR, RAMCR, FLMSR, FLMCR1, EBR1, and EBR2)used by the versions with on-chip flash memory are not provided mask ROM versions. Reading the corresponding addresses in a mask ROM version always return 1s, and writes to these addresses are disabled. This must be borne in m switching from a flash memory version to a mask ROM version.

		Status	
Register	**Bit**	**F-ZTAT (Single Power Supply) Version**	**Mask-ROM Version**
FLMCR1	FWE	0: Application software running 1: Programming	0: Is not read out 1: Application software

Note: This difference applies to all the F-ZTAT (single power supply) versions and all ROM versions that have different ROM size.

pin*1 and furnished as a master clock to prescalers that supply clock signals to the on-
supporting modules. Frequency division ratios of 1/1, 1/2, 1/4, and 1/8 can be selected
frequency divider by settings in a division control register (DIVCR). Power consumpt
chip is reduced in almost direct proportion to the frequency division ratio*2.

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)

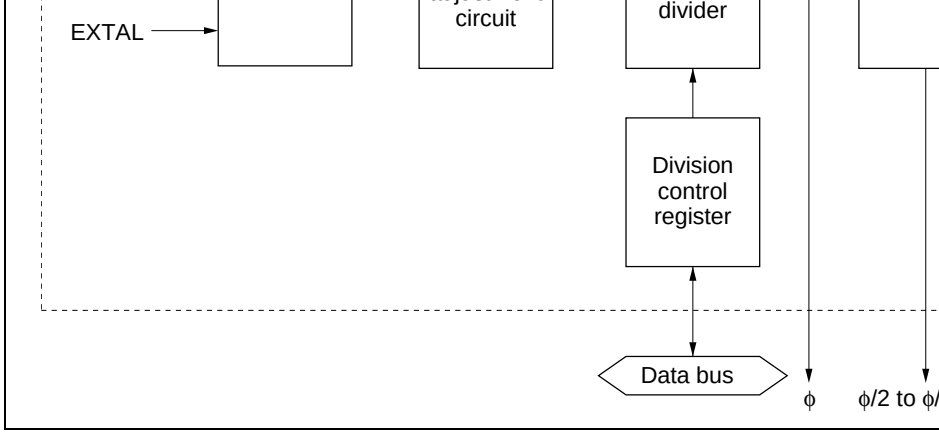


Figure 19.1 Block Diagram of Clock Pulse Generator

19.2 Oscillator Circuit

Clock pulses can be supplied by connecting a crystal resonator, or by input of an extern
signal.

19.2.1 Connecting a Crystal Resonator

Circuit Configuration

A crystal resonator can be connected as in the example in figure 19.2. The damping res
should be selected according to table 19.1, and external capacitance C_{L1} or C_{L2} to table 1
AT-cut parallel-resonance crystal should be used.

Figure 19.2 Connection of Crystal Resonator (Example)

If a crystal resonator with a frequency higher than 20 MHz in the case of the 5 V versi
MHz in the case of the 3 V version, is connected, the external load capacitance values
should not exceed 10 [pF]. Also, in order to improve the accuracy of the oscillation fr
thorough study of oscillation matching evaluation, etc., should be carried out when de
circuit constants.

Table 19.1 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 ≤
Rd (Ω)	H8/3048B Group	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 b
at less than 2 MHz, the on-chip frequency divider should be used. (A crystal res
less than 2 MHz cannot be used.)

Table 19.2 External Capacitance Values

External Capacitance Value	5 V Version		3 V Versi	
Frequency f (MHz)	20 < f ≤ 25	2 ≤ f ≤ 20	2 ≤ f ≤ 13	1
$C_{L1} = C_{L2}$ (pF)	10	10 to 22	10 to 22	

Figure 19.3 Crystal Resonator Equivalent Circuit

Table 19.3 Crystal Resonator Parameters

Frequency (MHz)	2	4	8	10	12	16	18	20
Rs max (Ω)	500	120	80	70	60	50	40	40
C_0 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 should be noted:

Other signal lines should be routed away from the oscillator circuit to prevent induction interfering with correct oscillation. See figure 19.4.

When the board is designed, the crystal resonator and its load capacitors should be plac as possible to the XTAL and EXTAL pins.

C_{L1}

Figure 19.4 Example of Incorrect Board Design

19.2.2 External Clock Input

Circuit Configuration

An external clock signal can be input as shown in the examples in figure 19.5. The ex
is input from the EXTAL pin. If the XTAL pin is left open, the stray capacitance shou
exceed 10 pF. If the stray capacitance at the XTAL pin exceeds 10 pF in configuratio
configuration b instead and hold the clock high in standby mode.

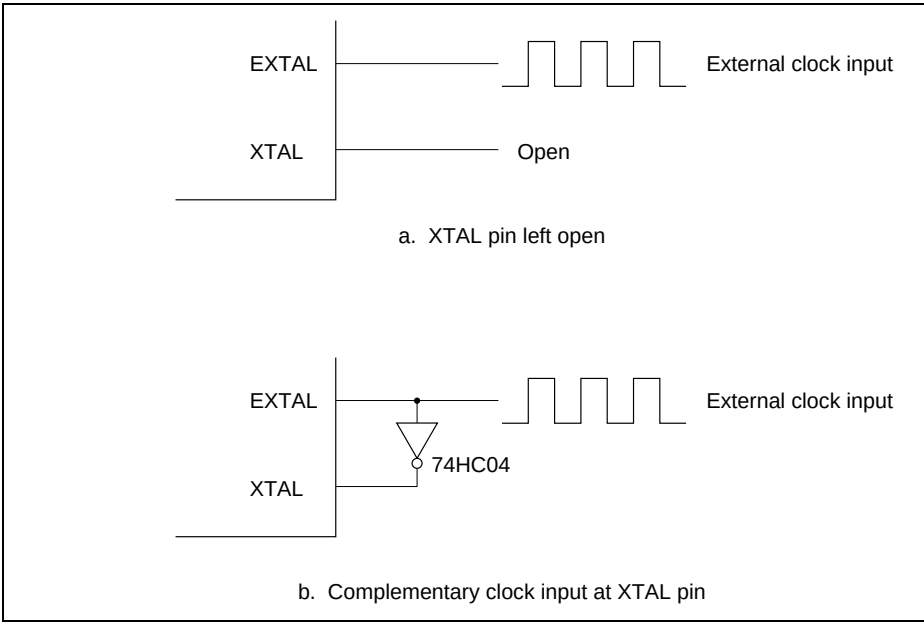


Figure 19.5 External Clock Input (Examples)

Table 19.4(1) Clock Timing for H8/3048B Group (8 MHz ≤ f ≤ 25 MHz)

Item	Symbol	V_{cc} = 3.0 V to 3.6 V		V_{cc} = 5.0 V ±10%		Unit	T C
		Min	Max	Min	Max		
External clock input low pulse width	t_{EXL}	t_{cyc}/2–5	—	t_{cyc}/2–5	—	ns	Fi
External clock input high pulse width	t_{EXH}	t_{cyc}/2–5	—	t_{cyc}/2–5	—	ns	
External clock rise time	t_{EXr}	—	5	—	5	ns	
External clock fall time	t_{EXf}	—	5	—	5	ns	
Clock low pulse width	t_{CL}	0.4	0.6	0.4	0.6	t_{cyc}	F
Clock high pulse width	t_{CH}	0.4	0.6	0.4	0.6	t_{cyc}	
External clock output settling delay time	t_{DEXT}*	500	—	500	—	µs	Fi

Note: * t_{DEXT} includes a $\overline{RES}$ pulse width (t_{RESW}). t_{RESW} = 20 t_{cyc}

input high pulse width	t_{EXH}						
External clock rise time	t_{EXr}	—	5	—	5	ns	
External clock fall time	t_{EXf}	—	5	—	5	ns	
Clock low pulse width	t_{CL}	0.4	0.6	0.4	0.6	t_{cyc}	$\phi \geq 5$ M
		80	—	80	—	ns	$\phi < 5$ M
Clock high pulse width	t_{CH}	0.4	0.6	0.4	0.6	t_{cyc}	$\phi \geq 5$ M
		80	—	80	—	ns	$\phi < 5$ M
External clock output settling delay time	t_{DEXT}*	500	—	500	—	µs	Figure

Note: * t_{DEXT} includes a $\overline{RES}$ pulse width (t_{RESW}). $t_{RESW} = 20\ t_{cyc}$

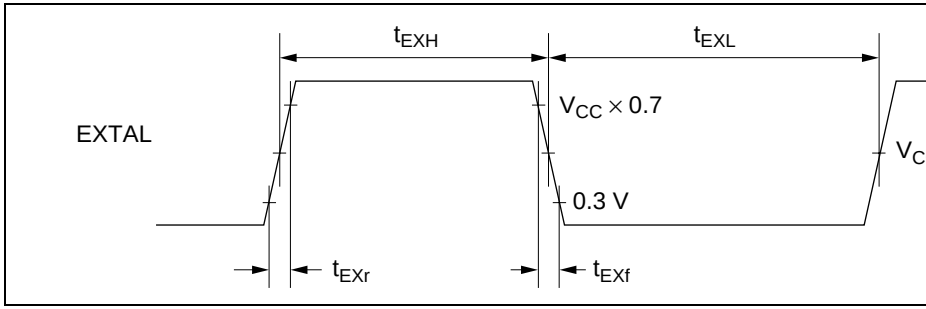


Figure 19.6 External Clock Input Timing

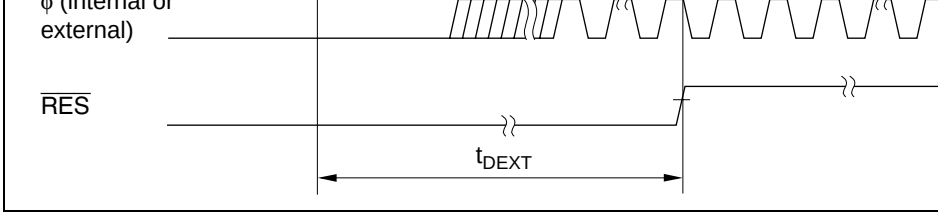


Figure 19.7 External Clock Output Settling Delay Timing

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 the signal that becomes the syst

19.4 Prescalers

The prescalers divide the system clock (ϕ) to generate internal clocks (ϕ/2 to ϕ/4096).

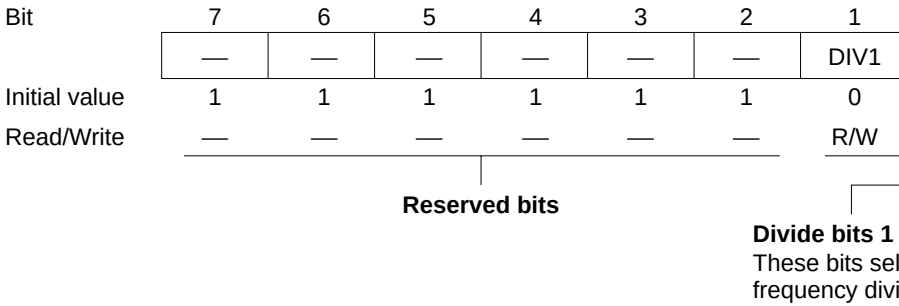
19.5 Frequency Divider

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.

Note: The lower 16 bits of the address are shown.

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: Read-only bits, always read as 1.

Bits 1 and 0—Divide (DIV1 and DIV0): These bits select the frequency division rati
follows.

Bit 1: DIV1	Bit 0: DIV0	Frequency Division Ratio	
0	0	1/1	(
	1	1/2	
1	0	1/4	
	1	1/8	

Table 19.6 shows the comparison with the clock frequency range for each version.

Table 19.6 Comparison with the Clock Frequency Ranges in the H8/3048 Group H8/3048B Group

ROM type		F-ZTAT		ZTAT	Mask ROM			
Product type		H8/3048 F-ONE	H8/3048F	H8/3048	H8/3048 Mask ROM Version	H8/3047 Mask ROM Version	H8/3045 Mask ROM Version	H8/304 Mask ROM Versio
Guaranteed clock frequency range	4.5–5.5 V	2–25 MHz	1–16 MHz	1–18 MHz	1–18 MHz			
	3.15–5.5 V	—		1–13 MHz	1–13 MHz			
	2.7–5.5 V	—	1–8 MHz	1–8 MHz	1–8 MHz			
	3.0–3.6 V	2–25 MHz	—	—	—			
Crystal oscillation range		2–25 MHz	2–16 MHz	2–18 MHz	2–18 MHz			

- All on-chip module operations are based on ϕ. Note that the timing of timer operati communication, and other time-dependent processing differs before and after any c 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 ITU, SCI0, SCI1, DMAC, refresh
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	Refresh Controller	ITU	SCI0	SCI1	A/D	Other Modules	RAM	φ Clock Output
Sleep mode	SLEEP instruction executed while SSBY = 0 in SYSCR	Active	Halted	Held	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	Held	High output
Hardware standby mode	Low input at $\overline{\text{STBY}}$ pin	Halted	Halted	Undeter-mined	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	Active	—	High impedanc

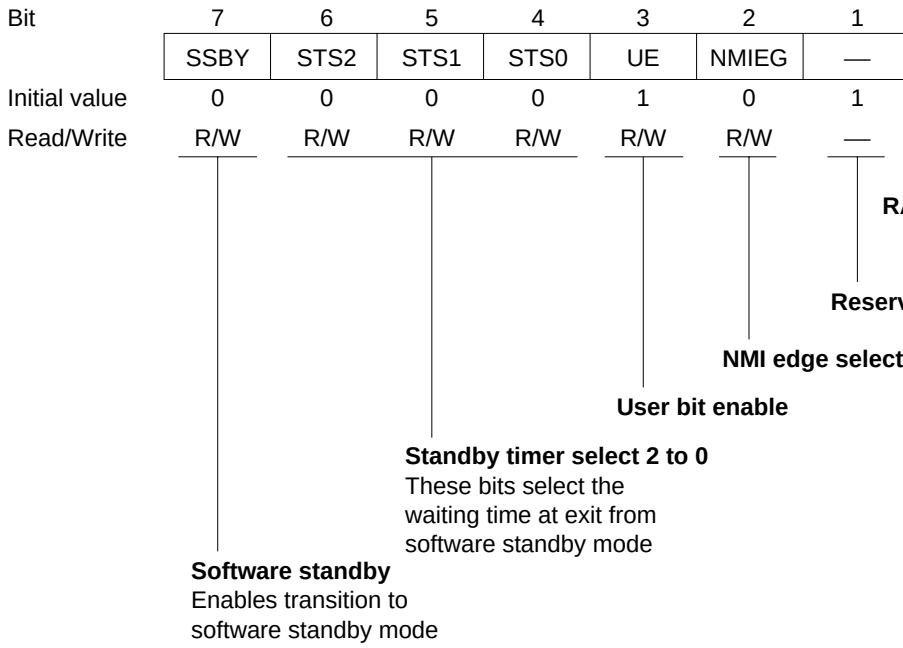
Legend:
SYSCR: System control register
SSBY: Software standby bit
MSTCR: Module standby control register

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 (MSTCR).
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 a MSTCR bit is set to 1, the registers of the corresponding on-chip supporting module are initialized. To restart the module, first clea up the module registers again.

H'FFF2	System control register	SYSCR	R/W	H'0
H'FF5E	Module standby control register	MSTCR	R/W	H'4

Note: * Lower 16 bits of the address.

20.2.1 System Control Register (SYSCR)

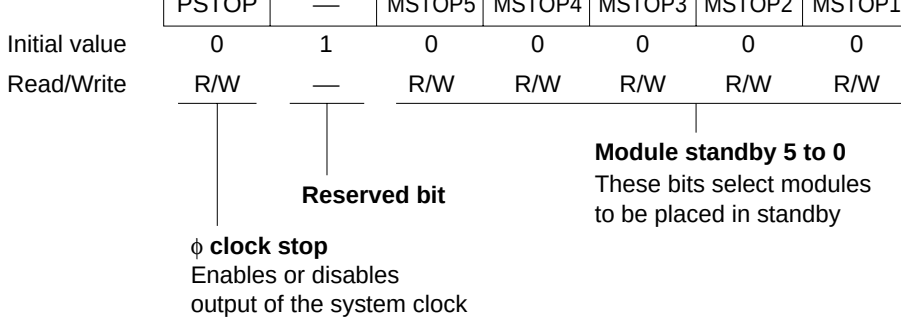


SYSCR is an 8-bit readable/writable register. Bit 7 (SSBY) and bits 6 to 4 (STS2 to S the power-down state. For information on the other SYSCR bits, see section 3.3, Syste Register (SYSCR).

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. See table 20.3.

If an external clock is used, select the setting so that the waiting time is 100 µs or more to the clock frequency.

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	



MSTCR is initialized to H'40 by a reset and in hardware standby mode. It is not initial 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	(
1	System clock output is disabled	

Bit 6—Reserved: Read-only bit, always read as 1.

Bit 5—Module Standby 5 (MSTOP5): Selects whether to place the ITU in standby.

Bit 5: MSTOP5	Description	
0	ITU operates normally	(
1	ITU is in standby state	

Bit 3: MSTOP3	Description	
0	SCI1 operates normally	(In
1	SCI1 is in standby state	

Bit 2—Module Standby 2 (MSTOP2): Selects whether to place the DMAC in standby

Bit 2: MSTOP2	Description	
0	DMAC operates normally	(In
1	DMAC is in standby state	

Bit 1—Module Standby 1 (MSTOP1): Selects whether to place the refresh controller

Bit 1: MSTOP1	Description	
0	Refresh controller operates normally	(In
1	Refresh controller is in standby state	

Bit 0—Module Standby 0 (MSTOP0): Selects whether to place the A/D converter in

Bit 0: MSTOP0	Description	
0	A/D converter operates normally	(In
1	A/D converter is in standby state	

Modules which have been placed in standby by the module standby function, however 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 i 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 no an interrupt other than NMI if the interrupt is masked by the I and UI bits in CCR and

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 mode.

20.4 Software Standby Mode

20.4.1 Transition to Software Standby Mode

To enter software standby mode, execute the SLEEP instruction while the SSBY bit is SYSCR.

In software standby mode, current dissipation is reduced to an extremely low level be CPU, clock, and on-chip supporting modules all halt. The DMAC and on-chip suppor are reset. As long as the specified voltage is supplied, however, CPU register contents RAM data are retained. The settings of the I/O ports and refresh controller* are also h

Note: * RTCNT and bits 7 and 6 of RTMCSR are initialized. Other bits and registe previous states.

Software standby mode can be exited by input of an external interrupt at the NMI, 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 recei
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 e
interrupt exception handling begins. Software standby mode is not exited if the interrup
bits of interrupts IRQ_0, IRQ_1, and IRQ_2 are cleared to 0, or if these interrupts are maske
CPU.

Exit by $\overline{RES}$ Input: When the $\overline{RES}$ input goes low, the clock oscillator starts and clock
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 handl

Exit by $\overline{STBY}$ Input: Low input at the $\overline{STBY}$ pin causes a transition to hardware stand

20.4.3 Selection of Waiting Time for Exit from Software Standby Mode

Bits STS2 to STS0 in SYSCR and bits DIV1 and DIV0 in DIVCR should be set as foll

Crystal Resonator: Set STS2 to STS0, DIV1, and DIV0 so that the waiting time (for t
stabilize) is at least 7 ms. Table 20.3 indicates the waiting times that are selected by ST
STS0, DIV1, and DIV0 settings at various system clock frequencies. Refer to the clock
and the waiting time in which it takes for the clock to settle, as shown in table 20.3.

External Clock: Set bits STS2 to STS0, Bits DIV0, and DIV1 so that the waiting time
or more.

		1	1	0	1024 states	0.04	0.05	0.057	0.064	0.085	0.10	0.13	0.17	0.20
		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
		0	0	1	16384 states	1.3	1.6	1.8	2.0	2.7	3.3	4.1	5.5	8.2*
		0	1	0	32768 states	2.6	3.3	3.6	4.1	5.5	6.6	8.2*	10.9*	16.4
		0	1	1	65536 states	5.2	6.6	7.3*	8.2*	10.9*	13.1*	16.4	21.8	32.8
		1	0	0	131072 states	10.5*	13.1*	14.6	16.4	21.8	26.2	32.8	43.7	65.5
		1	0	1	262144 states	21.0	26.2	29.1	32.8	43.7	52.4	65.5	87.4	131.1
		1	1	0	1024 states	0.08	0.10	0.11	0.13	0.17	0.20	0.26	0.34	0.51
		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*
		0	0	1	16384 states	2.6	3.3	3.6	4.1	5.5	6.6	8.2*	10.9*	16.4
		0	1	0	32768 states	5.2	6.6	7.3*	8.2*	10.9*	13.1*	16.4	21.8	32.8
		0	1	1	65536 states	10.5*	13.1*	14.6	16.4	21.8	26.2	32.8	43.7	65.5
		1	0	0	131072 states	21.0	26.2	29.1	32.8	43.7	52.4	65.5	87.4	131.1
		1	0	1	262144 states	41.9	52.4	58.3	65.5	87.4	104.9	131.1	174.8	262.1
		1	1	0	1024 states	0.16	0.20	0.23	0.26	0.34	0.41	0.51	0.68	1.02
		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*
		0	0	1	16384 states	5.2	6.6	7.3*	8.2*	10.9*	13.1*	16.4	21.8	32.8
		0	1	0	32768 states	10.5	13.1*	14.6	16.4	21.8	26.2	32.8	43.7	65.5
		0	1	1	65536 states	21.0*	26.2	29.1	32.8	43.7	52.4	65.5	87.4	131.1
		1	0	0	131072 states	41.9	52.4	58.3	65.5	87.4	104.9	131.1	174.8	262.1
		1	0	1	262144 states	83.9	104.9	116.5	131.1	174.8	209.7	262.1	349.5	524.3
		1	1	0	1024 states	0.33	0.41	0.46	0.51	0.68	0.82	1.0	1.4	2.0
		1	1	1						Illegal setting				

Note: * Recommended setting

Software standby mode is exited at the next rising edge of the NMI signal.

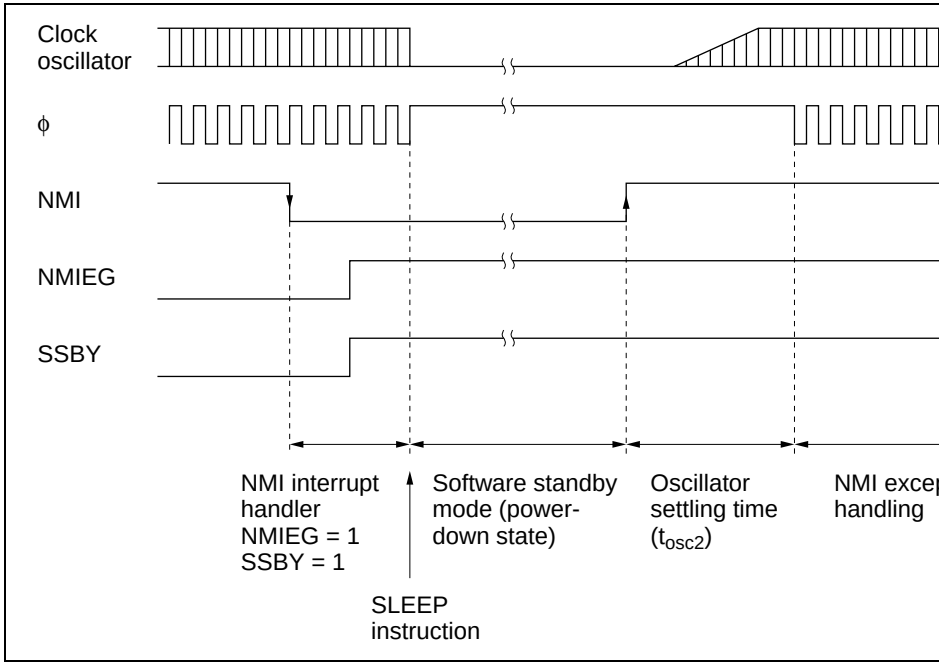


Figure 20.1 NMI Timing for Software Standby Mode (Example)

20.4.5 Note

The I/O ports retain their existing states in software standby mode. If a port is in the hig state, its output current is not reduced.

retained. I/O ports are placed in the high-impedance state.

Clear the RAME bit to 0 in SYSCR before $\overline{\text{STBY}}$ goes low to retain on-chip RAM da

The inputs at the mode pins (MD_2 to MD_0) should not be changed during hardware sta

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 $\overline{\text{STBY}}$ goes high, the clock oscillator starts running. $\overline{\text{RES}}$ should be held low long eno clock oscillator to settle. When $\overline{\text{RES}}$ goes high, reset exception handling begins, follo 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, firs $\overline{\text{STBY}}$ high, wait for the clock to settle, then bring $\overline{\text{RES}}$ from low to high.

Figure 20.2 Hardware Standby Mode Timing

20.6 Module Standby Function

20.6.1 Module Standby Timing

The module standby function can halt several of the on-chip supporting modules (the I
SCI1, DMAC, refresh controller, and A/D converter) independently of the power-down
standby function is controlled by bits MSTOP5 to MSTOP0 in MSTCR. When one of t
set to 1, the corresponding on-chip supporting module is placed in standby and halts at
beginning of the 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 regis
disabled. Read access always results in H'FF data. Write access is ignored.

Internal Peripheral Module Interrupt: When MSTCR is set to 1, prevent module in advance. When an on-chip supporting module is placed in standby by the module stan function, its registers, including the interrupt flag, are initialized.

Pin States: Pins used by an on-chip supporting module lose their module functions wh module is placed in module standby. What happens after that depends on the particula details, see section 9, 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 generic I/O pin. If its data direction bit is set to 1, the pin beco output pin, and its output may collide with external serial data. Data collisions should by clearing the data direction 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.

MSTCR Access from DMAC Disabled: To prevent malfunctions, MSTCR can only from the CPU. It can be read by the DMAC, but it cannot be written by the DMAC.

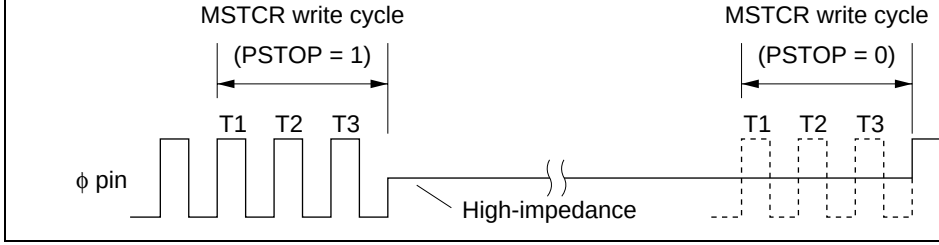


Figure 20.3 Starting and Stopping of System Clock Output

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

Item		Symbol	Unit	H8/3048 F-ZTAT (Dual Power Supply)	H8/3048 H8/3047 H8/3045 H8/3044	H8/3048 ZTAT	H8/30 F-ON (Sing Powe Suppl
Operating range	V_{CC} = 4.5 to 5.5 V		MHz	1 to 16	1 to 18	1 to 18	2 to 25 (5 V operati model)
	V_{CC} = 3.15 to 5.5 V			—	1 to 13	1 to 13	—
	V_{CC} = 2.7 to 5.5 V			1 to 8	1 to 8	1 to 8	—
	V_{CC} = 3.0 to 3.6 V			—	—	—	2 to 25 (3 V operati model)
Operating temperature range	Regular specifications	T_{opr}	°C	–20 to +75	–20 to +75	–20 to +75	–20 t +75*
	Wide-range specifications			–40 to +85	–40 to +85	–40 to +85	–40 t +85*
Absolute maximum ratings	V_{PP} pin rating	V_{in}		Yes	—	Yes	—
	FWE pin rating			—	—	—	Yes
	V_{CL} pin			—	—	—	Cannot connec to pow supply (5 V operati model only)

							model) –0.3 to +4.6 (3 V operatio model)
DC characteristics	$\overline{\text{RESO}}$ pin specification			Yes	Yes	Yes	—
	FWE pin specification			—	—	—	Yes
	Determination level for applying high voltage (12 V)			Yes	—	—	—
	Standby current ($T_a \leq 50°C$)	I_{CC} *3	µA	Max 5	Max 5	Max 5	Max 10
	Standby current ($50°C < T_a$)			Max 20	Max 20	Max 20	Max 80
AC characteristics	Clock cycle time	t_{cyc}	ns	Max 1000	Max 1000	Max 1000	Max 50
	$\overline{\text{RES}}$ pulse width	t_{RESW}	t_{cyc}	Min 10	Min 10	Min 10	Min 20
	$\overline{\text{RESO}}$ output delay time	t_{RESD}	ns	Max 100	Max 100	Max 100	—
	$\overline{\text{RESO}}$ output pulse width	t_{RESOW}	t_{cyc}	Min 132	Min 132	Min 132	—
Flash memory characteristics*4				Refer to the H8/3048 Group Hardware Manual (revision 7.0) for details.	—	—	See tabl 21.11

Notes: 1. The operating temperature range for flash memory programming/erasing is 0
2. Connect an external capacitor between the V_{CL} pin and GND.
3. See the DC Characteristics table for current dissipation during operation.
4. Refer to the program/erase algorithms for details of flash memory characteri

Power supply voltage	V_{CC} *1	5 V operation model: –0.3 to +7.0
		3 V operation model: –0.3 to +4.6
Input voltage (FWE)*2	V_{in}	–0.3 to V_{CC} +0.3
Input voltage (except for port 7)*2	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}	5 V operation model: –0.3 to +7.0
		3 V operation model: –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*3
		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

Notes: 1. Do not apply the power supply voltage to the V_{CL} pin in 5 V operation model an external capacitor between this pin and GND.

2. 12 V must not be applied to any pin, as this may cause permanent damage device.

3. The operating temperature range for flash memory programming/erasing is +75°C.

Item		Symbol	Min	Typ	Max	Unit	Test C
Schmitt trigger input voltages	Port A, $P8_2$ to $P8_0$, PB_3 to PB_0	V_T^-	1.0	—	—	V	
		V_T^+	—	—	$V_{CC} \times 0.7$	V	
		$V_T^+ - V_T^-$	0.4	—	—	V	
Input high voltage	$\overline{RES}$, $\overline{STBY}$, FWE, NMI, MD_2 to MD_0	V_{IH}	V_{CC} –0.7	—	V_{CC} +0.3	V	
	EXTAL		$V_{CC} \times 0.7$	—	V_{CC} +0.3	V	
	Port 7		2.0	—	AV_{CC} +0.3	V	
	Ports 1 to 6, 9, $P8_4$, $P8_3$, PB_7 to PB_4		2.0	—	V_{CC} +0.3	V	
Input low voltage	$\overline{RES}$, $\overline{STBY}$, MD_2 to MD_0, FWE	V_{IL}	–0.3	—	0.5	V	
	NMI, EXTAL, ports 1 to 7, 9, $P8_4$, $P8_3$, PB_7 to PB_4		–0.3	—	0.8	V	
Output high voltage	All output pins	V_{OH}	V_{CC} –0.5	—	—	V	I_{OH} = -
			3.5	—	—	V	I_{OH} = -
Output low voltage	All output pins	V_{OL}	—	—	0.4	V	I_{OL} = 1
	Ports 1, 2, 5, and B		—	—	1.0	V	I_{OL} =
Input leakage current	$\overline{STBY}$, NMI, $\overline{RES}$, FWE, MD_2 to MD_0	$\lvert I_{in} \rvert$	—	—	1.0	µA	V_{in} = 0 V_{CC} –0
	Port 7		—	—	1.0	µA	V_{in} = 0 AV_{CC} -

	NMI		—	—	50	pF	T_a =
	All input pins except NMI, FWE		—	—	15	pF	
Current dissipation*2	Normal operation*5	I_{CC}*6	—	45	60	mA	f = 2
	Sleep mode		—	35	50	mA	
	Module standby mode*4		—	20	25	mA	
	Standby mode*3		—	1	10	µA	$T_a \leq$
			—	—	80	µA	50°C
Analog power supply current	During A/D conversion	AI_{CC}	—	0.5	1.5	mA	AV_{CC}
	During A/D and D/A conversion		—	0.5	1.5	mA	
	Idle		—	0.01	5.0	µA	DAS
Reference current	During A/D conversion	AI_{CC}	—	0.4	0.8	mA	V_{REF}
	During A/D and D/A conversion		—	1.5	3.0	mA	
	Idle		—	0.01	5.0	µA	DAS
RAM standby voltage		V_{RAM}	2.0	—	—	V	

Notes: 1. If the A/D and D/A converters are not used, do not leave the AV_{CC}, AV_{SS}, an open. Connect AV_{CC} and V_{REF} to V_{CC}, and connect AV_{SS} to V_{SS}.

2. Current dissipation values are for V_{IH} min = V_{CC} –0.5 V and V_{IL} max = 0.5 V output pins unloaded and the on-chip pull-up transistors in the off state.

3. The values are for $V_{RAM} \leq V_{CC} < 4.5$ V, V_{IH} min = $V_{CC} \times 0.9$, and V_{IL} max = 0.3

4. Module standby current values apply in sleep mode with all modules halted

5. The current dissipation value for flash memory program/erase operations (T +75°C) is 10 mA (max.) greater than the current dissipation value for norma

voltages	PB_3 to PB_0						
		$V_T^+ - V_T^-$	$V_{CC} \times 0.05$	—	—	V	
Input high voltage	$\overline{RES}$, $\overline{STBY}$, FWE, 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, 9, $P8_4$, $P8_3$, PB_7 to PB_4		$V_{CC} \times 0.7$	—	V_{CC} +0.3	V	
Input low voltage	$\overline{RES}$, $\overline{STBY}$, MD_2 to MD_0, FWE	V_{IL}	–0.3	—	$V_{CC} \times 0.1$	V	
	NMI, EXTAL, ports 1 to 7, 9, $P8_4$, $P8_3$, PB_7 to PB_4		–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, 5, and B		—	—	1.0	V	I_{OL} =
Input leakage current	$\overline{STBY}$, NMI, $\overline{RES}$, FWE, MD_2 to MD_0	$\lvert I_{in} \rvert$	—	—	1.0	µA	V_{in} = V_{CC} -
	Port 7		—	—	1.0	µA	V_{in} = AV_{CC}
Three-state leakage current (off state)	Ports 1 to 6, 8 to B	$\lvert I_{TSI} \rvert$	—	—	1.0	µA	V_{in} = V_{CC} -

Current dissipation*2	Normal operation*5	I_{CC}*6	—	40	60	mA	f = 25
	Sleep mode		—	30	50	mA	
	Module standby mode*4		—	20	25	mA	
	Standby mode*3		—	1	10	µA	$T_a \leq 5$
			—	—	80	µA	50°C
Analog power supply current	During A/D conversion	AI_{CC}	—	0.5	1.5	mA	AV_{CC} =
	During A/D and D/A conversion		—	0.5	1.5	mA	
	Idle		—	0.01	5	µA	DAST
Reference current	During A/D conversion	AI_{CC}	—	0.4	0.8	mA	V_{REF} =
	During A/D and D/A conversion		—	1.5	3	mA	
	Idle		—	0.01	5	µA	DAST
RAM standby voltage		V_{RAM}	2.0	—	—	V	

Notes: 1. If the A/D and D/A converters are not used, do not leave the AV_{CC}, AV_{SS}, and open. Connect AV_{CC} and V_{REF} to V_{CC}, and connect AV_{SS} to V_{SS}.
2. Current dissipation values are for V_{IH} min = V_{CC} –0.5 V and V_{IL} max = 0.5 V w output pins unloaded and the on-chip pull-up transistors in the off state.
3. 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 V
4. Module standby current values apply in sleep mode with all modules halted.
5. The current dissipation value for flash memory program/erase operations (T_a +75°C) is 10 mA (max.) greater than the current dissipation value for normal
6. I_{CC} depends on V_{CC} and f, according to the following expressions.
I_{CC} max. (normal operation) = 6.0 [mA] + 0.60 [mA/(MHz × V)] × V_{CC} × f
I_{CC} max. (sleep mode) = 6.0 [mA] + 0.49 [mA/(MHz × V)] × V_{CC} × f
I_{CC} max. (sleep mode and module standby mode) = 6.0 [mA] + 0.21 [mA/(MH
V_{CC} × f
The typical values of current dissipation are reference values.

Item		Symbol	Conditions A, B Min	Typ	Max
Permissible output low current (per pin)	Ports 1, 2, 5, and B	I_{OL}	—	—	10
	Other output pins		—	—	2.0
Permissible output low current (total)	Total of 28 pins in ports 1, 2, 5, and B	ΣI_{OL}	—	—	80
	Total of all output pins, including the above		—	—	120
Permissible output high current (per pin)	All output pins	I_{OH}	—	—	2.0
Permissible output high current (total)	Total of all output pins	ΣI_{OH}	—	—	40

Notes: 1. To protect chip reliability, do not exceed the output current values in table 2
2. When driving a darlington pair or LED, always insert a current-limiting resis output line, as shown in figures 21.1 and 21.2.

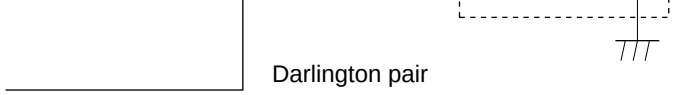


Figure 21.1 Darlington Pair Drive Circuit (Example)

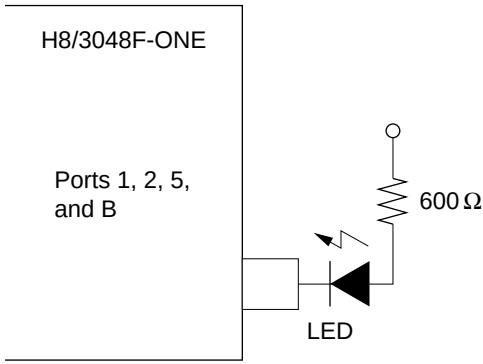


Figure 21.2 LED Drive Circuit (Example)

V_{SS} = AV_{SS} = 0 V, ϕ = 2 MHz to 25 MHz, T_a = –20°C to +75°C (regula
specifications), T_a = –40°C to +85°C (wide-range specifications)

Condition B: V_{CC} = 5.0 V ±10%, AV_{CC} = 5.0 V ±10%, V_{REF} = 4.5 V to AV_{CC},
V_{SS} = AV_{SS} = 0 V, ϕ = 2 MHz to 25 MHz, T_a = –20°C to +75°C (regula
specifications), T_a = –40°C to +85°C (wide-range specifications)

		Condition A		Condition B		
		25 MHz		25 MHz		
Item	**Symbol**	**Min**	**Max**	**Min**	**Max**	**Unit**
Clock cycle time	t_{cyc}	40	500	40	500	ns
Clock pulse low width	t_{CL}	10	—	10	—	
Clock pulse high width	t_{CH}	10	—	10	—	
Clock rise time	t_{CR}	—	10	—	10	
Clock fall time	t_{CF}	—	10	—	10	
Address delay time	t_{AD}	—	28	—	25	
Address hold time	t_{AH}	$0.5t_{cyc}$ –20	—	$0.5t_{cyc}$ –20	—	
Address strobe delay time	t_{ASD}	—	25	—	25	
Write strobe delay time	t_{WSD}	—	25	—	25	
Strobe delay time	t_{SD}	—	25	—	25	
Write data strobe pulse width 1	t_{WSW1}	$1.0t_{cyc}$ –25	—	$1.0t_{cyc}$ –25	—	
Write data strobe pulse width 2	t_{WSW2}	$1.5t_{cyc}$ –25	—	$1.5t_{cyc}$ –25	—	
Address setup time 1	t_{AS1}	$0.5t_{cyc}$ –20	—	$0.5t_{cyc}$ –20	—	
Address setup time 2	t_{AS2}	$1.0t_{cyc}$ –20	—	$1.0t_{cyc}$ –20	—	
Read data setup time	t_{RDS}	15	—	15	—	
Read data hold time	t_{RDH}	0	—	0	—	

	t_{ACC1}		t_{cyc}		t_{cyc}	
Read data access time 2	t_{ACC2}	—	$2.5t_{cyc}$ –40	—	$2.5t_{cyc}$ –40	
Read data access time 3	t_{ACC3}	—	$1.0t_{cyc}$ –28	—	$1.0t_{cyc}$ –28	
Read data access time 4	t_{ACC4}	—	$2.0t_{cyc}$ –32	—	$2.0t_{cyc}$ –32	
Precharge time	t_{PCH}	$1.0t_{cyc}$ –20	—	$1.0t_{cyc}$ –20	—	
Wait setup time	t_{WTS}	25	—	25	—	ns
Wait hold time	t_{WTH}	5	—	5	—	
Bus request setup time	t_{BRQS}	25	—	25	—	ns
Bus acknowledge delay time 1	t_{BACD1}	—	30	—	30	
Bus acknowledge delay time 2	t_{BACD2}	—	30	—	30	
Bus-floating time	t_{BZD}	—	40	—	40	

		Condition A		Condition B		
		25 MHz		25 MHz		
Item	**Symbol**	**Min**	**Max**	**Min**	**Max**	**Unit**
$\overline{RAS}$ delay time 1*1	t_{RAD1}	—	20	—	18	ns
$\overline{RAS}$ delay time 2*1	t_{RAD2}	—	20	—	18	
$\overline{RAS}$ delay time 3*1	t_{RAD3}	—	20	—	18	
Row address hold time	t_{RAH}	$0.5t_{cyc}$ –5	—	$0.5t_{cyc}$ –5	—	
$\overline{RAS}$ precharge time*1	t_{RP}	$1.0t_{cyc}$ –15	—	$1.0t_{cyc}$ –15	—	
$\overline{CAS}$ to $\overline{RAS}$ precharge time*1 *2	t_{CRP}	$1.0t_{cyc}$ –15	—	$1.0t_{cyc}$ –15	—	
$\overline{CAS}$ pulse width*2	t_{CAS}	$1.0t_{cyc}$ –18	—	$1.0t_{cyc}$ –18	—	
$\overline{RAS}$ access time*1	t_{RAC}	—	$2.0t_{cyc}$ –35	—	$2.0t_{cyc}$ –35	
Address access time	t_{AA}	—	$1.5t_{cyc}$ –40	—	$1.5t_{cyc}$ –40	
$\overline{CAS}$ access time*2	t_{CAC}	—	$1.0t_{cyc}$ –30	—	$1.0t_{cyc}$ –30	
Write data setup time 3	t_{WDS3}	$1.0t_{cyc}$ –25	—	$1.0t_{cyc}$ –25	—	
$\overline{CAS}$ setup time*2	t_{CSR}	$0.5t_{cyc}$ –15	—	$0.5t_{cyc}$ –15	—	
Read strobe delay time	t_{RSD}	—	25	—	25	
Signal rise time (all input pins except EXTAL)	t_{SR}	—	100	—	100	ns
Signal fall time (all input pins except EXTAL)	t_{SF}	—	100	—	100	

Notes: 1. The $\overline{RAS}$ pin is assigned to the $\overline{CS3}$ pin.
2. The $\overline{CAS}$ pin is assigned to the $\overline{RD}$ pin.

		Condition A		Condition B		
		25 MHz		25 MHz		
Item	**Symbol**	**Min**	**Max**	**Min**	**Max**	**Unit**
$\overline{RES}$ setup time	t_{RESS}	200	—	200	—	ns
$\overline{RES}$ pulse width	t_{RESW}	20	—	20	—	t_{cyc}
Mode programming setup time	t_{MDS}	200	—	200	—	ns
NMI setup time (NMI, $\overline{IRQ_5}$ to $\overline{IRQ_0}$)	t_{NMIS}	150	—	150	—	ns
NMI hold time (NMI, $\overline{IRQ_5}$ to $\overline{IRQ_0}$)	t_{NMIH}	10	—	10	—	
Interrupt pulse width (NMI, $\overline{IRQ_2}$ to $\overline{IRQ_0}$ when exiting software standby mode)	t_{NMIW}	200	—	200	—	
Clock oscillator settling time at reset (crystal)	t_{OSC1}	20	—	20	—	ms
Clock oscillator settling time in software standby (crystal)	t_{OSC2}	7	—	7	—	ms

Item			Symbol	Condition A 25 MHz Min	Condition A 25 MHz Max	Condition B 25 MHz Min	Condition B 25 MHz Max	Unit
DMAC	$\overline{\text{DREQ}}$ setup time		t_{DRQS}	20	—	20	—	ns
	$\overline{\text{DREQ}}$ hold time		t_{DRQH}	10	—	10	—	
	$\overline{\text{TEND}}$ delay time 1		t_{TED1}	—	50	—	50	
	$\overline{\text{TEND}}$ delay time 2		t_{TED2}	—	50	—	50	
ITU	Timer output delay time		t_{TOCD}	—	50	—	50	ns
	Timer input setup time		t_{TICS}	40	—	40	—	
	Timer clock input setup time		t_{TCKS}	40	—	40	—	
	Timer clock pulse width	Single edge	t_{TCKWH}	1.5	—	1.5	—	t_{CYC}
		Both edges	t_{TCKWL}	2.5	—	2.5	—	
SCI	Input clock cycle	Asynchronous	t_{SCYC}	4	—	4	—	t_{CYC}
		Synchronous	t_{SCYC}	6	—	6	—	
	Input clock rise time		t_{SCKr}	—	1.5	—	1.5	
	Input clock fall time		t_{SCKf}	—	1.5	—	1.5	
	Input clock pulse width		t_{SCKW}	0.4	0.6	0.4	0.6	t_{SCYC}
	Transmit data delay time		t_{TXD}	—	100	—	100	ns
	Receive data setup time (synchronous)		t_{RXS}	100	—	100	—	
	Receive data hold time (synchronous)	Clock input	t_{RXH}	100	—	100	—	
		Clock output	t_{RXH}	0	—	0	—	
Ports and TPC	Output data delay time		t_{PWD}	—	50	—	50	ns
	Input data setup time		t_{PRS}	50	—	50	—	
	Input data hold time		t_{PRH}	50	—	50	—	

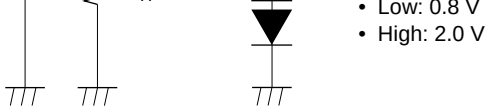


Figure 21.3 Output Load Circuit

specifications), T_a = –40°C to +85°C (wide-range specifications)

Condition B: V_{CC} = 5.0 V ±10%, AV_{CC} = 5.0 V ±10%, V_{REF} = 4.5 V to AV_{CC}, V_{SS} = AV_{SS} = 0 V, ϕ = 2 MHz to 25 MHz, T_a = –20°C to +75°C (regula specifications), T_a = –40°C to +85°C (wide-range specifications)

		Condition A			Condition B		
		25 MHz			25 MHz		
Item		**Min**	**Typ**	**Max**	**Min**	**Typ**	**Max**
Resolution		10	10	10	10	10	10
Conversion time (single mode)		5.36	—	—	5.36	—	—
Analog input capacitance		—	—	20	—	—	20
Permissible signal-source impedance	$\phi \leq$ 13 MHz	—	—	10	—	—	10
	$\phi >$ 13 MHz	—	—	5	—	—	5
Nonlinearity error		—	—	±3.5	—	—	±3.5
Offset error		—	—	±3.5	—	—	±3.5
Full-scale error		—	—	±3.5	—	—	±3.5
Quantization error		—	—	±0.5	—	—	±0.5
Absolute accuracy		—	—	±4.0	—	—	±4.0

Condition B: V_{CC} = 5.0 V ±10%, AV_{CC} = 5.0 V ±10%, V_{REF} = 4.5 V to AV_{CC}, V_{SS} = AV_{SS} = 0 V, ϕ = 2 MHz to 25 MHz, T_a = –20°C to +75°C (regular specifications), T_a = –40°C to +85°C (wide-range specifications)

	Condition A			Condition B				
	25 MHz			25 MHz				
Item	**Min**	**Typ**	**Max**	**Min**	**Typ**	**Max**	**Unit**	**Test C**
Resolution	8	8	8	8	8	8	bits	
Conversion time (centering time)	—	—	10	—	—	10	µs	20-pF load
Absolute accuracy	—	±2.0	±3.0	—	±1.5	±2.0	LSB	2-MΩ load
	—	—	±2.0	—	—	±1.5	LSB	4-MΩ load

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 ti
		t_{sp200}	198	200	202	µs	P ti
		t_{sp10}	8	10	12	µs	A p ti
	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
	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} = 30 µs

Programming counter (n) = 7 to 1000: t_{sp200} = 200 µs

Programming counter (n) [in additional programming] = 1 to 6: t_{sp10} = 10 µs

5. For the maximum erase time (t_E(max)), the following relationship applies betv
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}) x maximum erase count (N)

To set the maximum erase time, the values of t_{se} and N should be set so as t
the above formula.

Examples: When t_{se} = 100 [ms], N = 12

When t_{se} = 10 [ms], N = 120

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 ti
		t_{sp200}	198	200	202	µs	P ti
		t_{sp10}	8	10	12	µs	A p ti
	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
	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 wait time after P bit setting should be changed as follows according to th
the programming counter (n).

Programming counter (n) = 1 to 6: t_{sp30} = 3
Programming counter (n) = 7 to 1000: t_{sp200} = 2
Programming counter (n) [in additional programming] = 1 to 6: t_{sp10} = 1

5. For the maximum erase time (t_E(max)), the following relationship applies betw
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 t
the above formula.

Examples: When t_{se} = 100 [ms], N = 12 times
When t_{se} = 10 [ms], N = 120 times

Power supply voltage	V_{CC} *1	5 V operation model: –0.3 to +7.0
		3 V operation model: –0.3 to +4.6
Input voltage (FWE)*2	V_{in}	–0.3 to V_{CC} +0.3
Input voltage (except for port 7)*2	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}	5 V operation model: –0.3 to +7.0
		3 V operation model: –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*3
		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

Notes: 1. Do not apply the power supply voltage to the V_{CL} pin in 5 V operation model an external capacitor between this pin and GND.

2. 12 V must not be applied to any pin, as this may cause permanent damage device.

3. The operating temperature range for flash memory programming/erasing is +75°C.

Item		Symbol	Min	Typ	Max	Unit	Test C
Schmitt trigger input voltages	Port A, $P8_2$ to $P8_0$, PB_3 to PB_0	V_T^-	1.0	—	—	V	
		V_T^+	—	—	$V_{CC} \times 0.7$	V	
		$V_T^+ - V_T^-$	0.4	—	—	V	
Input high voltage	$\overline{RES}$, $\overline{STBY}$, NMI, MD_2 to MD_0	V_{IH}	V_{CC} –0.7	—	V_{CC} +0.3	V	
	EXTAL		$V_{CC} \times 0.7$	—	V_{CC} +0.3	V	
	Port 7		2.0	—	AV_{CC} +0.3	V	
	Ports 1 to 6, 9, $P8_4$, $P8_3$, PB_7 to PB_4		2.0	—	V_{CC} +0.3	V	
Input low voltage	$\overline{RES}$, $\overline{STBY}$, MD_2 to MD_0	V_{IL}	–0.3	—	0.5	V	
	NMI, EXTAL, ports 1 to 7, 9, $P8_4$, $P8_3$, PB_7 to PB_4		–0.3	—	0.8	V	
Output high voltage	All output pins (Except $\overline{RESO}$)	V_{OH}	V_{CC} –0.5	—	—	V	I_{OH} = -
			3.5	—	—	V	I_{OH} = -
Output low voltage	All output pins (Except $\overline{RESO}$)	V_{OL}	—	—	0.4	V	I_{OL} = 1
	Ports 1, 2, 5, and B		—	—	1.0	V	I_{OL} = 1
	$\overline{RESO}$		—	—	0.4		I_{OL} = 1
Input leakage current	$\overline{STBY}$, NMI, $\overline{RES}$, MD_2 to MD_0	$\lvert I_{in} \rvert$	—	—	1.0	µA	V_{in} = 0 V_{CC} –0
	Port 7		—	—	1.0	µA	V_{in} = 0 AV_{CC} -

	All input pins except NMI		—	—	15	pF	T_a =
Current dissipation*2	Normal operation	I_{CC}*5	—	45	60	mA	f = 2
	Sleep mode		—	35	50	mA	
	Module standby mode*4		—	20	25	mA	
	Standby mode*3		—	1	10	µA	$T_a \leq$
			—	—	80	µA	50°C
Analog power supply current	During A/D conversion	AI_{CC}	—	0.5	1.5	mA	AV_{CC}
	During A/D and D/A conversion		—	0.5	1.5	mA	
	Idle		—	0.01	5.0	µA	DAS
Reference current	During A/D conversion	AI_{CC}	—	0.4	0.8	mA	V_{REF}
	During A/D and D/A conversion		—	1.5	3.0	mA	
	Idle		—	0.01	5.0	µA	DAS
RAM standby voltage		V_{RAM}	2.0	—	—	V	

Notes: 1. If the A/D and D/A converters are not used, do not leave the AV_{CC}, AV_{SS}, an open. Connect AV_{CC} and V_{REF} to V_{CC}, and connect AV_{SS} to V_{SS}.

2. Current dissipation values are for V_{IH} min = V_{CC} –0.5 V and V_{IL} max = 0.5 V output pins unloaded and the on-chip pull-up transistors in the off state.

3. The values are for $V_{RAM} \leq V_{CC} < 4.5$ V, V_{IH} min = $V_{CC} \times 0.9$, and V_{IL} max = 0.3

4. Module standby current values apply in sleep mode with all modules halted

5. I_{CC} depends on V_{CC} and f, according to the following expressions.
[Applicable operating frequency: 2 to 25 MHz]
I_{CC} max. (normal operation) = 5.0 [mA] + 0.32 [mA/(MHz × V)] × V_{CC} × (f –2)
I_{CC} max. (sleep mode) = 7.0 [mA] + 0.26 [mA/(MHz × V)] × V_{CC} × (f –2)
I_{CC} max. (sleep mode and module standby mode) = 6.0 [mA] + 0.11 [mA/(M
V_{CC} × (f –2)
The typical values of current dissipation are reference values.

voltages	PB_3 to PB_0						
		$V_T^+ - V_T^-$	$V_{CC} \times 0.05$	—	—	V	
Input high voltage	$\overline{RES}$, $\overline{STBY}$, NMI, MD_2 to MD_0	V_{IH}	$V_{CC} \times 0.9$	—	V_{CC} +0.3	V	
	EXTAL		$V_{CC} \times 0.7$	—	V_{CC} +0.3	V	
	Port 7		$V_{CC} \times 0.7$	—	AV_{CC} +0.3	V	
	Ports 1 to 6, 9, $P8_4$, $P8_3$, PB_7 to PB_4		$V_{CC} \times 0.7$	—	V_{CC} +0.3	V	
Input low voltage	$\overline{RES}$, $\overline{STBY}$, MD_2 to MD_0	V_{IL}	–0.3	—	$V_{CC} \times 0.1$	V	
	NMI, EXTAL, ports 1 to 7, 9, $P8_4$, $P8_3$, PB_7 to PB_4		–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} = 1
	Ports 1, 2, 5, and B		—	—	1.0	V	I_{OL} = 5
	$\overline{RESO}$		—	—	0.4		I_{OL} = 1
Input leakage current	$\overline{STBY}$, NMI, $\overline{RES}$, MD_2 to MD_0	$\lvert I_{in} \rvert$	—	—	1.0	µA	V_{in} = 0 V_{CC} –0
	Port 7		—	—	1.0	µA	V_{in} = 0 AV_{CC} -

	All input pins except NMI		—	—	15	pF	T_a =
Current dissipation*2	Normal operation	I_{CC}*5	—	40	50	mA	f = 2
	Sleep mode		—	25	40	mA	
	Module standby mode*4		—	15	20	mA	
	Standby mode*3		—	1	10	µA	$T_a \leq$
			—	—	80	µA	50°C
Analog power supply current	During A/D conversion	AI_{CC}	—	0.5	1.5	mA	AV_{CC}
	During A/D and D/A conversion		—	0.5	1.5	mA	
	Idle		—	0.01	5	µA	DAS
Reference current	During A/D conversion	AI_{CC}	—	0.4	0.8	mA	V_{REF}
	During A/D and D/A conversion		—	1.5	3	mA	
	Idle		—	0.01	5	µA	DAS
RAM standby voltage		V_{RAM}	2.0	—	—	V	

Notes: 1. If the A/D and D/A converters are not used, do not leave the AV_{CC}, AV_{SS}, an open. Connect AV_{CC} and V_{REF} to V_{CC}, and connect AV_{SS} to V_{SS}.

2. Current dissipation values are for V_{IH} min = V_{CC} –0.5 V and V_{IL} max = 0.5 V output pins unloaded and the on-chip pull-up transistors in the off state.

3. The values are for $V_{RAM} \leq V_{CC} < 3.0$ V, V_{IH} min = $V_{CC} \times 0.9$, and V_{IL} max = 0.3

4. Module standby current values apply in sleep mode with all modules halted

5. I_{CC} depends on V_{CC} and f, according to the following expressions.
[Applicable operating frequency: 2 to 25 MHz]
I_{CC} max. (normal operation) = 6.0 [mA] + 0.53 [mA/(MHz × V)] × V_{CC} × (f –2)
I_{CC} max. (sleep mode) = 4.0 [mA] + 0.43 [mA/(MHz × V)] × V_{CC} × (f –2)
I_{CC} max. (sleep mode and module standby mode) = 3.0 [mA] + 0.20 [mA/(M
V_{CC} × (f –2)
The typical values of current dissipation are reference values.

Item		Symbol	Conditions A, B		
			Min	Typ	Max
Permissible output low current (per pin)	Ports 1, 2, 5, and B	I_{OL}	—	—	10
	Other output pins		—	—	2.0
Permissible output low current (total)	Total of 28 pins in ports 1, 2, 5, and B	ΣI_{OL}	—	—	80
	Total of all output pins, including the above		—	—	120
Permissible output high current (per pin)	All output pins	I_{OH}	—	—	2.0
Permissible output high current (total)	Total of all output pins	ΣI_{OH}	—	—	40

Notes: 1. To protect chip reliability, do not exceed the output current values in table 21

2. When driving a darlington pair or LED, always insert a current-limiting resisto output line, as shown in figures 21.4 and 21.5.

Figure 21.4 Darlington Pair Drive Circuit (Example)

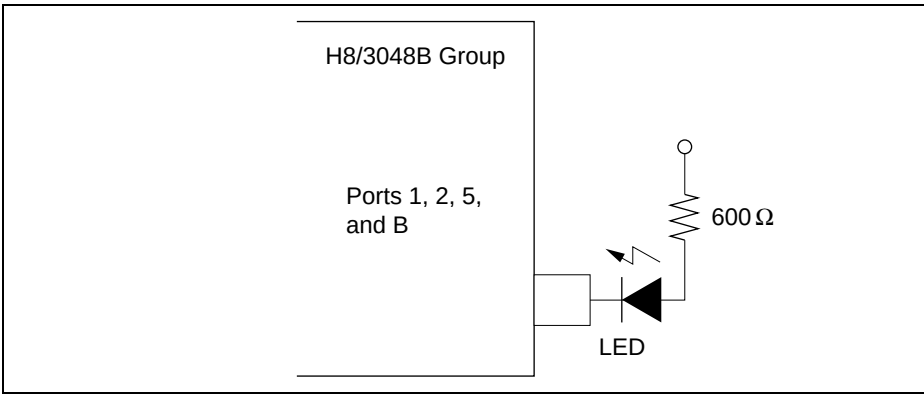


Figure 21.5 LED Drive Circuit (Example)

V_{SS} = AV_{SS} = 0 V, ϕ = 2 MHz to 25 MHz, T_a = –20°C to +75°C (regular specifications), T_a = –40°C to +85°C (wide-range specifications)

Condition B: V_{CC} = 5.0 V ±10%, AV_{CC} = 5.0 V ±10%, V_{REF} = 4.5 V to AV_{CC}, V_{SS} = AV_{SS} = 0 V, ϕ = 2 MHz to 25 MHz, T_a = –20°C to +75°C (regular specifications), T_a = –40°C to +85°C (wide-range specifications)

		Condition A		Condition B		
		25 MHz		25 MHz		
Item	**Symbol**	**Min**	**Max**	**Min**	**Max**	**Unit**
Clock cycle time	t_{cyc}	40	500	40	500	ns
Clock pulse low width	t_{CL}	10	—	10	—	
Clock pulse high width	t_{CH}	10	—	10	—	
Clock rise time	t_{CR}	—	10	—	10	
Clock fall time	t_{CF}	—	10	—	10	
Address delay time	t_{AD}	—	28	—	25	
Address hold time	t_{AH}	0.5t_{cyc} –20	—	0.5t_{cyc} –20	—	
Address strobe delay time	t_{ASD}	—	25	—	25	
Write strobe delay time	t_{WSD}	—	25	—	25	
Strobe delay time	t_{SD}	—	25	—	25	
Write data strobe pulse width 1	t_{WSW1}	1.0t_{cyc} –25	—	1.0t_{cyc} –25	—	
Write data strobe pulse width 2	t_{WSW2}	1.5t_{cyc} –25	—	1.5t_{cyc} –25	—	
Address setup time 1	t_{AS1}	0.5t_{cyc} –20	—	0.5t_{cyc} –20	—	
Address setup time 2	t_{AS2}	1.0t_{cyc} –20	—	1.0t_{cyc} –20	—	
Read data setup time	t_{RDS}	15	—	15	—	
Read data hold time	t_{RDH}	0	—	0	—	

	t_{ACC1}		t_{cyc}		t_{cyc}	
Read data access time 2	t_{ACC2}	—	$2.5t_{cyc}$ –40	—	$2.5t_{cyc}$ –40	
Read data access time 3	t_{ACC3}	—	$1.0t_{cyc}$ –28	—	$1.0t_{cyc}$ –28	
Read data access time 4	t_{ACC4}	—	$2.0t_{cyc}$ –32	—	$2.0t_{cyc}$ –32	
Precharge time	t_{PCH}	$1.0t_{cyc}$ –20	—	$1.0t_{cyc}$ –20	—	
Wait setup time	t_{WTS}	25	—	25	—	ns
Wait hold time	t_{WTH}	5	—	5	—	
Bus request setup time	t_{BRQS}	25	—	25	—	ns
Bus acknowledge delay time 1	t_{BACD1}	—	30	—	30	
Bus acknowledge delay time 2	t_{BACD2}	—	30	—	30	
Bus-floating time	t_{BZD}	—	40	—	40	

		Condition A		Condition B		
		25 MHz		25 MHz		
Item	**Symbol**	**Min**	**Max**	**Min**	**Max**	**Unit**
$\overline{RAS}$ delay time 1*1	t_{RAD1}	—	20	—	18	ns
$\overline{RAS}$ delay time 2*1	t_{RAD2}	—	20	—	18	
$\overline{RAS}$ delay time 3*1	t_{RAD3}	—	20	—	18	
Row address hold time	t_{RAH}	$0.5t_{cyc}$ –5	—	$0.5t_{cyc}$ –5	—	
$\overline{RAS}$ precharge time*1	t_{RP}	$1.0t_{cyc}$ –15	—	$1.0t_{cyc}$ –15	—	
$\overline{CAS}$ to $\overline{RAS}$ precharge time*1 *2	t_{CRP}	$1.0t_{cyc}$ –15	—	$1.0t_{cyc}$ –15	—	
$\overline{CAS}$ pulse width*2	t_{CAS}	$1.0t_{cyc}$ –18	—	$1.0t_{cyc}$ –18	—	
$\overline{RAS}$ access time*1	t_{RAC}	—	$2.0t_{cyc}$ –35	—	$2.0t_{cyc}$ –35	
Address access time	t_{AA}	—	$1.5t_{cyc}$ –40	—	$1.5t_{cyc}$ –40	
$\overline{CAS}$ access time*2	t_{CAC}	—	$1.0t_{cyc}$ –30	—	$1.0t_{cyc}$ –30	
Write data setup time 3	t_{WDS3}	$1.0t_{cyc}$ –25	—	$1.0t_{cyc}$ –25	—	
$\overline{CAS}$ setup time*2	t_{CSR}	$0.5t_{cyc}$ –15	—	$0.5t_{cyc}$ –15	—	
Read strobe delay time	t_{RSD}	—	25	—	25	
Signal rise time (all input pins except EXTAL)	t_{SR}	—	100	—	100	ns
Signal fall time (all input pins except EXTAL)	t_{SF}	—	100	—	100	

Notes: 1. The $\overline{RAS}$ pin is assigned to the $\overline{CS3}$ pin.
2. The $\overline{CAS}$ pin is assigned to the $\overline{RD}$ pin.

		Condition A		Condition B		
		25 MHz		25 MHz		
Item	**Symbol**	**Min**	**Max**	**Min**	**Max**	**Unit**
$\overline{\text{RES}}$ setup time	t_{RESS}	200	—	200	—	ns
$\overline{\text{RES}}$ pulse width	t_{RESW}	20	—	20	—	t_{cyc}
Mode programming setup time	t_{MDS}	200	—	200	—	ns
$\overline{\text{RESO}}$ output delay time	t_{RESD}	—	50	—	50	ns
$\overline{\text{RESO}}$ output pulse width	t_{RESOW}	132	—	132	—	t_{cyc}
NMI setup time (NMI, $\overline{\text{IRQ}}_5$ to $\overline{\text{IRQ}}_0$)	t_{NMIS}	150	—	150	—	ns
NMI hold time (NMI, $\overline{\text{IRQ}}_5$ to $\overline{\text{IRQ}}_0$)	t_{NMIH}	10	—	10	—	
Interrupt pulse width (NMI, $\overline{\text{IRQ}}_2$ to $\overline{\text{IRQ}}_0$ when exiting software standby mode)	t_{NMIW}	200	—	200	—	
Clock oscillator settling time at reset (crystal)	t_{OSC1}	20	—	20	—	ms
Clock oscillator settling time in software standby (crystal)	t_{OSC2}	7	—	7	—	ms

Item			Symbol	Condition A 25 MHz Min	Condition A 25 MHz Max	Condition B 25 MHz Min	Condition B 25 MHz Max	Unit	T C
DMAC	$\overline{\text{DREQ}}$ setup time		t_{DRQS}	20	—	20	—	ns	F
	$\overline{\text{DREQ}}$ hold time		t_{DRQH}	10	—	10	—		
	$\overline{\text{TEND}}$ delay time 1		t_{TED1}	—	50	—	50		F
	$\overline{\text{TEND}}$ delay time 2		t_{TED2}	—	50	—	50		a
ITU	Timer output delay time		t_{TOCD}	—	50	—	50	ns	F
	Timer input setup time		t_{TICS}	40	—	40	—		
	Timer clock input setup time		t_{TCKS}	40	—	40	—		F
	Timer clock pulse width	Single edge	t_{TCKWH}	1.5	—	1.5	—	t_{CYC}	
		Both edges	t_{TCKWL}	2.5	—	2.5	—		
SCI	Input clock cycle	Asynchronous	t_{SCYC}	4	—	4	—	t_{CYC}	F
		Synchronous	t_{SCYC}	6	—	6	—		
	Input clock rise time		t_{SCKr}	—	1.5	—	1.5		
	Input clock fall time		t_{SCKf}	—	1.5	—	1.5		
	Input clock pulse width		t_{SCKW}	0.4	0.6	0.4	0.6	t_{SCYC}	
	Transmit data delay time		t_{TXD}	—	100	—	100	ns	F
	Receive data setup time (synchronous)		t_{RXS}	100	—	100	—		
	Receive data hold time (synchronous)	Clock input	t_{RXH}	100	—	100	—		
		Clock output	t_{RXH}	0	—	0	—		
Ports and TPC	Output data delay time		t_{PWD}	—	50	—	50	ns	F
	Input data setup time		t_{PRS}	50	—	50	—		
	Input data hold time		t_{PRH}	50	—	50	—		

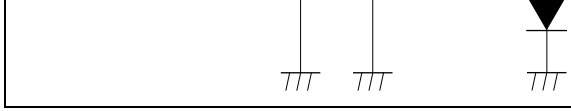


Figure 21.6 Output Load Circuit

Condition B: V_{CC} = 5.0 V ±10%, AV_{CC} = 5.0 V ±10%, V_{REF} = 4.5 V to AV_{CC}, V_{SS} = AV_{SS} = 0 V, ϕ = 2 MHz to 25 MHz, T_a = –20°C to +75°C (regular specifications), T_a = –40°C to +85°C (wide-range specifications)

		Condition A			Condition B		
		25 MHz			25 MHz		
Item		**Min**	**Typ**	**Max**	**Min**	**Typ**	**Max**
Resolution		10	10	10	10	10	10
Conversion time (single mode)		5.36	—	—	5.36	—	—
Analog input capacitance		—	—	20	—	—	20
Permissible signal-source impedance	$\phi \leq$ 13 MHz	—	—	10	—	—	10
	$\phi >$ 13 MHz	—	—	5	—	—	5
Nonlinearity error		—	—	±3.5	—	—	±3.5
Offset error		—	—	±3.5	—	—	±3.5
Full-scale error		—	—	±3.5	—	—	±3.5
Quantization error		—	—	±0.5	—	—	±0.5
Absolute accuracy		—	—	±4.0	—	—	±4.0

Condition B: V_{CC} = 5.0 V ±10%, AV_{CC} = 5.0 V ±10%, V_{REF} = 4.5 V to AV_{CC}, V_{SS} = AV_{SS} = 0 V, ϕ = 2 MHz to 25 MHz, T_a = –20°C to +75°C (regula specifications), T_a = –40°C to +85°C (wide-range specifications)

	Condition A			Condition B				
	25 MHz			25 MHz				
Item	**Min**	**Typ**	**Max**	**Min**	**Typ**	**Max**	**Unit**	**Test**
Resolution	8	8	8	8	8	8	bits	
Conversion time (centering time)	—	—	10	—	—	10	µs	20-p load
Absolute accuracy	—	±2.0	±3.0	—	±1.5	±2.0	LSB	2-MΩ load
	—	—	±2.0	—	—	±1.5	LSB	4-MΩ load

- Basic bus cycle: two-state access

  Figure 21.7 shows the timing of the external two-state access cycle.
- Basic bus cycle: three-state access

  Figure 21.8 shows the timing of the external three-state access cycle.
- Basic bus cycle: three-state access with one wait state

  Figure 21.9 shows the timing of the external three-state access cycle with one wait s
  inserted.

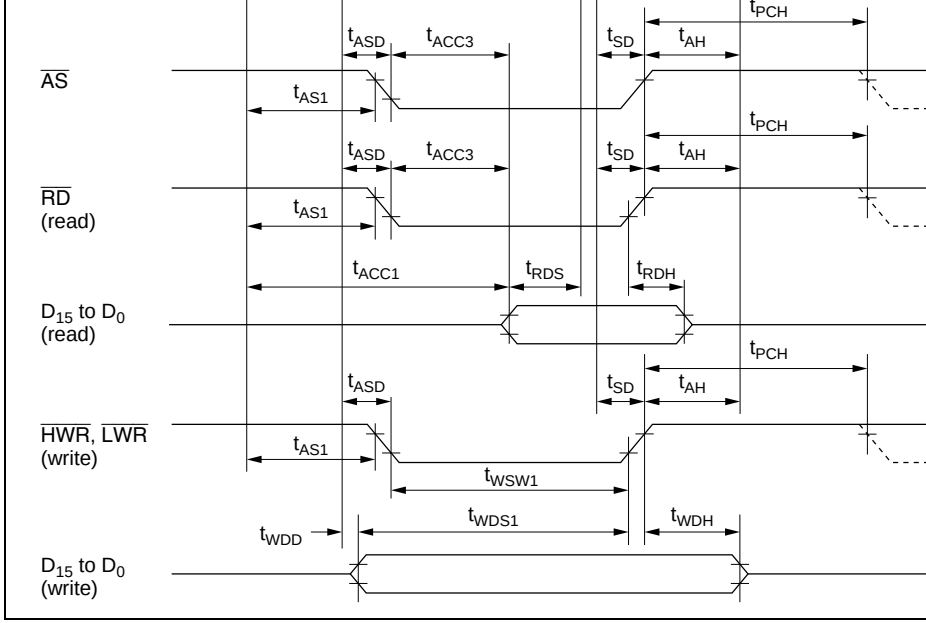


Figure 21.7 Basic Bus Cycle: Two-State Access

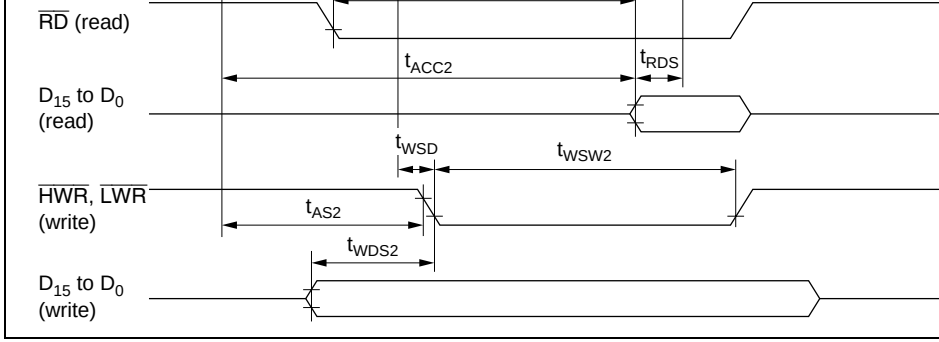


Figure 21.8 Basic Bus Cycle: Three-State Access

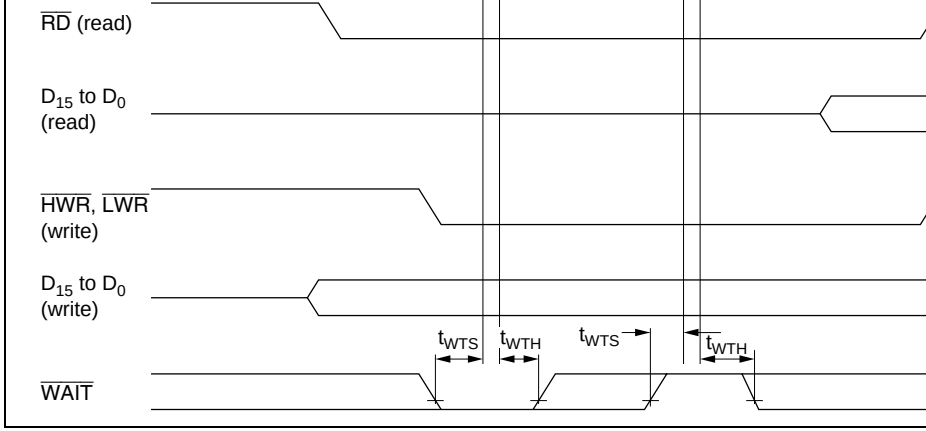


Figure 21.9 Basic Bus Cycle: Three-State Access with One Wait Stat

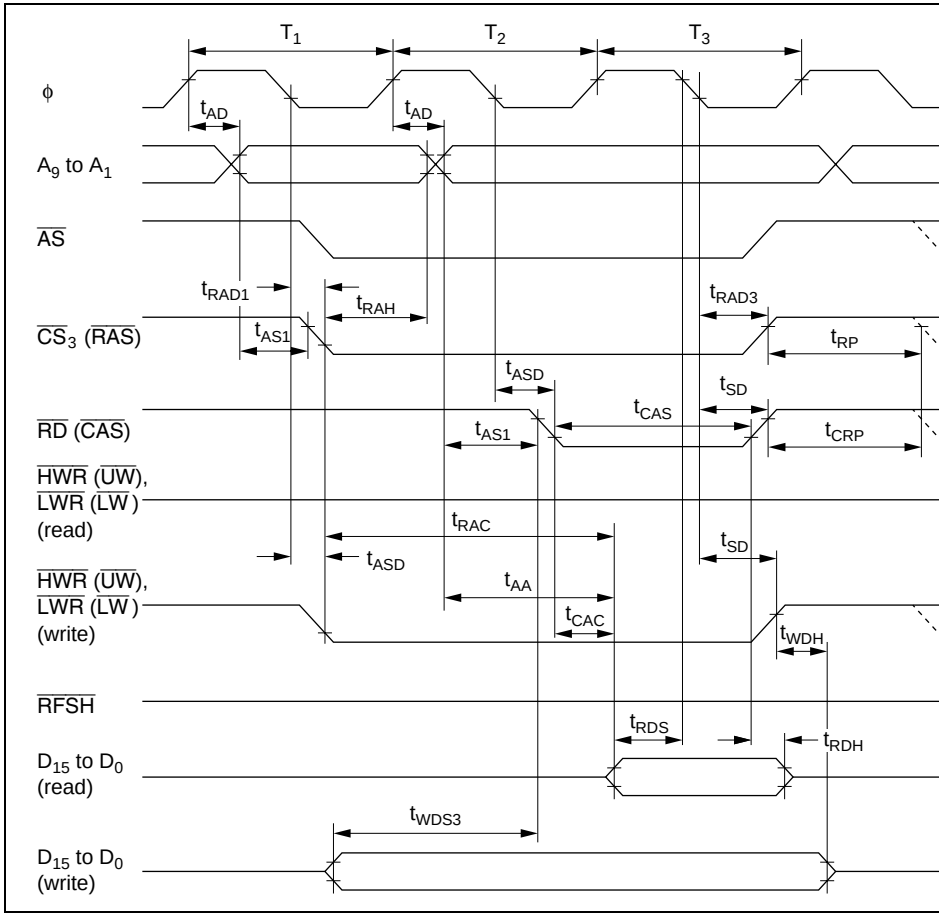


Figure 21.10 DRAM Bus Timing (Read/Write): Three-State Access — 2$\overline{WE}$ Mode —

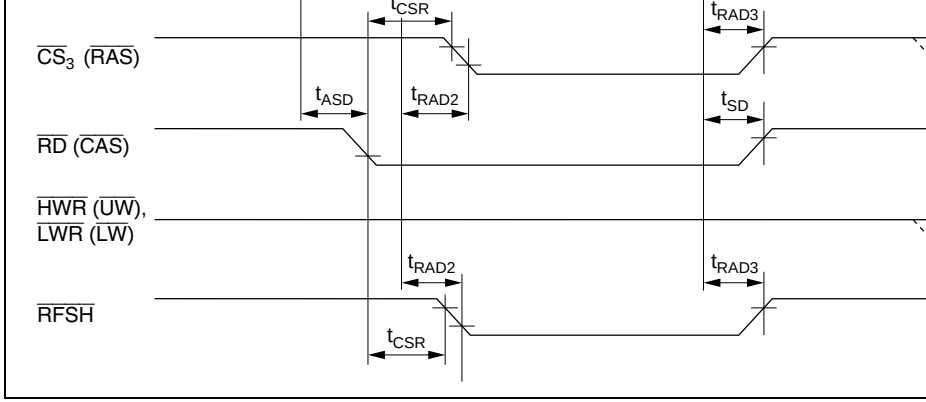


Figure 21.11 DRAM Bus Timing (Refresh Cycle): Three-State Acces
— 2WE Mode —

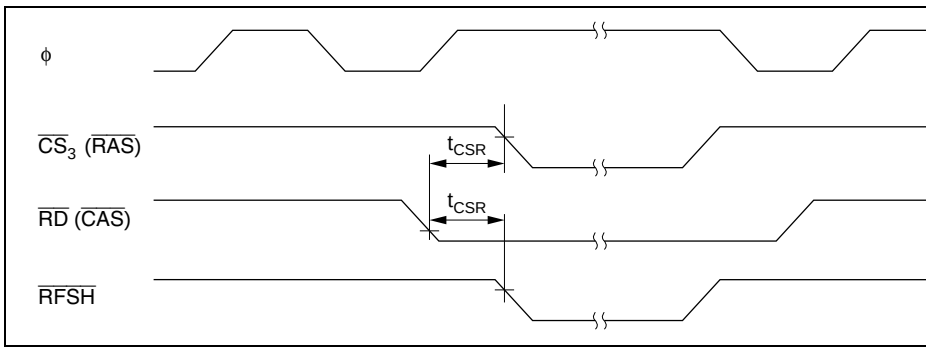


Figure 21.12 DRAM Bus Timing (Self-Refresh Mode)
— 2WE Mode —

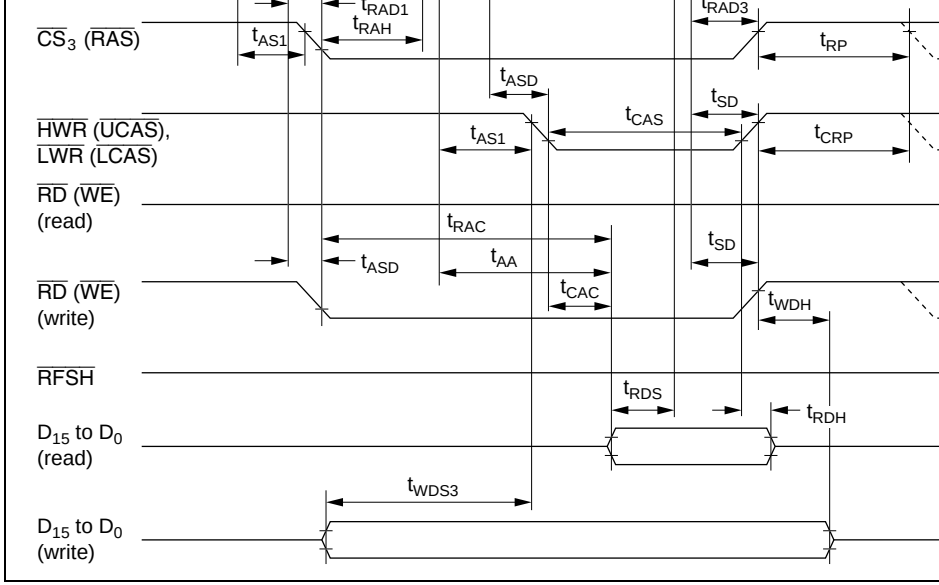


Figure 21.13 DRAM Bus Timing (Read/Write): Three-State Access — 2$\overline{CAS}$ Mode —

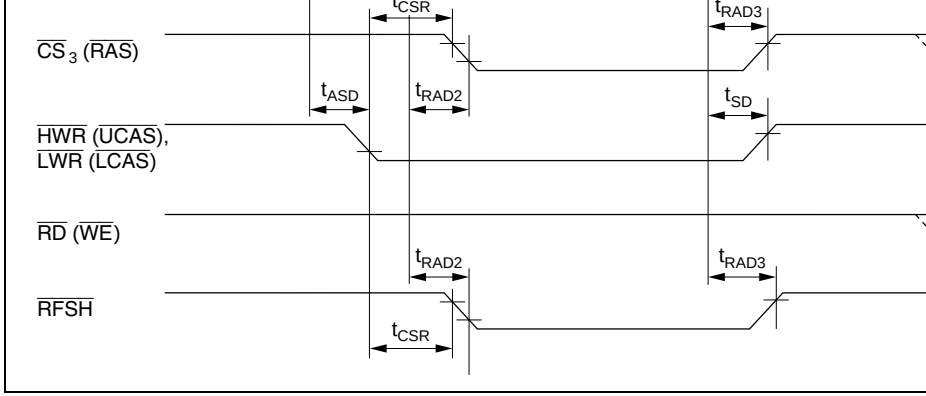


Figure 21.14 DRAM Bus Timing (Refresh Cycle): Three-State Acces
— 2$\overline{CAS}$ Mode —

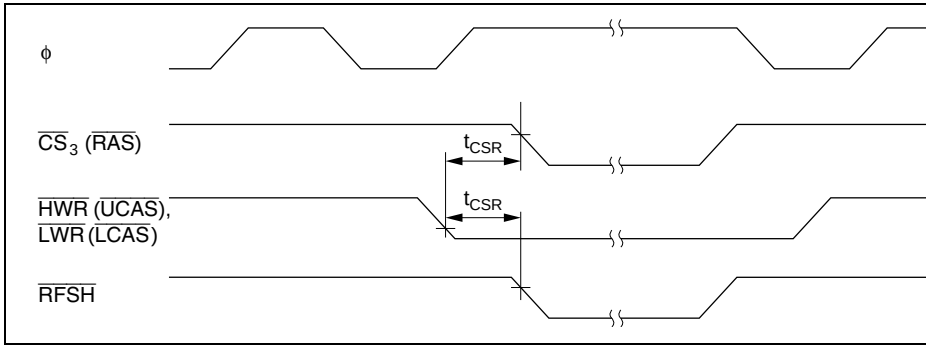


Figure 21.15 DRAM Bus Timing (Self-Refresh Mode)
— 2$\overline{CAS}$ Mode —

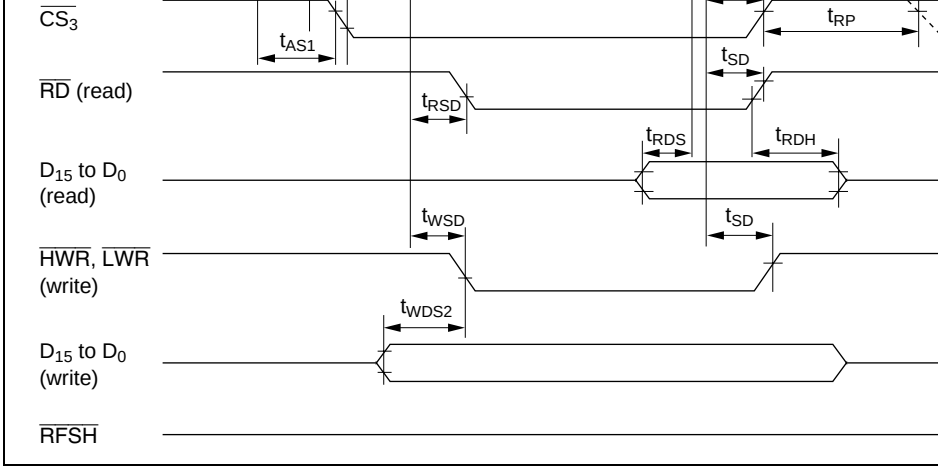


Figure 21.16 PSRAM Bus Timing (Read/Write): Three-State Access

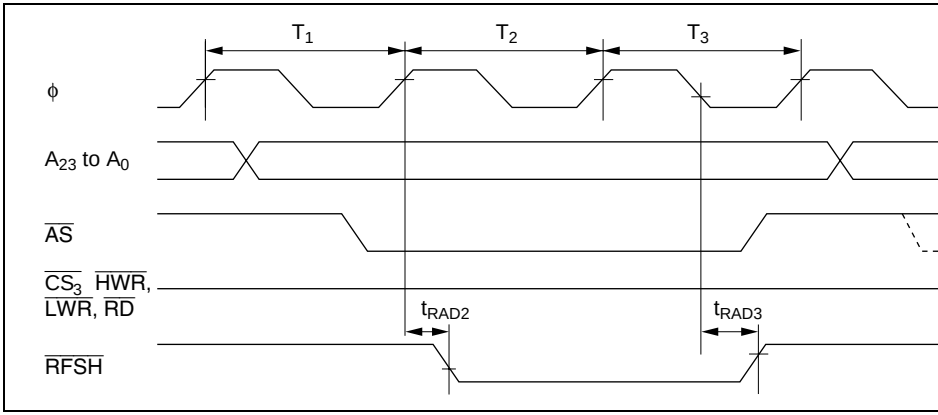


Figure 21.17 PSRAM Bus Timing (Refresh Cycle): Three-State Access

- Interrupt input timing
  Figure 21.20 shows the input timing for NMI and $\overline{IRQ_5}$ to $\overline{IRQ_0}$.
- Bus-release mode timing
  Figure 21.21 shows the bus-release mode timing.

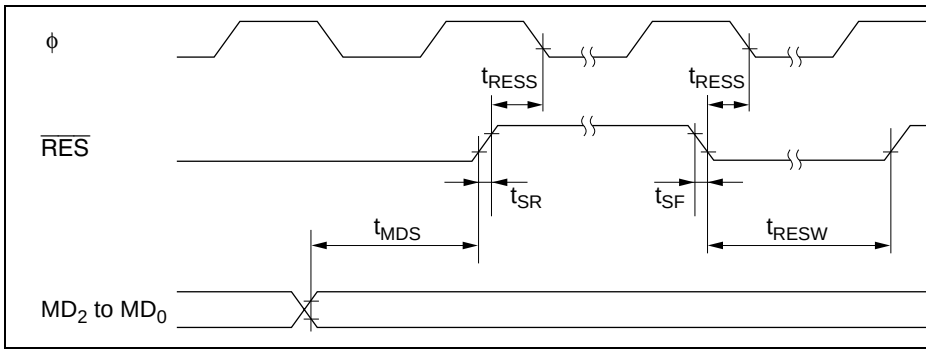


Figure 21.18 Reset Input Timing

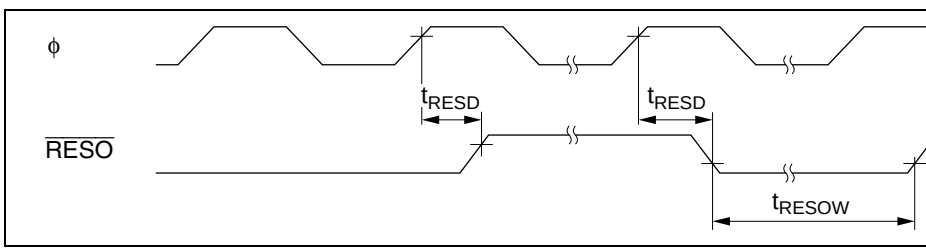


Figure 21.19 Reset Output Timing*

Note: * This is a function for models with on-chip mask ROM (H8/3048B, H8/304
H8/3045, and H8/3044), PROM (H8/3048ZTAT), and on-chip flash memo
dual power supply (H8/3048F). The function does not exist in the product
flash memory with a single power supply (H8/3048F-ONE).

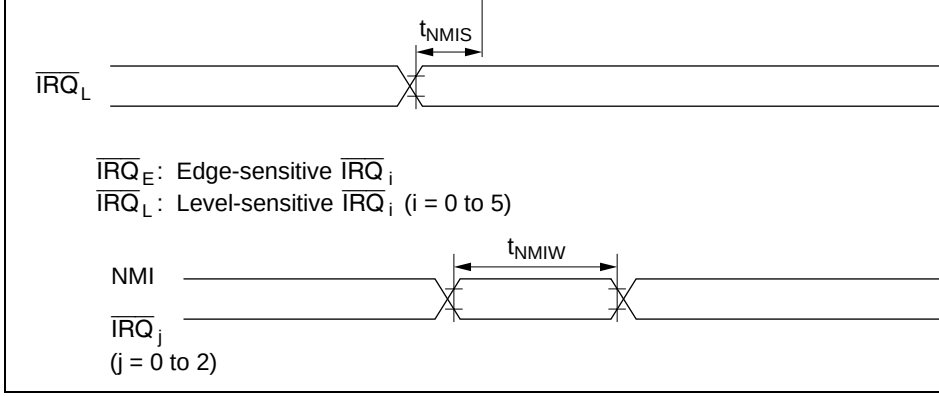


Figure 21.20 Interrupt Input Timing

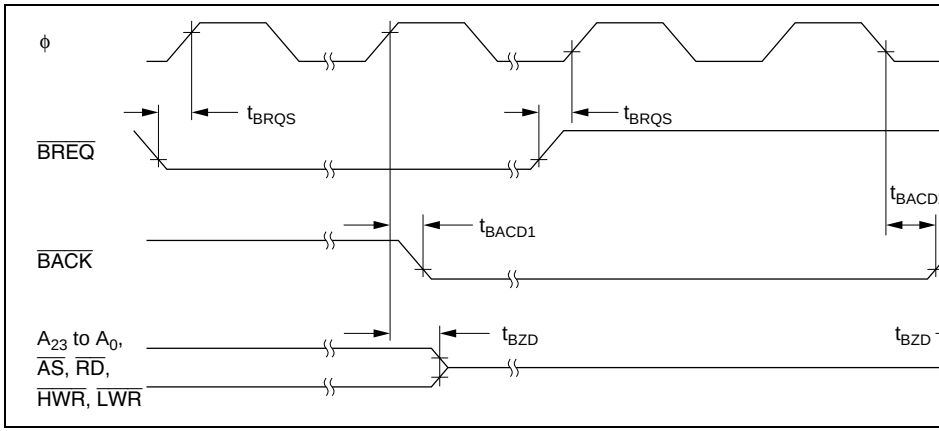


Figure 21.21 Bus-Release Mode Timing

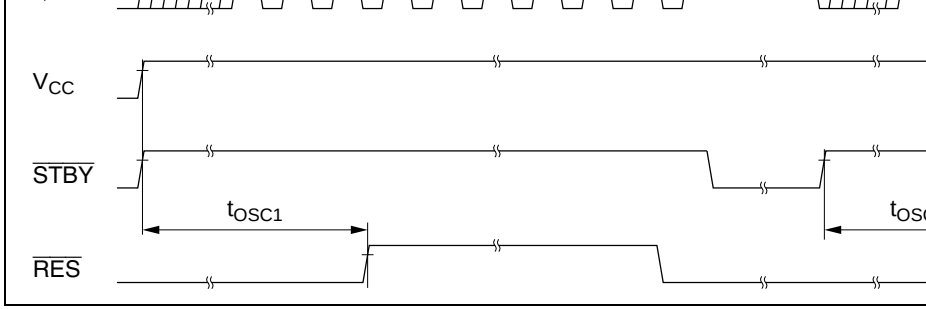


Figure 21.22 Oscillator Settling Timing

21.3.5 TPC and I/O Port Timing

Figure 21.23 shows the TPC and I/O port timing.

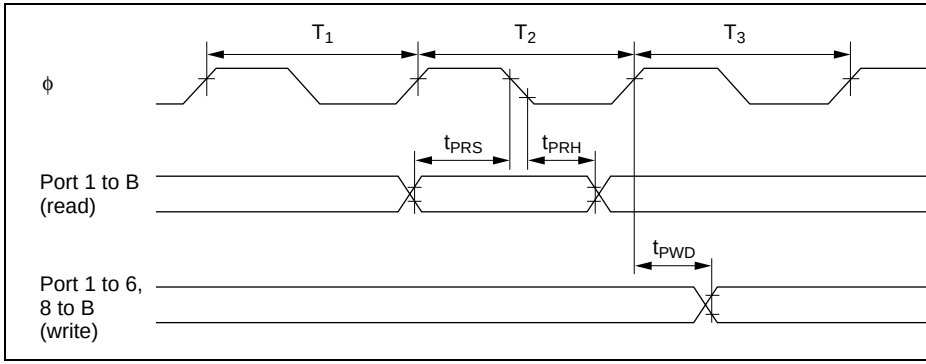
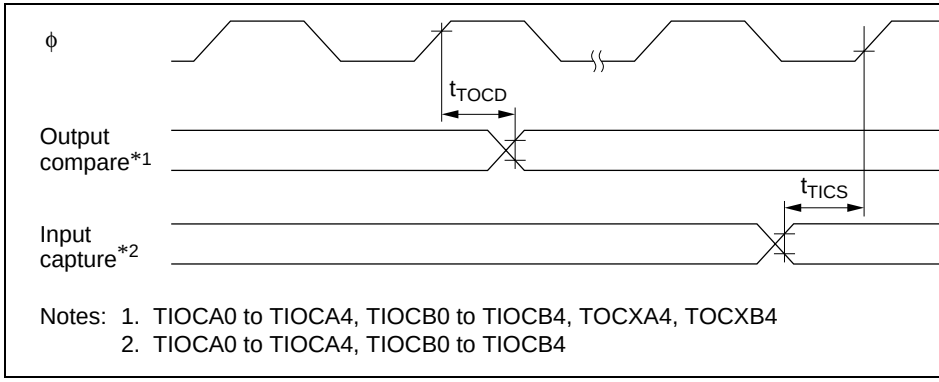


Figure 21.23 TPC and I/O Port Input/Output Timing



Notes: 1. TIOCA0 to TIOCA4, TIOCB0 to TIOCB4, TOCXA4, TOCXB4
2. TIOCA0 to TIOCA4, TIOCB0 to TIOCB4

Figure 21.24 ITU Input/Output Timing

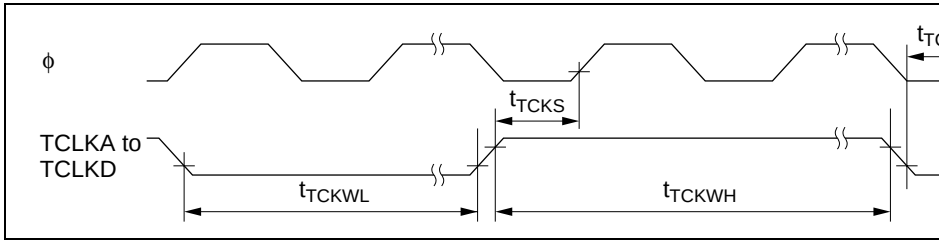


Figure 21.25 ITU External Clock Input Timing

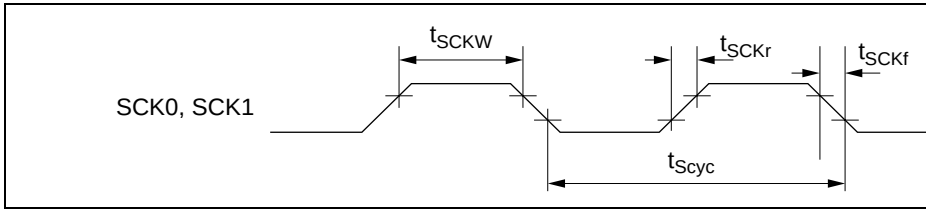


Figure 21.26 SCK Input Clock Timing

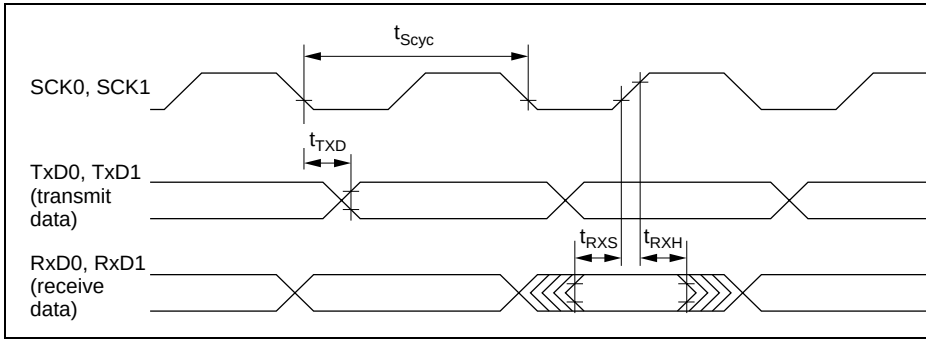


Figure 21.27 SCI Input/Output Timing in Synchronous Mode

- DMAC DREQ input timing

  Figure 21.30 shows DMAC $\overline{\text{DREQ}}$ input timing.

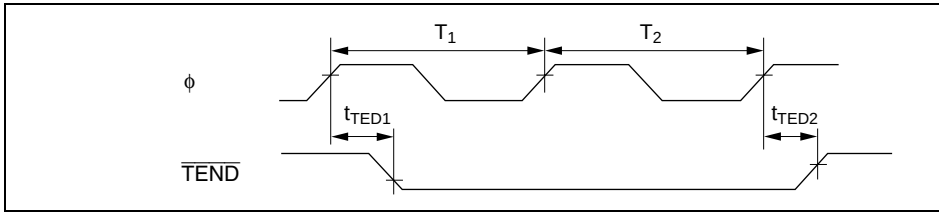


Figure 21.28 DMAC $\overline{\text{TEND}}$ Output Timing for 2 State Access

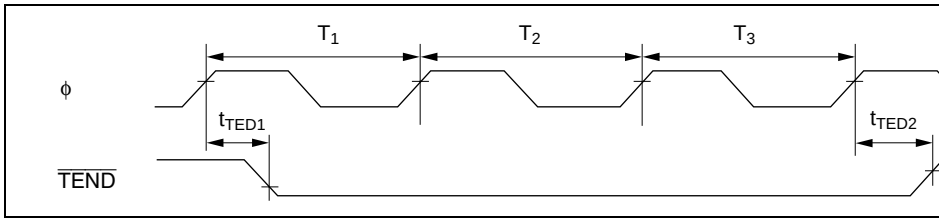


Figure 21.29 DMAC $\overline{\text{TEND}}$ Output Timing for 3 State Access

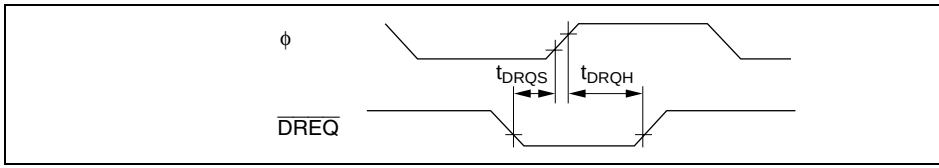


Figure 21.30 DMAC $\overline{\text{DREQ}}$ Input Timing

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 tran 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

	Op		#xx	Rn	@ERn	@(d:16, ERn)	@–ERn/@ERn+	@aa	@(d:24, ERn)	@@aa	—	I	H	N	Z	V
MOV.B #xx:8, Rd	B	#xx:8 → Rd8	2									—	—	↕	↕	0
MOV.B Rs, Rd	B	Rs8 → Rd8		2								—	—	↕	↕	0
MOV.B @ERs, Rd	B	@ERs → Rd8			2							—	—	↕	↕	0
MOV.B @(d:16, ERs), Rd	B	@(d:16, ERs) → Rd8				4						—	—	↕	↕	0
MOV.B @(d:24, ERs), Rd	B	@(d:24, ERs) → Rd8				8						—	—	↕	↕	0
MOV.B @ERs+, Rd	B	@ERs → Rd8, ERs32+1 → ERs32					2					—	—	↕	↕	0
MOV.B @aa:8, Rd	B	@aa:8 → Rd8						2				—	—	↕	↕	0
MOV.B @aa:16, Rd	B	@aa:16 → Rd8						4				—	—	↕	↕	0
MOV.B @aa:24, Rd	B	@aa:24 → Rd8						6				—	—	↕	↕	0
MOV.B Rs, @ERd	B	Rs8 → @ERd			2							—	—	↕	↕	0
MOV.B Rs, @(d:16, ERd)	B	Rs8 → @(d:16, ERd)				4						—	—	↕	↕	0
MOV.B Rs, @(d:24, ERd)	B	Rs8 → @(d:24, ERd)				8						—	—	↕	↕	0
MOV.B Rs, @–ERd	B	ERd32–1 → ERd32, Rs8 → @ERd					2					—	—	↕	↕	0
MOV.B Rs, @aa:8	B	Rs8 → @aa:8						2				—	—	↕	↕	0
MOV.B Rs, @aa:16	B	Rs8 → @aa:16						4				—	—	↕	↕	0
MOV.B Rs, @aa:24	B	Rs8 → @aa:24						6				—	—	↕	↕	0
MOV.W #xx:16, Rd	W	#xx:16 → Rd16	4									—	—	↕	↕	0
MOV.W Rs, Rd	W	Rs16 → Rd16		2								—	—	↕	↕	0
MOV.W @ERs, Rd	W	@ERs → Rd16			2							—	—	↕	↕	0
MOV.W @(d:16, ERs), Rd	W	@(d:16, ERs) → Rd16				4						—	—	↕	↕	0
MOV.W @(d:24, ERs), Rd	W	@(d:24, ERs) → Rd16				8						—	—	↕	↕	0
MOV.W @ERs+, Rd	W	@ERs → Rd16, ERs32+2 → @ERd32					2					—	—	↕	↕	0
MOV.W @aa:16, Rd	W	@aa:16 → Rd16						4				—	—	↕	↕	0
MOV.W @aa:24, Rd	W	@aa:24 → Rd16						6				—	—	↕	↕	0
MOV.W Rs, @ERd	W	Rs16 → @ERd			2							—	—	↕	↕	0
MOV.W Rs, @(d:16, ERd)	W	Rs16 → @(d:16, ERd)				4						—	—	↕	↕	0
MOV.W Rs, @(d:24, ERd)	W	Rs16 → @(d:24, ERd)				8						—	—	↕	↕	0

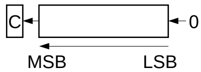
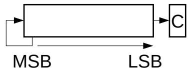
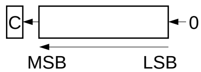
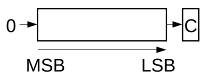
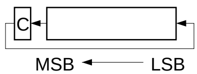
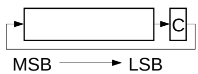
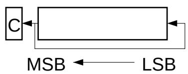
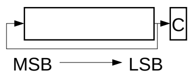
MOV.W Rs, @aa:16	W	Rs16 → @aa:16						4				—	—	↕	↕	0
MOV.W Rs, @aa:24	W	Rs16 → @aa:24						6				—	—	↕	↕	0
MOV.L #xx:32, Rd	L	#xx:32 → Rd32	6									—	—	↕	↕	0
MOV.L ERs, ERd	L	ERs32 → ERd32		2								—	—	↕	↕	0
MOV.L @ERs, ERd	L	@ERs → ERd32			4							—	—	↕	↕	0
MOV.L @(d:16, ERs), ERd	L	@(d:16, ERs) → ERd32				6						—	—	↕	↕	0
MOV.L @(d:24, ERs), ERd	L	@(d:24, ERs) → ERd32				10						—	—	↕	↕	0
MOV.L @ERs+, ERd	L	@ERs → ERd32, ERs32+4 → ERs32					4					—	—	↕	↕	0
MOV.L @aa:16, ERd	L	@aa:16 → ERd32						6				—	—	↕	↕	0
MOV.L @aa:24, ERd	L	@aa:24 → ERd32						8				—	—	↕	↕	0
MOV.L ERs, @ERd	L	ERs32 → @ERd			4							—	—	↕	↕	0
MOV.L ERs, @(d:16, ERd)	L	ERs32 → @(d:16, ERd)				6						—	—	↕	↕	0
MOV.L ERs, @(d:24, ERd)	L	ERs32 → @(d:24, ERd)				10						—	—	↕	↕	0
MOV.L ERs, @–ERd	L	ERd32–4 → ERd32, ERs32 → @ERd					4					—	—	↕	↕	0
MOV.L ERs, @aa:16	L	ERs32 → @aa:16						6				—	—	↕	↕	0
MOV.L ERs, @aa:24	L	ERs32 → @aa:24						8				—	—	↕	↕	0
POP.W Rn	W	@SP → Rn16, SP+2 → SP									2	—	—	↕	↕	0
POP.L ERn	L	@SP → ERn32, SP+4 → SP									4	—	—	↕	↕	0
PUSH.W Rn	W	SP–2 → SP, Rn16 → @SP									2	—	—	↕	↕	0
PUSH.L ERn	L	SP–4 → SP, ERn32 → @SP									4	—	—	↕	↕	0
MOVFPE @aa:16, Rd	B	Cannot be used in the H8/3048B Group						4				Cannot be used in the H8/3048B Gro				
MOVTPE Rs, @aa:16	B	Cannot be used in the H8/3048B Group						4				Cannot be used in the H8/3048B Gro				

ADD.B #xx:8, Rd	B	Rd8+#xx:8 → Rd8	2									—	↕	↕	↕	
ADD.B Rs, Rd	B	Rd8+Rs8 → Rd8		2								—	↕	↕	↕	
ADD.W #xx:16, Rd	W	Rd16+#xx:16 → Rd16	4									—	(1)	↕	↕	
ADD.W Rs, Rd	W	Rd16+Rs16 → Rd16		2								—	(1)	↕	↕	
ADD.L #xx:32, ERd	L	ERd32+#xx:32 → ERd32	6									—	(2)	↕	↕	
ADD.L ERs, ERd	L	ERd32+ERs32 → ERd32		2								—	(2)	↕	↕	
ADDX.B #xx:8, Rd	B	Rd8+#xx:8 +C → Rd8	2									—	↕	↕	(3)	
ADDX.B Rs, Rd	B	Rd8+Rs8 +C → Rd8		2								—	↕	↕	(3)	
ADDS.L #1, ERd	L	ERd32+1 → ERd32		2								—	—	—	—	
ADDS.L #2, ERd	L	ERd32+2 → ERd32		2								—	—	—	—	
ADDS.L #4, ERd	L	ERd32+4 → ERd32		2								—	—	—	—	
INC.B Rd	B	Rd8+1 → Rd8		2								—	—	↕	↕	
INC.W #1, Rd	W	Rd16+1 → Rd16		2								—	—	↕	↕	
INC.W #2, Rd	W	Rd16+2 → Rd16		2								—	—	↕	↕	
INC.L #1, ERd	L	ERd32+1 → ERd32		2								—	—	↕	↕	
INC.L #2, ERd	L	ERd32+2 → ERd32		2								—	—	↕	↕	
DAA Rd	B	Rd8 decimal adjust → Rd8		2								—	*	↕	↕	
SUB.B Rs, Rd	B	Rd8–Rs8 → Rd8		2								—	↕	↕	↕	
SUB.W #xx:16, Rd	W	Rd16–#xx:16 → Rd16	4									—	(1)	↕	↕	
SUB.W Rs, Rd	W	Rd16–Rs16 → Rd16		2								—	(1)	↕	↕	
SUB.L #xx:32, ERd	L	ERd32–#xx:32 → ERd32	6									—	(2)	↕	↕	
SUB.L ERs, ERd	L	ERd32–ERs32 → ERd32		2								—	(2)	↕	↕	
SUBX.B #xx:8, Rd	B	Rd8–#xx:8–C → Rd8	2									—	↕	↕	(3)	
SUBX.B Rs, Rd	B	Rd8–Rs8–C → Rd8		2								—	↕	↕	(3)	
SUBS.L #1, ERd	L	ERd32–1 → ERd32		2								—	—	—	—	
SUBS.L #2, ERd	L	ERd32–2 → ERd32		2								—	—	—	—	
SUBS.L #4, ERd	L	ERd32–4 → ERd32		2								—	—	—	—	
DEC.B Rd	B	Rd8–1 → Rd8		2								—	—	↕	↕	
DEC.W #1, Rd	W	Rd16–1 → Rd16		2								—	—	↕	↕	
DEC.W #2, Rd	W	Rd16–2 → Rd16		2								—	—	↕	↕	

DAS.Rd	B	Rd8 decimal adjust → Rd8		2								—	*	↕	↕	*
MULXU. B Rs, Rd	B	Rd8 × Rs8 → Rd16 (unsigned multiplication)		2								—	—	—	—	—
MULXU. W Rs, ERd	W	Rd16 × Rs16 → ERd32 (unsigned multiplication)		2								—	—	—	—	—
MULXS. B Rs, Rd	B	Rd8 × Rs8 → Rd16 (signed multiplication)		4								—	—	↕	↕	—
MULXS. W Rs, ERd	W	Rd16 × Rs16 → ERd32 (signed multiplication)		4								—	—	↕	↕	—
DIVXU. B Rs, Rd	B	Rd16 ÷ Rs8 → Rd16 (RdH: remainder, RdL: quotient) (unsigned division)		2								—	—	(6)	(7)	—
DIVXU. W Rs, ERd	W	ERd32 ÷ Rs16 → ERd32 (Ed: remainder, Rd: quotient) (unsigned division)		2								—	—	(6)	(7)	—
DIVXS. B Rs, Rd	B	Rd16 ÷ Rs8 → Rd16 (RdH: remainder, RdL: quotient) (signed division)		4								—	—	(8)	(7)	—
DIVXS. W Rs, ERd	W	ERd32 ÷ Rs16 → ERd32 (Ed: remainder, Rd: quotient) (signed division)		4								—	—	(8)	(7)	—
CMP.B #xx:8, Rd	B	Rd8–#xx:8	2									—	↕	↕	↕	↕
CMP.B Rs, Rd	B	Rd8–Rs8		2								—	↕	↕	↕	↕
CMP.W #xx:16, Rd	W	Rd16–#xx:16	4									—	(1)	↕	↕	↕
CMP.W Rs, Rd	W	Rd16–Rs16		2								—	(1)	↕	↕	↕
CMP.L #xx:32, ERd	L	ERd32–#xx:32	6									—	(2)	↕	↕	↕
CMP.L ERs, ERd	L	ERd32–ERs32		2								—	(2)	↕	↕	↕

NEG.L ERd	L	0–ERd32 → ERd32		2								—	↕	↕	↕	↕		
EXTU.W Rd	W	0 → (<bits 15 to 8> of Rd16)		2								—	—	0	↕	0		
EXTU.L ERd	L	0 → (<bits 31 to 16> of ERd32)		2								—	—	0	↕	0		
EXTS.W Rd	W	(<bit 7> of Rd16) → (<bits 15 to 8> of Rd16)		2								—	—	↕	↕	0		
EXTS.L ERd	L	(<bit 15> of ERd32) → (<bits 31 to 16> of ERd32)		2								—	—	↕	↕	0		

AND.B #xx:8, Rd	B	Rd8∧#xx:8 → Rd8	2								—	—	↕	↕	0
AND.B Rs, Rd	B	Rd8∧Rs8 → Rd8		2							—	—	↕	↕	0
AND.W #xx:16, Rd	W	Rd16∧#xx:16 → Rd16	4								—	—	↕	↕	0
AND.W Rs, Rd	W	Rd16∧Rs16 → Rd16		2							—	—	↕	↕	0
AND.L #xx:32, ERd	L	ERd32∧#xx:32 → ERd32	6								—	—	↕	↕	0
AND.L ERs, ERd	L	ERd32∧ERs32 → ERd32		4							—	—	↕	↕	0
OR.B #xx:8, Rd	B	Rd8∨#xx:8 → Rd8	2								—	—	↕	↕	0
OR.B Rs, Rd	B	Rd8∨Rs8 → Rd8		2							—	—	↕	↕	0
OR.W #xx:16, Rd	W	Rd16∨#xx:16 → Rd16	4								—	—	↕	↕	0
OR.W Rs, Rd	W	Rd16∨Rs16 → Rd16		2							—	—	↕	↕	0
OR.L #xx:32, ERd	L	ERd32∨#xx:32 → ERd32	6								—	—	↕	↕	0
OR.L ERs, ERd	L	ERd32∨ERs32 → ERd32		4							—	—	↕	↕	0
XOR.B #xx:8, Rd	B	Rd8⊕#xx:8 → Rd8	2								—	—	↕	↕	0
XOR.B Rs, Rd	B	Rd8⊕Rs8 → Rd8		2							—	—	↕	↕	0
XOR.W #xx:16, Rd	W	Rd16⊕#xx:16 → Rd16	4								—	—	↕	↕	0
XOR.W Rs, Rd	W	Rd16⊕Rs16 → Rd16		2							—	—	↕	↕	0
XOR.L #xx:32, ERd	L	ERd32⊕#xx:32 → ERd32	6								—	—	↕	↕	0
XOR.L ERs, ERd	L	ERd32⊕ERs32 → ERd32		4							—	—	↕	↕	0
NOT.B Rd	B	¬ Rd8 → Rd8		2							—	—	↕	↕	0
NOT.W Rd	W	¬ Rd16 → Rd16		2							—	—	↕	↕	0
NOT.L ERd	L	¬ Rd32 → Rd32		2							—	—	↕	↕	0

SHAL.B Rd	B			2								—	—	↕	↕	↕
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, Rd	B	(#xx:3 of Rd8) ← 1		2								—	—	—	—	—
BSET #xx:3, @ERd	B	(#xx:3 of @ERd) ← 1			4							—	—	—	—	—
BSET #xx:3, @aa:8	B	(#xx:3 of @aa:8) ← 1						4				—	—	—	—	—
BSET Rn, Rd	B	(Rn8 of Rd8) ← 1		2								—	—	—	—	—
BSET Rn, @ERd	B	(Rn8 of @ERd) ← 1			4							—	—	—	—	—
BSET Rn, @aa:8	B	(Rn8 of @aa:8) ← 1						4				—	—	—	—	—
BCLR #xx:3, Rd	B	(#xx:3 of Rd8) ← 0		2								—	—	—	—	—
BCLR #xx:3, @ERd	B	(#xx:3 of @ERd) ← 0			4							—	—	—	—	—
BCLR #xx:3, @aa:8	B	(#xx:3 of @aa:8) ← 0						4				—	—	—	—	—
BCLR Rn, Rd	B	(Rn8 of Rd8) ← 0		2								—	—	—	—	—
BCLR Rn, @ERd	B	(Rn8 of @ERd) ← 0			4							—	—	—	—	—
BCLR Rn, @aa:8	B	(Rn8 of @aa:8) ← 0						4				—	—	—	—	—
BNOT #xx:3, Rd	B	(#xx:3 of Rd8) ← ¬ (#xx:3 of Rd8)		2								—	—	—	—	—
BNOT #xx:3, @ERd	B	(#xx:3 of @ERd) ← ¬ (#xx:3 of @ERd)			4							—	—	—	—	—
BNOT #xx:3, @aa:8	B	(#xx:3 of @aa:8) ← ¬ (#xx:3 of @aa:8)						4				—	—	—	—	—
BNOT Rn, Rd	B	(Rn8 of Rd8) ← ¬ (Rn8 of Rd8)		2								—	—	—	—	—
BNOT Rn, @ERd	B	(Rn8 of @ERd) ← ¬ (Rn8 of @ERd)			4							—	—	—	—	—
BNOT Rn, @aa:8	B	(Rn8 of @aa:8) ← ¬ (Rn8 of @aa:8)						4				—	—	—	—	—
BTST #xx:3, Rd	B	¬ (#xx:3 of Rd8) → Z		2								—	—	—	↕	—
BTST #xx:3, @ERd	B	¬ (#xx:3 of @ERd) → Z			4							—	—	—	↕	—
BTST #xx:3, @aa:8	B	¬ (#xx:3 of @aa:8) → Z						4				—	—	—	↕	—
BTST Rn, Rd	B	¬ (Rn8 of @Rd8) → Z		2								—	—	—	↕	—
BTST Rn, @ERd	B	¬ (Rn8 of @ERd) → Z			4							—	—	—	↕	—
BTST Rn, @aa:8	B	¬ (Rn8 of @aa:8) → Z						4				—	—	—	↕	—
BLD #xx:3, Rd	B	(#xx:3 of Rd8) → C		2								—	—	—	—	—

BILD #xx:3, Rd	B	¬ (#xx:3 of Rd8) → C		2								—	—	—	—	—
BILD #xx:3, @ERd	B	¬ (#xx:3 of @ERd) → C			4							—	—	—	—	—
BILD #xx:3, @aa:8	B	¬ (#xx:3 of @aa:8) → C						4				—	—	—	—	—
BST #xx:3, Rd	B	C → (#xx:3 of Rd8)		2								—	—	—	—	—
BST #xx:3, @ERd	B	C → (#xx:3 of @ERd24)			4							—	—	—	—	—
BST #xx:3, @aa:8	B	C → (#xx:3 of @aa:8)						4				—	—	—	—	—
BIST #xx:3, Rd	B	¬ C → (#xx:3 of Rd8)		2								—	—	—	—	—
BIST #xx:3, @ERd	B	¬ C → (#xx:3 of @ERd24)			4							—	—	—	—	—
BIST #xx:3, @aa:8	B	¬ C → (#xx:3 of @aa:8)						4				—	—	—	—	—
BAND #xx:3, Rd	B	C∧(#xx:3 of Rd8) → C		2								—	—	—	—	—
BAND #xx:3, @ERd	B	C∧(#xx:3 of @ERd24) → C			4							—	—	—	—	—
BAND #xx:3, @aa:8	B	C∧(#xx:3 of @aa:8) → C						4				—	—	—	—	—
BIAND #xx:3, Rd	B	C∧ ¬ (#xx:3 of Rd8) → C		2								—	—	—	—	—
BIAND #xx:3, @ERd	B	C∧ ¬ (#xx:3 of @ERd24) → C			4							—	—	—	—	—
BIAND #xx:3, @aa:8	B	C∧ ¬ (#xx:3 of @aa:8) → C						4				—	—	—	—	—
BOR #xx:3, Rd	B	C∨(#xx:3 of Rd8) → C		2								—	—	—	—	—
BOR #xx:3, @ERd	B	C∨(#xx:3 of @ERd24) → C			4							—	—	—	—	—
BOR #xx:3, @aa:8	B	C∨(#xx:3 of @aa:8) → C						4				—	—	—	—	—
BIOR #xx:3, Rd	B	C∨ ¬ (#xx:3 of Rd8) → C		2								—	—	—	—	—
BIOR #xx:3, @ERd	B	C∨ ¬ (#xx:3 of @ERd24) → C			4							—	—	—	—	—
BIOR #xx:3, @aa:8	B	C∨ ¬ (#xx:3 of @aa:8) → C						4				—	—	—	—	—
BXOR #xx:3, Rd	B	C⊕(#xx:3 of Rd8) → C		2								—	—	—	—	—
BXOR #xx:3, @ERd	B	C⊕(#xx:3 of @ERd24) → C			4							—	—	—	—	—
BXOR #xx:3, @aa:8	B	C⊕(#xx:3 of @aa:8) → C						4				—	—	—	—	—
BIXOR #xx:3, Rd	B	C⊕ ¬ (#xx:3 of Rd8) → C		2								—	—	—	—	—
BIXOR #xx:3, @ERd	B	C⊕ ¬ (#xx:3 of @ERd24) → C			4							—	—	—	—	—
BIXOR #xx:3, @aa:8	B	C⊕ ¬ (#xx:3 of @aa:8) → C						4				—	—	—	—	—

BRA d:8 (BT d:8)	—	If condition is true then PC ← PC+d else next;	Always							2			—	—	—	—	—
BRA d:16 (BT d:16)	—									4			—	—	—	—	—
BRN d:8 (BF d:8)	—		Never							2			—	—	—	—	—
BRN d:16 (BF d:16)	—									4			—	—	—	—	—
BHI d:8	—		C ∨ Z = 0							2			—	—	—	—	—
BHI d:16	—									4			—	—	—	—	—
BLS d:8	—		C ∨ Z = 1							2			—	—	—	—	—
BLS d:16	—									4			—	—	—	—	—
BCC d:8 (BHS d:8)	—		C = 0							2			—	—	—	—	—
BCC d:16 (BHS d:16)	—									4			—	—	—	—	—
BCS d:8 (BLO d:8)	—		C = 1							2			—	—	—	—	—
BCS d:16 (BLO d:16)	—									4			—	—	—	—	—
BNE d:8	—		Z = 0							2			—	—	—	—	—
BNE d:16	—									4			—	—	—	—	—
BEQ d:8	—		Z = 1							2			—	—	—	—	—
BEQ d:16	—									4			—	—	—	—	—
BVC d:8	—		V = 0							2			—	—	—	—	—
BVC d:16	—									4			—	—	—	—	—
BVS d:8	—		V = 1							2			—	—	—	—	—
BVS d:16	—									4			—	—	—	—	—
BPL d:8	—		N = 0							2			—	—	—	—	—
BPL d:16	—									4			—	—	—	—	—
BMI d:8	—		N = 1							2			—	—	—	—	—
BMI d:16	—									4			—	—	—	—	—
BGE d:8	—		N⊕V = 0							2			—	—	—	—	—
BGE d:16	—									4			—	—	—	—	—
BLT d:8	—		N⊕V = 1							2			—	—	—	—	—
BLT d:16	—									4			—	—	—	—	—
BGT d:8	—		Z ∨ (N⊕V) = 0							2			—	—	—	—	—
BGT d:16	—									4			—	—	—	—	—
BLE d:8	—		Z ∨ (N⊕V) = 1							2			—	—	—	—	—
BLE d:16	—									4			—	—	—	—	—

JMP @@aa:8	—	PC ← @aa:8								2		—	—	—	—		
BSR d:8	—	PC → @–SP PC ← PC+d:8							2			—	—	—	—		
BSR d:16	—	PC → @–SP PC ← PC+d:16							4			—	—	—	—		
JSR @ERn	—	PC → @–SP PC ← @ERn			2							—	—	—	—		
JSR @aa:24	—	PC → @–SP PC ← @aa:24						4				—	—	—	—		
JSR @@aa:8	—	PC → @–SP PC ← @aa:8								2		—	—	—	—		
RTS	—	PC ← @SP+									2	—	—	—	—		

TRAPA #x:2	—	PC → @–SP, CCR → @–SP, <vector> → PC								2	1	—	—	—	—
RTE	—	CCR ← @SP+, PC ← @SP+									↕	↕	↕	↕	↕
SLEEP	—	Transition to power-down state									—	—	—	—	—
LDC #xx:8, CCR	B	#xx:8 → CCR	2								↕	↕	↕	↕	↕
LDC Rs, CCR	B	Rs8 → CCR		2							↕	↕	↕	↕	↕
LDC @ERs, CCR	W	@ERs → CCR			4						↕	↕	↕	↕	↕
LDC @(d:16, ERs), CCR	W	@(d:16, ERs) → CCR				6					↕	↕	↕	↕	↕
LDC @(d:24, ERs), CCR	W	@(d:24, ERs) → CCR				10					↕	↕	↕	↕	↕
LDC @ERs+, CCR	W	@ERs → CCR, ERs32+2 → ERs32					4				↕	↕	↕	↕	↕
LDC @aa:16, CCR	W	@aa:16 → CCR						6			↕	↕	↕	↕	↕
LDC @aa:24, CCR	W	@aa:24 → CCR						8			↕	↕	↕	↕	↕
STC CCR, Rd	B	CCR → Rd8		2							—	—	—	—	—
STC CCR, @ERd	W	CCR → @ERd			4						—	—	—	—	—
STC CCR, @(d:16, ERd)	W	CCR → @(d:16, ERd)				6					—	—	—	—	—
STC CCR, @(d:24, ERd)	W	CCR → @(d:24, ERd)				10					—	—	—	—	—
STC CCR, @–ERd	W	ERd32–2 → ERd32, CCR → @ERd					4				—	—	—	—	—
STC CCR, @aa:16	W	CCR → @aa:16						6			—	—	—	—	—
STC CCR, @aa:24	W	CCR → @aa:24						8			—	—	—	—	—
ANDC #xx:8, CCR	B	CCR∧#xx:8 → CCR	2								↕	↕	↕	↕	↕
ORC #xx:8, CCR	B	CCR∨#xx:8 → CCR	2								↕	↕	↕	↕	↕
XORC #xx:8, CCR	B	CCR⊕#xx:8 → CCR	2								↕	↕	↕	↕	↕
NOP	—	PC ← PC+2								2	—	—	—	—	—

EEPMOV. B	—	if R4L ≠ 0 then repeat @R5 → @R6 R5+1 → R5 R6+1 → R6 R4L–1 → R4L until R4L=0 else next									4	—	—	—	—	
EEPMOV. W	—	if R4 ≠ 0 then repeat @R5 → @R6 R5+1 → R5 R6+1 → R6 R4–1 → R4 until R4=0 else next									4	—	—	—	—	

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 case
section A.3, Number of States Required for Execution.

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	BEQ	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	T	
8	ADD													
9	ADDX													
A	CMP													
B	SUBX													
C	OR													
D	XOR													
E	AND													

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
01	MOV				LDC/STC				SLEEP			Tr
0A	INC											ADD
0B	ADDS					INC		INC		ADDS		
0F	DAA											MOV
10	SHLL	SHLL		SHLL						SHAL		SHAL
11	SHLR	SHLR		SHLR						SHAR		SHAR
12	ROTXL	ROTXL		ROTXL						ROTL		ROTL
13	ROTXR	ROTXR		ROTXR						ROTR		ROTR
17	NOT	NOT		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
7C	MOV	ADD	CMP	SUB	OR	XOR	AND					

RENESAS

Instruction code:

1st byte		2nd byte		3rd byte		4th byte	
AH	AL	BH	BL	CH	CL	DH	DL

Instruction when most

Instruction when most

CL / AH / ALBH / BLCH	0	1	2	3	4	5	6	7	8	9	A	B
01406										LDC / STC		LDC / STC
01C05	MULXS		MULXS									
01D05		DIVXS		DIVXS								
01F06					OR	XOR	AND					
7Cr06 *1				BTST								
7Cr07 *1				BTST	BOR / BIOR	BXOR / BIXOR	BAND / BIAND	BLD / BILD				
7Dr06 *1	BSET	BNOT	BCLR					BST / BIST				
7Dr07 *1	BSET	BNOT	BCLR									
7Eaa6 *2				BTST								
7Eaa7 *2				BTST	BOR / BIOR	BXOR / BIXOR	BAND / BIAND	BLD / BILD				
7Faa6 *2	BSET	BNOT	BCLR					BST / BIST				
7Faa7 *2	BSET	BNOT	BCLR									

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 = $2 \times 4 + 2 \times 4 + 2 \times 4 = 24$

Instruction fetch	S_I	2	3	3	4	3 + 2m	2
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	1	1	1	1	1

Legend:

m: Number of wait states inserted into external device access

	ADD.L #xx:32, ERd	3		
	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		

	BRN d:16 (BF d:16)	2		
	BHI d:16	2		
	BLS d:16	2		
	BCC d:16 (BHS d:16)	2		
	BCS d:16 (BLO d:16)	2		
	BNE d:16	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

	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
BNOT	BNOT #xx:3, Rd		1		
	BNOT #xx:3, @ERd		2		2
	BNOT #xx:3, @aa:8		2		2
	BNOT Rn, Rd		1		
	BNOT Rn, @ERd		2		2
	BNOT Rn, @aa:8		2		2
BOR	BOR #xx:3, Rd		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*1	2	1	
		Advanced	2	2	
	BSR d:16	Normal*1	2	1	
		Advanced	2	2	

	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		
DEC	DEC.B Rd	1		
	DEC.W #1/2, Rd	1		
	DEC.L #1/2, ERd	1		
DIVXS	DIVXS.B Rs, Rd	2		
	DIVXS.W Rs, ERd	2		
DIVXU	DIVXU.B Rs, Rd	1		
	DIVXU.W Rs, ERd	1		
EEPMOV	EEPMOV.B	2		2n + 2[*2]
	EEPMOV.W	2		2n + 2[*2]
EXTS	EXTS.W Rd	1		
	EXTS.L ERd	1		
EXTU	EXTU.W Rd	1		
	EXTU.L ERd	1		

	JMP @@aa:8	Normal	2	1				
		Advanced	2	2				
JSR	JSR @ERn	Normal[*1]	2		1			
		Advanced	2		2			
	JSR @aa:24	Normal[*1]	2		1			
		Advanced	2		2			
	JSR @@aa:8	Normal[*1]	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	MOV.B #xx:8, Rd		1					
	MOV.B Rs, Rd		1					
	MOV.B @ERs, Rd		1			1		
	MOV.B @(d:16, ERs), Rd		2			1		
	MOV.B @(d:24, ERs), Rd		4			1		
	MOV.B @ERs+, Rd		1			1		
	MOV.B @aa:8, Rd		1			1		
	MOV.B @aa:16, Rd		2			1		
	MOV.B @aa:24, Rd		3			1		
	MOV.B Rs, @ERd		1			1		
	MOV.B Rs, @(d:16, ERd)		2			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.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		

		Advanced	2	2	
SHAL	SHAL.B Rd		1		
	SHAL.W Rd		1		
	SHAL.L ERd		1		
SHAR	SHAR.B Rd		1		
	SHAR.W Rd		1		
	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		

	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. Not available in the H8/3048B Group.

2. n is the value set in register R4L or R4. The source and destination are acc
times each.

		H8/3044 Mask ROM Version			F-ZTAT Version
H'FF40	—	—	FLMCR	FLMCR1	—
H'FF41	—	—	—	FLMCR2	—
H'FF42	—	—	EBR1	EBR	—
H'FF43	—	—	EBR2	—	—
H'FF47	—	—	—	RAMCR	—
H'FF48	—	—	RAMCR	—	—

Note: A dash (“—”) indicates that access is prohibited. Normal operation is not guarant
these addresses are accessed.

H'20	MAR0AR	8								
H'21	MAR0AE	8								
H'22	MAR0AH	8								
H'23	MAR0AL	8								
H'24	ETCR0AH	8								
H'25	ETCR0AL	8								
H'26	IOAR0A	8								
H'27	DTCR0A	8	DTE	DTSZ	DTID	RPE	DTIE	DTS2	DTS1	DTS0
			DTE	DTSZ	SAID	SAIDE	DTIE	DTS2A	DTS1A	DTS0A
H'28	MAR0BR	8								
H'29	MAR0BE	8								
H'2A	MAR0BH	8								
H'2B	MAR0BL	8								
H'2C	ETCR0BH	8								
H'2D	ETCR0BL	8								
H'2E	IOAR0B	8								
H'2F	DTCR0B	8	DTE	DTSZ	DTID	RPE	DTIE	DTS2	DTS1	DTS0
			DTME	—	DAID	DAIDE	TMS	DTS2B	DTS1B	DTS0B

H'37	DTCR1A	8	DTE	DTSZ	DTID	RPE	DTIE	DTS2	DTS1	DTS0
			DTE	DTSZ	SAID	SAIDE	DTIE	DTS2A	DTS1A	DTS0A
H'38	MAR1BR	8								
H'39	MAR1BE	8								
H'3A	MAR1BH	8								
H'3B	MAR1BL	8								
H'3C	ETCR1BH	8								
H'3D	ETCR1BL	8								
H'3E	IOAR1B	8								
H'3F	DTCR1B	8	DTE	DTSZ	DTID	RPE	DTIE	DTS2	DTS1	DTS0
			DTME	—	DAID	DAIDE	TMS	DTS2B	DTS1B	DTS0B
H'40	FLMCR1*4	8	FWE	SWE	ESU	PSU	EV	PV	E	P
H'41	FLMCR2*4	8	FLER	—*3	—*3	—*3	—*3	—*3	—*3	—*3
H'42	EBR*4	8	EB7	EB6	EB5	EB4	EB3	EB2	EB1	EB0
H'43	Reserved area (access prohibited)									
H'44										
H'45										
H'46										
H'47	RAMCR*4	8	—	—	—	—	RAMS	RAM2	RAM1	—
H'48	Reserved area (access prohibited)									
H'49										
H'4A										
H'4B										
H'4C										
H'4D										
H'4E										
H'4F										

H'56										
H'57										
H'58										
H'59										
H'5A										
H'5B										
H'5C	DASTCR	8	—	—	—	—	—	—	—	DASTE
H'5D	DIVCR	8	—	—	—	—	—	—	DIV1	DIV0
H'5E	MSTCR	8	PSTOP	—	MSTOP5	MSTOP4	MSTOP3	MSTOP2	MSTOP1	MSTOF
H'5F	CSCR	8	CS7E	CS6E	CS5E	CS4E	—	—	—	—
H'60	TSTR	8	—	—	—	STR4	STR3	STR2	STR1	STR0
H'61	TSNC	8	—	—	—	SYNC4	SYNC3	SYNC2	SYNC1	SYNC0
H'62	TMDR	8	—	MDF	FDIR	PWM4	PWM3	PWM2	PWM1	PWM0
H'63	TFCR	8	—	—	CMD1	CMD0	BFB4	BFA4	BFB3	BFA3
H'64	TCR0	8	—	CCLR1	CCLR0	CKEG1	CKEG0	TPSC2	TPSC1	TPSC0
H'65	TIOR0	8	—	IOB2	IOB1	IOB0	—	IOA2	IOA1	IOA0
H'66	TIER0	8	—	—	—	—	—	OVIE	IMIEB	IMIEA
H'67	TSR0	8	—	—	—	—	—	OVF	IMFB	IMFA
H'68	TCNT0H	16								
H'69	TCNT0L									
H'6A	GRA0H	16								
H'6B	GRA0L									
H'6C	GRB0H	16								
H'6D	GRB0L									
H'6E	TCR1	8	—	CCLR1	CCLR0	CKEG1	CKEG0	TPSC2	TPSC1	TPSC0
H'6F	TIOR1	8	—	IOB2	IOB1	IOB0	—	IOA2	IOA1	IOA0

H'77	GRB1L									
H'78	TCR2	8	—	CCLR1	CCLR0	CKEG1	CKEG0	TPSC2	TPSC1	TPSC0
H'79	TIOR2	8	—	IOB2	IOB1	IOB0	—	IOA2	IOA1	IOA0
H'7A	TIER2	8	—	—	—	—	—	OVIE	IMIEB	IMIEA
H'7B	TSR2	8	—	—	—	—	—	OVF	IMFB	IMFA
H'7C	TCNT2H	16								
H'7D	TCNT2L									
H'7E	GRA2H	16								
H'7F	GRA2L									
H'80	GRB2H	16								
H'81	GRB2L									
H'82	TCR3	8	—	CCLR1	CCLR0	CKEG1	CKEG0	TPSC2	TPSC1	TPSC0
H'83	TIOR3	8	—	IOB2	IOB1	IOB0	—	IOA2	IOA1	IOA0
H'84	TIER3	8	—	—	—	—	—	OVIE	IMIEB	IMIEA
H'85	TSR3	8	—	—	—	—	—	OVF	IMFB	IMFA
H'86	TCNT3H	16								
H'87	TCNT3L									
H'88	GRA3H	16								
H'89	GRA3L									
H'8A	GRB3H	16								
H'8B	GRB3L									
H'8C	BRA3H	16								
H'8D	BRA3L									
H'8E	BRB3H	16								
H'8F	BRB3L									

H'96	TCNT4H	16								
H'97	TCNT4L									
H'98	GRA4H	16								
H'99	GRA4L									
H'9A	GRB4H	16								
H'9B	GRB4L									
H'9C	BRA4H	16								
H'9D	BRA4L									
H'9E	BRB4H	16								
H'9F	BRB4L									
H'A0	TPMR	8	—	—	—	—	G3NOV	G2NOV	G1NOV	G0NOV
H'A1	TPCR	8	G3CMS1	G3CMS0	G2CMS1	G2CMS0	G1CMS1	G1CMS0	G0CMS1	G0CMS
H'A2	NDERB	8	NDER15	NDER14	NDER13	NDER12	NDER11	NDER10	NDER9	NDER8
H'A3	NDERA	8	NDER7	NDER6	NDER5	NDER4	NDER3	NDER2	NDER1	NDER0
H'A4	NDRB*1	8	NDR15	NDR14	NDR13	NDR12	NDR11	NDR10	NDR9	NDR8
		8	NDR15	NDR14	NDR13	NDR12	—	—	—	—
H'A5	NDRA*1	8	NDR7	NDR6	NDR5	NDR4	NDR3	NDR2	NDR1	NDR0
		8	NDR7	NDR6	NDR5	NDR4	—	—	—	—
H'A6	NDRB*1	8	—	—	—	—	—	—	—	—
		8	—	—	—	—	NDR11	NDR10	NDR9	NDR8
H'A7	NDRA*1	8	—	—	—	—	—	—	—	—
		8	—	—	—	—	NDR3	NDR2	NDR1	NDR0
H'A8	TCSR*2	8	OVF	WT/$\overline{IT}$	TME	—	—	CKS2	CKS1	CKS0
H'A9	TCNT*2	8								
H'AA	—		—	—	—	—	—	—	—	—
H'AB	RSTCSR*2	8	WRST	—	—	—	—	—	—	—
H'AC	RFSHCR	8	SRFMD	PSRAME	DRAME	CAS/$\overline{WE}$	M9/$\overline{M8}$	RFSHE	—	RCYCE
H'AD	RTMCSR	8	CMF	CMIE	CKS2	CKS1	CKS0	—	—	—
H'AE	RTCNT	8								
H'AF	RTCOR	8								

H'B6	SCMR	8	—	—	—	—	SDIR	SINV	—	SMIF
H'B7	Reserved area (access prohibited)									
H'B8	SMR	8	C/$\overline{A}$	CHR	PE	O/$\overline{E}$	STOP	MP	CKS1	CKS0
H'B9	BRR	8								
H'BA	SCR	8	TIE	RIE	TE	RE	MPIE	TEIE	CKE1	CKE0
H'BB	TDR	8								
H'BC	SSR	8	TDRE	RDRF	ORER	FER	PER	TEND	MPB	MPBT
H'BD	RDR	8								
H'BE	Reserved area (access prohibited)									
H'BF										
H'C0	P1DDR	8	$P1_7$DDR	$P1_6$DDR	$P1_5$DDR	$P1_4$DDR	$P1_3$DDR	$P1_2$DDR	$P1_1$DDR	$P1_0$DDR
H'C1	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'C2	P1DR	8	$P1_7$	$P1_6$	$P1_5$	$P1_4$	$P1_3$	$P1_2$	$P1_1$	$P1_0$
H'C3	P2DR	8	$P2_7$	$P2_6$	$P2_5$	$P2_4$	$P2_3$	$P2_2$	$P2_1$	$P2_0$
H'C4	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'C5	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'C6	P3DR	8	$P3_7$	$P3_6$	$P3_5$	$P3_4$	$P3_3$	$P3_2$	$P3_1$	$P3_0$
H'C7	P4DR	8	$P4_7$	$P4_6$	$P4_5$	$P4_4$	$P4_3$	$P4_2$	$P4_1$	$P4_0$
H'C8	P5DDR	8	—	—	—	—	$P5_3$DDR	$P5_2$DDR	$P5_1$DDR	$P5_0$DDR
H'C9	P6DDR	8	—	$P6_6$DDR	$P6_5$DDR	$P6_4$DDR	$P6_3$DDR	$P6_2$DDR	$P6_1$DDR	$P6_0$DDR
H'CA	P5DR	8	—	—	—	—	$P5_3$	$P5_2$	$P5_1$	$P5_0$
H'CB	P6DR	8	—	$P6_6$	$P6_5$	$P6_4$	$P6_3$	$P6_2$	$P6_1$	$P6_0$
H'CC	—		—	—	—	—	—	—	—	—
H'CD	P8DDR	8	—	—	—	$P8_4$DDR	$P8_3$DDR	$P8_2$DDR	$P8_1$DDR	$P8_0$DDR
H'CE	P7DR	8	$P7_7$	$P7_6$	$P7_5$	$P7_4$	$P7_3$	$P7_2$	$P7_1$	$P7_0$
H'CF	P8DR	8	—	—	—	$P8_4$	$P8_3$	$P8_2$	$P8_1$	$P8_0$

H'D6	PBDR	8	PB_7	PB_6	PB_5	PB_4	PB_3	PB_2	PB_1	PB_0
H'D7	—	—	—	—	—	—	—	—	—	—
H'D8	P2PCR	8	$P2_7$PCR	$P2_6$PCR	$P2_5$PCR	$P2_4$PCR	$P2_3$PCR	$P2_2$PCR	$P2_1$PCR	$P2_0$PCR
H'D9	—		—	—	—	—	—	—	—	—
H'DA	P4PCR	8	$P4_7$PCR	$P4_6$PCR	$P4_5$PCR	$P4_4$PCR	$P4_3$PCR	$P4_2$PCR	$P4_1$PCR	$P4_0$PCR
H'DB	P5PCR	8	—	—	—	—	$P5_3$PCR	$P5_2$PCR	$P5_1$PCR	$P5_0$PCR
H'DC	DADR0	8								
H'DD	DADR1	8								
H'DE	DACR	8	DAOE1	DAOE0	DAE	—	—	—	—	—
H'DF	Reserved area (access prohibited)									
H'E0	ADDRAH	8	AD9	AD8	AD7	AD6	AD5	AD4	AD3	AD2
H'E1	ADDRAL	8	AD1	AD0	—	—	—	—	—	—
H'E2	ADDRBH	8	AD9	AD8	AD7	AD6	AD5	AD4	AD3	AD2
H'E3	ADDRBL	8	AD1	AD0	—	—	—	—	—	—
H'E4	ADDRCH	8	AD9	AD8	AD7	AD6	AD5	AD4	AD3	AD2
H'E5	ADDRCL	8	AD1	AD0	—	—	—	—	—	—
H'E6	ADDRDH	8	AD9	AD8	AD7	AD6	AD5	AD4	AD3	AD2
H'E7	ADDRDL	8	AD1	AD0	—	—	—	—	—	—
H'E8	ADCSR	8	ADF	ADIE	ADST	SCAN	CKS	CH2	CH1	CH0
H'E9	ADCR	8	TRGE	—	—	—	—	—	—	—
H'EA	Reserved area (access prohibited)									
H'EB										
H'EC	ABWCR	8	ABW7	ABW6	ABW5	ABW4	ABW3	ABW2	ABW1	ABW0
H'ED	ASTCR	8	AST7	AST6	AST5	AST4	AST3	AST2	AST1	AST0
H'EE	WCR	8	—	—	—	—	WMS1	WMS0	WC1	WC0
H'EF	WCER	8	WCE7	WCE6	WCE5	WCE4	WCE3	WCE2	WCE1	WCE0

H'F6	ISR	8			IRQ5F	IRQ4F	IRQ3F	IRQ2F	IRQ1F	IRQ0F
H'F7	Reserved area (access prohibited)									
H'F8	IPRA	8	IPRA7	IPRA6	IPRA5	IPRA4	IPRA3	IPRA2	IPRA1	IPRA0
H'F9	IPRB	8	IPRB7	IPRB6	IPRB5	—	IPRB3	IPRB2	IPRB1	—
H'FA	Reserved area (access prohibited)									
H'FB										
H'FC										
H'FD										
H'FE										
H'FF										

Legend:

DMAC: DMA controller

ITU: 16-bit integrated timer unit

TPC: Programmable timing pattern controller

WDT: Watchdog timer

SCI: Serial communication interface

Notes: 1. The address depends on the output trigger setting.

2. For write access to TCSR TCNT, and RSTCR see section 12.2.4, Notes on Rewriting.

3. Bits 6 to 0 in FLMCR2 are reserved bits but are readable/writable.

4. Byte data must be used to access FLMCR1, FLMCR2, EBR, and RAMCR. Registers FLMCR1, FLMCR2, EBR, and RAMCR are implemented in the fla version only. The mask ROM version does not have these registers.

H'1E										
H'1F										
H'20	MAR0AR	8								
H'21	MAR0AE	8								
H'22	MAR0AH	8								
H'23	MAR0AL	8								
H'24	ETCR0AH	8								
H'25	ETCR0AL	8								
H'26	IOAR0A	8								
H'27	DTCR0A	8	DTE	DTSZ	DTID	RPE	DTIE	DTS2	DTS1	DTS0
			DTE	DTSZ	SAID	SAIDE	DTIE	DTS2A	DTS1A	DTS0A
H'28	MAR0BR	8								
H'29	MAR0BE	8								
H'2A	MAR0BH	8								
H'2B	MAR0BL	8								
H'2C	ETCR0BH	8								
H'2D	ETCR0BL	8								
H'2E	IOAR0B	8								
H'2F	DTCR0B	8	DTE	DTSZ	DTID	RPE	DTIE	DTS2	DTS1	DTS0
			DTME	—	DAID	DAIDE	TMS	DTS2B	DTS1B	DTS0B

H'36	IOAR1A	8							
H'37	DTCR1A	8	DTE	DTSZ	DTID	RPE	DTIE	DTS2	DTS1	DTS0
			DTE	DTSZ	SAID	SAIDE	DTIE	DTS2A	DTS1A	DTS0A
H'38	MAR1BR	8								
H'39	MAR1BE	8								
H'3A	MAR1BH	8								
H'3B	MAR1BL	8								
H'3C	ETCR1BH	8								
H'3D	ETCR1BL	8								
H'3E	IOAR1B	8								
H'3F	DTCR1B	8	DTE	DTSZ	DTID	RPE	DTIE	DTS2	DTS1	DTS0
			DTME	—	DAID	DAIDE	TMS	DTS2B	DTS1B	DTS0B
H'40	FLMCR	8	V_{PP}	$V_{PP}E$	—	—	EV	PV	E	P
H'41	—		—	—	—	—	—	—	—	—
H'42	EBR1	8	LB7	LB6	LB5	LB4	LB3	LB2	LB1	LB0
H'43	EBR2	8	SB7	SB6	SB5	SB4	SB3	SB2	SB1	SB0
H'44	—		—	—	—	—	—	—	—	—
H'45	—		—	—	—	—	—	—	—	—
H'46	—		—	—	—	—	—	—	—	—
H'47	—		—	—	—	—	—	—	—	—
H'48	RAMCR	8	FLER	—	—	—	RAMS	RAM2	RAM1	RAM0
H'49	—		—	—	—	—	—	—	—	—
H'4A	—		—	—	—	—	—	—	—	—
H'4B	—		—	—	—	—	—	—	—	—
H'4C	—		—	—	—	—	—	—	—	—
H'4D	—		—	—	—	—	—	—	—	—
H'4E	—		—	—	—	—	—	—	—	—
H'4F	—		—	—	—	—	—	—	—	—

H'57	—		—	—	—	—	—	—	—	—
H'58	—		—	—	—	—	—	—	—	—
H'59	—		—	—	—	—	—	—	—	—
H'5A	—		—	—	—	—	—	—	—	—
H'5B	—		—	—	—	—	—	—	—	—
H'5C	DASTCR	8	—	—	—	—	—	—	—	DASTE
H'5D	DIVCR	8	—	—	—	—	—	—	DIV1	DIV0
H'5E	MSTCR	8	PSTOP	—	MSTOP5	MSTOP4	MSTOP3	MSTOP2	MSTOP1	MSTOF
H'5F	CSCR	8	CS7E	CS6E	CS5E	CS4E	—	—	—	—
H'60	TSTR	8	—	—	—	STR4	STR3	STR2	STR1	STR0
H'61	TSNC	8	—	—	—	SYNC4	SYNC3	SYNC2	SYNC1	SYNC0
H'62	TMDR	8	—	MDF	FDIR	PWM4	PWM3	PWM2	PWM1	PWM0
H'63	TFCR	8	—	—	CMD1	CMD0	BFB4	BFA4	BFB3	BFA3
H'64	TCR0	8	—	CCLR1	CCLR0	CKEG1	CKEG0	TPSC2	TPSC1	TPSC0
H'65	TIOR0	8	—	IOB2	IOB1	IOB0	—	IOA2	IOA1	IOA0
H'66	TIER0	8	—	—	—	—	—	OVIE	IMIEB	IMIEA
H'67	TSR0	8	—	—	—	—	—	OVF	IMFB	IMFA
H'68	TCNT0H	16								
H'69	TCNT0L									
H'6A	GRA0H	16								
H'6B	GRA0L									
H'6C	GRB0H	16								
H'6D	GRB0L									
H'6E	TCR1	8	—	CCLR1	CCLR0	CKEG1	CKEG0	TPSC2	TPSC1	TPSC0
H'6F	TIOR1	8	—	IOB2	IOB1	IOB0	—	IOA2	IOA1	IOA0

H'76	GRB1H	16								
H'77	GRB1L									
H'78	TCR2	8	—	CCLR1	CCLR0	CKEG1	CKEG0	TPSC2	TPSC1	TPSC0
H'79	TIOR2	8	—	IOB2	IOB1	IOB0	—	IOA2	IOA1	IOA0
H'7A	TIER2	8	—	—	—	—	—	OVIE	IMIEB	IMIEA
H'7B	TSR2	8	—	—	—	—	—	OVF	IMFB	IMFA
H'7C	TCNT2H	16								
H'7D	TCNT2L									
H'7E	GRA2H	16								
H'7F	GRA2L									
H'80	GRB2H	16								
H'81	GRB2L									
H'82	TCR3	8	—	CCLR1	CCLR0	CKEG1	CKEG0	TPSC2	TPSC1	TPSC0
H'83	TIOR3	8	—	IOB2	IOB1	IOB0	—	IOA2	IOA1	IOA0
H'84	TIER3	8	—	—	—	—	—	OVIE	IMIEB	IMIEA
H'85	TSR3	8	—	—	—	—	—	OVF	IMFB	IMFA
H'86	TCNT3H	16								
H'87	TCNT3L									
H'88	GRA3H	16								
H'89	GRA3L									
H'8A	GRB3H	16								
H'8B	GRB3L									
H'8C	BRA3H	16								
H'8D	BRA3L									
H'8E	BRB3H	16								
H'8F	BRB3L									

H'96	TCNT4H	16								
H'97	TCNT4L									
H'98	GRA4H	16								
H'99	GRA4L									
H'9A	GRB4H	16								
H'9B	GRB4L									
H'9C	BRA4H	16								
H'9D	BRA4L									
H'9E	BRB4H	16								
H'9F	BRB4L									
H'A0	TPMR	8	—	—	—	—	G3NOV	G2NOV	G1NOV	G0NOV
H'A1	TPCR	8	G3CMS1	G3CMS0	G2CMS1	G2CMS0	G1CMS1	G1CMS0	G0CMS1	G0CMS
H'A2	NDERB	8	NDER15	NDER14	NDER13	NDER12	NDER11	NDER10	NDER9	NDER8
H'A3	NDERA	8	NDER7	NDER6	NDER5	NDER4	NDER3	NDER2	NDER1	NDER0
H'A4	NDRB*1	8	NDR15	NDR14	NDR13	NDR12	NDR11	NDR10	NDR9	NDR8
		8	NDR15	NDR14	NDR13	NDR12	—	—	—	—
H'A5	NDRA*1	8	NDR7	NDR6	NDR5	NDR4	NDR3	NDR2	NDR1	NDR0
		8	NDR7	NDR6	NDR5	NDR4	—	—	—	—
H'A6	NDRB*1	8	—	—	—	—	—	—	—	—
		8	—	—	—	—	NDR11	NDR10	NDR9	NDR8
H'A7	NDRA*1	8	—	—	—	—	—	—	—	—
		8	—	—	—	—	NDR3	NDR2	NDR1	NDR0
H'A8	TCSR*2	8	OVF	WT/$\overline{IT}$	TME	—	—	CKS2	CKS1	CKS0
H'A9	TCNT*2	8								
H'AA	—		—	—	—	—	—	—	—	—
H'AB	RSTCSR*2	8	WRST	RSTOE	—	—	—	—	—	—
H'AC	RFSHCR	8	SRFMD	PSRAME	DRAME	CAS/$\overline{WE}$	M9/$\overline{M8}$	RFSHE	—	RCYCE
H'AD	RTMCSR	8	CMF	CMIE	CKS2	CKS1	CKS0	—	—	—
H'AE	RTCNT	8								
H'AF	RTCOR	8								

H'B6	SCMR	8	—	—	—	—	SDIR	SINV	—	SMIF
H'B7										
H'B8	SMR	8	C/$\overline{A}$	CHR	PE	O/$\overline{E}$	STOP	MP	CKS1	CKS0
H'B9	BRR	8								
H'BA	SCR	8	TIE	RIE	TE	RE	MPIE	TEIE	CKE1	CKE0
H'BB	TDR	8								
H'BC	SSR	8	TDRE	RDRF	ORER	FER	PER	TEND	MPB	MPBT
H'BD	RDR	8								
H'BE	—		—	—	—	—	—	—	—	—
H'BF										
H'C0	P1DDR	8	$P1_7$DDR	$P1_6$DDR	$P1_5$DDR	$P1_4$DDR	$P1_3$DDR	$P1_2$DDR	$P1_1$DDR	$P1_0$DDR
H'C1	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'C2	P1DR	8	$P1_7$	$P1_6$	$P1_5$	$P1_4$	$P1_3$	$P1_2$	$P1_1$	$P1_0$
H'C3	P2DR	8	$P2_7$	$P2_6$	$P2_5$	$P2_4$	$P2_3$	$P2_2$	$P2_1$	$P2_0$
H'C4	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'C5	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'C6	P3DR	8	$P3_7$	$P3_6$	$P3_5$	$P3_4$	$P3_3$	$P3_2$	$P3_1$	$P3_0$
H'C7	P4DR	8	$P4_7$	$P4_6$	$P4_5$	$P4_4$	$P4_3$	$P4_2$	$P4_1$	$P4_0$
H'C8	P5DDR	8	—	—	—	—	$P5_3$DDR	$P5_2$DDR	$P5_1$DDR	$P5_0$DDR
H'C9	P6DDR	8	—	$P6_6$DDR	$P6_5$DDR	$P6_4$DDR	$P6_3$DDR	$P6_2$DDR	$P6_1$DDR	$P6_0$DDR
H'CA	P5DR	8	—	—	—	—	$P5_3$	$P5_2$	$P5_1$	$P5_0$
H'CB	P6DR	8	—	$P6_6$	$P6_5$	$P6_4$	$P6_3$	$P6_2$	$P6_1$	$P6_0$
H'CC	—		—	—	—	—	—	—	—	—
H'CD	P8DDR	8	—	—	—	$P8_4$DDR	$P8_3$DDR	$P8_2$DDR	$P8_1$DDR	$P8_0$DDR
H'CE	P7DR	8	$P7_7$	$P7_6$	$P7_5$	$P7_4$	$P7_3$	$P7_2$	$P7_1$	$P7_0$
H'CF	P8DR	8	—	—	—	$P8_4$	$P8_3$	$P8_2$	$P8_1$	$P8_0$

H'D0	PBDR	8	PB_7	PB_6	PB_5	PB_4	PB_3	PB_2	PB_1	PB_0
H'D7	—		—	—	—	—	—	—	—	—
H'D8	P2PCR	8	$P2_7$PCR	$P2_6$PCR	$P2_5$PCR	$P2_4$PCR	$P2_3$PCR	$P2_2$PCR	$P2_1$PCR	$P2_0$PCR
H'D9	—		—	—	—	—	—	—	—	—
H'DA	P4PCR	8	$P4_7$PCR	$P4_6$PCR	$P4_5$PCR	$P4_4$PCR	$P4_3$PCR	$P4_2$PCR	$P4_1$PCR	$P4_0$PCR
H'DB	P5PCR	8	—	—	—	—	$P5_3$PCR	$P5_2$PCR	$P5_1$PCR	$P5_0$PCR
H'DC	DADR0	8								
H'DD	DADR1	8								
H'DE	DACR	8	DAOE1	DAOE0	DAE	—	—	—	—	—
H'DF	—		—	—	—	—	—	—	—	—
H'E0	ADDRAH	8	AD9	AD8	AD7	AD6	AD5	AD4	AD3	AD2
H'E1	ADDRAL	8	AD1	AD0	—	—	—	—	—	—
H'E2	ADDRBH	8	AD9	AD8	AD7	AD6	AD5	AD4	AD3	AD2
H'E3	ADDRBL	8	AD1	AD0	—	—	—	—	—	—
H'E4	ADDRCH	8	AD9	AD8	AD7	AD6	AD5	AD4	AD3	AD2
H'E5	ADDRCL	8	AD1	AD0	—	—	—	—	—	—
H'E6	ADDRDH	8	AD9	AD8	AD7	AD6	AD5	AD4	AD3	AD2
H'E7	ADDRDL	8	AD1	AD0	—	—	—	—	—	—
H'E8	ADCSR	8	ADF	ADIE	ADST	SCAN	CKS	CH2	CH1	CH0
H'E9	ADCR	8	TRGE	—	—	—	—	—	—	—
H'EA	—		—	—	—	—	—	—	—	—
H'EB	—		—	—	—	—	—	—	—	—
H'EC	ABWCR	8	ABW7	ABW6	ABW5	ABW4	ABW3	ABW2	ABW1	ABW0
H'ED	ASTCR	8	AST7	AST6	AST5	AST4	AST3	AST2	AST1	AST0
H'EE	WCR	8	—	—	—	—	WMS1	WMS0	WC1	WC0
H'EF	WCER	8	WCE7	WCE6	WCE5	WCE4	WCE3	WCE2	WCE1	WCE0

H'F6	ISR	8	—	—	IRQ5F	IRQ4F	IRQ3F	IRQ2F	IRQ1F	IRQ0F
H'F7	—		—	—	—	—	—	—	—	—
H'F8	IPRA	8	IPRA7	IPRA6	IPRA5	IPRA4	IPRA3	IPRA2	IPRA1	IPRA0
H'F9	IPRB	8	IPRB7	IPRB6	IPRB5	—	IPRB3	IPRB2	IPRB1	—
H'FA	—		—	—	—	—	—	—	—	—
H'FB	—		—	—	—	—	—	—	—	—
H'FC										
H'FD	—		—	—	—	—	—	—	—	—
H'FE	—		—	—	—	—	—	—	—	—
H'FF	—		—	—	—	—	—	—	—	—

Legend:

DMAC: DMA controller

ITU: 16-bit integrated timer unit

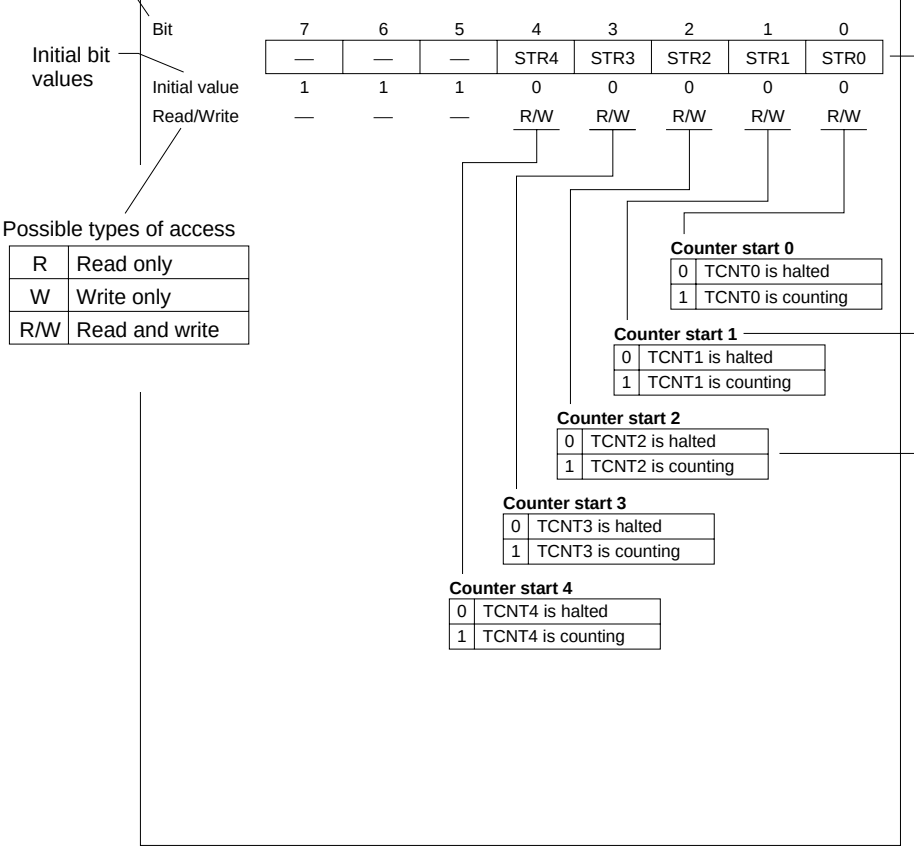
TPC: Programmable timing pattern controller

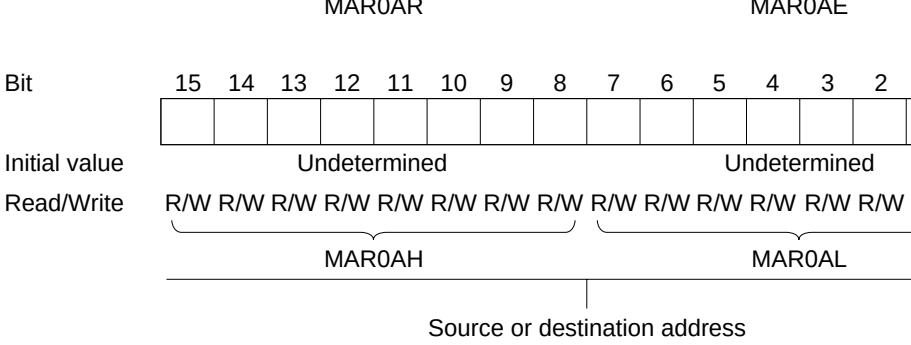
SCI: Serial communication interface

WDT: Watchdog timer

Notes: 1. The address depends on the output trigger setting.

2. For write access to TCSR, TCNT, and RSTCSR, see section 12.2.4, Notes o Rewriting.


Bit
7
6
5
4
3
2
1
0
Initial bit values
—
—
—
STR4
STR3
STR2
STR1
STR0
Initial value
1
1
1
0
0
0
0
0
Read/Write
—
—
—
R/W
R/W
R/W
R/W
R/W
Possible types of access
R Read only
W Write only
R/W Read and write
Counter start 0
0 TCNT0 is halted
1 TCNT0 is counting
Counter start 1
0 TCNT1 is halted
1 TCNT1 is counting
Counter start 2
0 TCNT2 is halted
1 TCNT2 is counting
Counter start 3
0 TCNT3 is halted
1 TCNT3 is counting
Counter start 4
0 TCNT4 is halted
1 TCNT4 is counting


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

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

Transfer counter

— Repeat mode

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

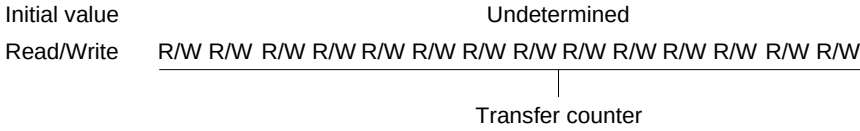
ETCR0AH

Transfer counter

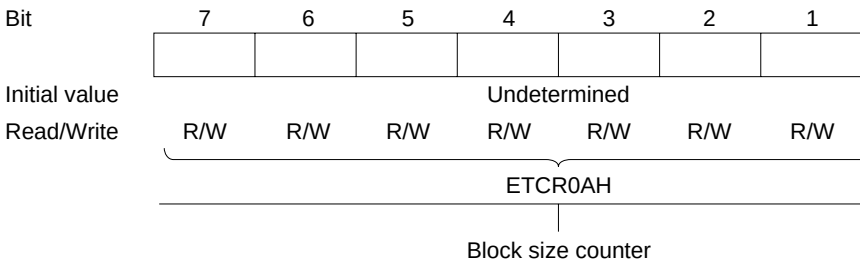
Bit	7	6	5	4	3	2	1	
Initial value	Undetermined							
Read/Write	R/W	R/W	R/W	R/W	R/W	R/W	R/W	

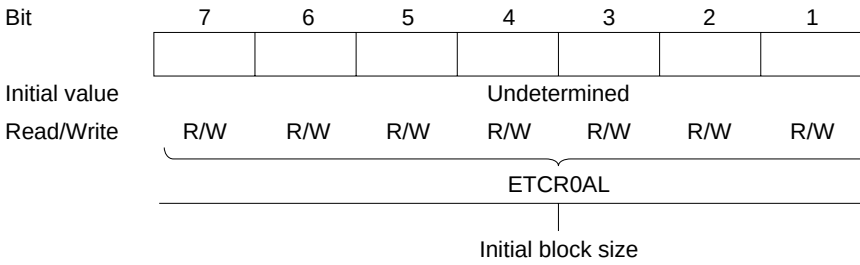
ETCR0AL

Initial count



— Block transfer mode





Full address mode: not used

Data transfer select

Bit 2	Bit 1	Bit 0	
DTS2	DTS1	DTS0	Data Transfer Activation Source
0	0	0	Compare match/input capture A interrupt from ITU chann
		1	Compare match/input capture A interrupt from ITU chann
	1	0	Compare match/input capture A interrupt from ITU chann
		1	Compare match/input capture A interrupt from ITU chann
1	0	0	SCI0 transmit-data-empty interrupt
		1	SCI0 receive-data-full interrupt
	1	0	Transfer in full address mode (channel A)
		1	Transfer in full address mode (channel A)

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 transfer sel

0	Normal mode
1	Block transfe

Data transfer select 2A

Set both bits to 1

Data transfer interrupt enable

0	Interrupt request by DTE bit is
1	Interrupt request by DTE bit is

Source address increment/decrement (bit 5)
Source address increment/decrement enable (bit 4)

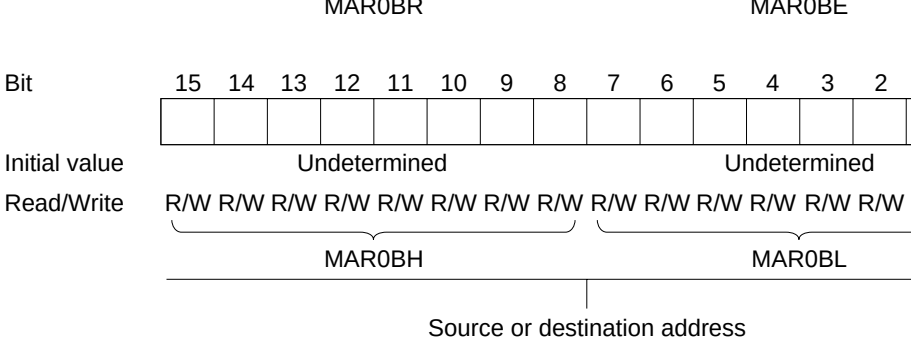
Bit 5	Bit 4	
SAID	SAIDE	Increment/Decrement Enable
0	0	MARA is held fixed
	1	Incremented: If DTSZ = 0, MARA is incremented by 1 after ea If DTSZ = 1, MARA is incremented by 2 after ea
1	0	MARA is held fixed
	1	Decremented: If DTSZ = 0, MARA is decremented by 1 after e If DTSZ = 1, MARA is decremented by 2 after e

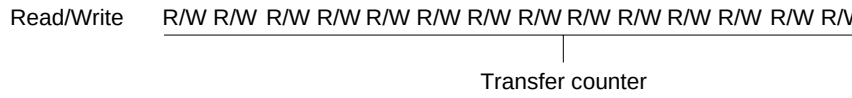
Data transfer size

0	Byte-size transfer
1	Word-size transfer

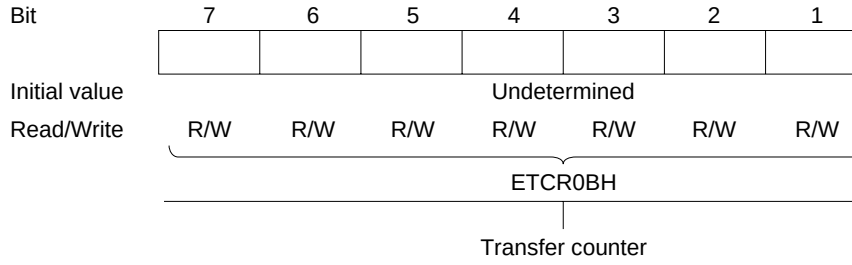
Data transfer enable

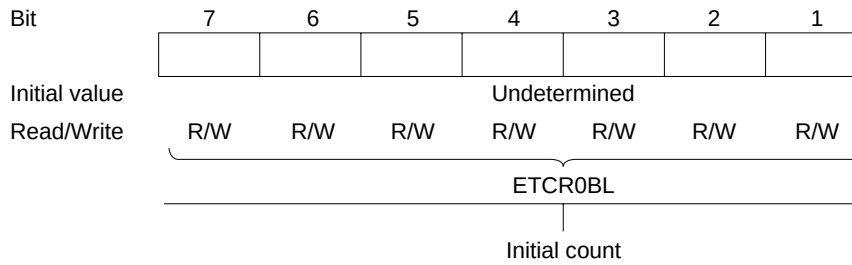
0	Data transfer is disabled
1	Data transfer is enabled

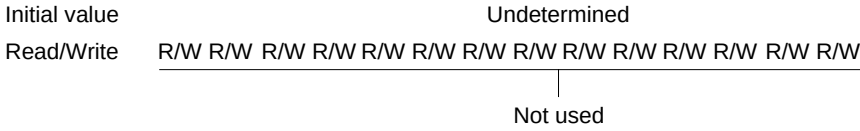

MAR0BR
MAR0BE
Bit
15 14 13 12 11 10 9 8 7 6 5 4 3 2
Initial value
Undetermined
Undetermined
Read/Write
R/W R/W R/W R/W R/W R/W R/W R/W R/W R/W R/W R/W R/W R/W
MAR0BH
MAR0BL
Source or destination address


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/
Transfer counter

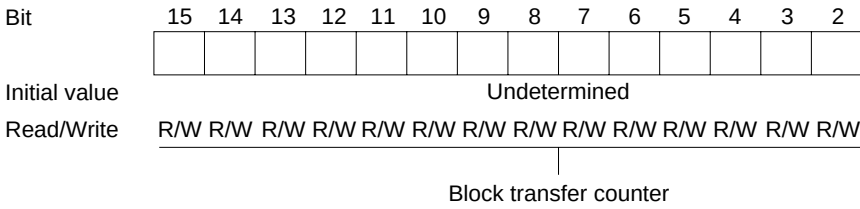
— Repeat mode


Bit
7 6 5 4 3 2 1
Initial value
Undetermined
Read/Write
R/W R/W R/W R/W R/W R/W R/W
ETCR0BH
Transfer counter

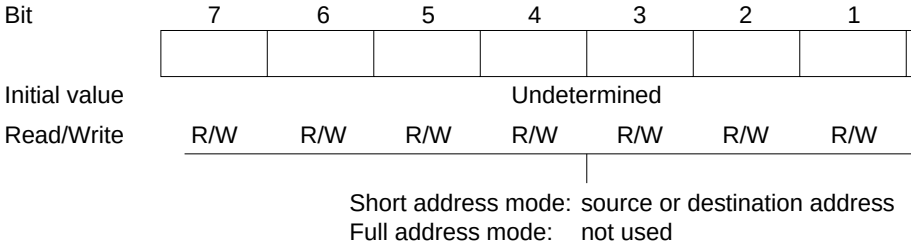

Bit
7 6 5 4 3 2 1
Initial value
Undetermined
Read/Write
R/W R/W R/W R/W R/W R/W R/W
ETCR0BL
Initial count



— Block transfer mode



IOAR0B—I/O Address Register 0B **H'2E**



Data transfer select

Bit 2	Bit 1	Bit 0	
DTS2	DTS1	DTS0	Data Transfer Activation Source
0	0	0	Compare match/input capture A interrupt from ITU c
		1	Compare match/input capture A interrupt from ITU c
	1	0	Compare match/input capture A interrupt from ITU c
		1	Compare match/input capture A interrupt from ITU c
1	0	0	SCI0 transmit-data-empty interrupt
		1	SCI0 receive-data-full interrupt
	1	0	Falling edge of $\overline{\text{DREQ}}$ input
		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 An interrupt request is issued to the CPU when the DTE bit = 0

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 transfer select 2B to 0B

Bit 2	Bit 1	Bit 0	Data Transfer Activation Source	
DTS2B	DTS1B	DTS0B	Normal Mode	Block Transfer Mode
0	0	0	Auto-request (burst mode)	Compare match/input capt A from ITU channel 0
		1	Not available	Compare match/input capt A from ITU channel 1
	1	0	Auto-request (cycle-steal mode)	Compare match/input capt A from ITU channel 2
		1	Not available	Compare match/input capt A from ITU channel 3
1	0	0	Not available	Not available
		1	Not available	Not available
	1	0	Falling edge of $\overline{\text{DREQ}}$	Falling edge of $\overline{\text{DREQ}}$
		1	Low level input at $\overline{\text{DREQ}}$	Not available

Transfer mode select

0	Destination is the block area in block transfer mode
1	Source is the block area in block transfer mode

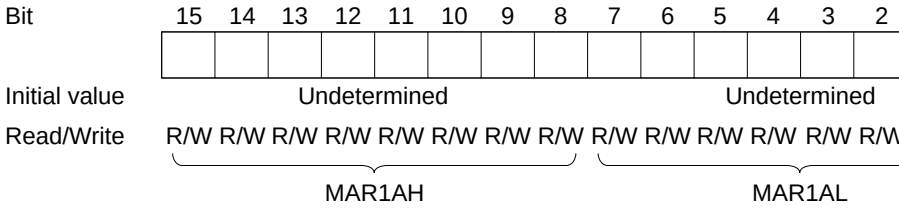
Destination address increment/decrement (bit 5)
Destination address increment/decrement enable (bit 4)

Bit 5	Bit 4	
DAID	DAIDE	Increment/Decrement Enable
0	0	MARB is held fixed
	1	Incremented: If DTSZ = 0, MARB is incremented by 1 after each tran If DTSZ = 1, MARB is incremented by 2 after each tran
1	0	MARB is held fixed
	1	Decremented: If DTSZ = 0, MARB is decremented by 1 after each tra If DTSZ = 1, MARB is decremented by 2 after each tra

Data transfer master enable

0	Data transfer is disabled
1	Data transfer is enabled

MAR1AR MAR1AE



Note: Bit functions are the same as for DMAC0.

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

ETCR1AH

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

ETCR1AL

Note: Bit functions are the same as for DMAC0.

IOAR1A—I/O Address Register 1A H'36

Bit	7	6	5	4	3	2	1	
Initial value	Undetermined							
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.

- Full address mode

Bit	7	6	5	4	3	2	1
	DTE	DTSZ	SAID	SAIDE	DTIE	DTS2A	DTS1A
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.

MAR1B R/E/H/L—Memory Address Register 1B R/E/H/L **H'38, H'39, H'3A, H'3B**

Bit	31	30	29	28	27	26	25	24	23	22	21	20	19	18
Initial value	1	1	1	1	1	1	1	1	Undetermined					
Read/Write	—	—	—	—	—	—	—	—	R/W	R/W	R/W	R/W	R/W	R/W
	MAR1BR								MAR1BE					

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
	MAR1BH								MAR1BL					

Note: Bit functions are the same as for DMAC0.

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

ETCR1BH

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

ETCR1BL

Note: Bit functions are the same as for DMAC0.

IOAR1B—I/O Address Register 1B **H'3E**

Bit	7	6	5	4	3	2	1	
Initial value	Undetermined							
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.

- Full address mode

Bit	7	6	5	4	3	2	1
	DTME	—	DAID	DAIDE	TMS	DTS2B	DTS1I
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.

Program mode

0	Exit from program mod (Initi
1	Transition to program m

Erase mode

0	Exit from erase mode	(Initia
1	Transition to erase mode	

Program-verify mode

0	Exit from program-verify mode	(Initi
1	Transition to program-verify mode	

Erase-verify mode

0	Exit from erase-verify mode	(Initial value)
1	Transition to erase-verify mode	

V_{PP} enable

0	V_{PP} pin 12 V power supply is disabled	(Initial value)
1	V_{PP} pin 12 V power supply is enabled	

Programming power

0	Cleared when 12 V is not applied to V_{PP}	(Initial value)
1	Set when 12 V is applied to V_{PP}	

Note: * The initial value is H'00 in modes 5, 6, and 7 (on-chip flash memory enabled). In 1, 2, 3, and 4 (on-chip flash memory disabled), this register cannot be modified a always read as H'FF.

H8/3048F	Include this register
H8/3048B mask ROM version H8/3048F-ONE H8/3048ZTAT H8/3048 mask ROM version H8/3047 mask ROM version H8/3045 mask ROM version H8/3044 mask ROM version	Not include this register

Program mode

0	Program mode cl (I
1	Transition to prog

Erase mode

0	Erase mode cleared (I
1	Transition to erase mode

Program-verify mode

0	Program-verify mode cleared (I
1	Transition to program-verify mode

Erase-verify mode

0	Erase-verify mode cleared (Initial valu
1	Transition to erase-verify mode

Program setup bit

0	Program setup cleared (Initial value)
1	Program setup

Erase setup bit

0	Erase setup cleared (Initial value)
1	Erase setup

Software write enable bit

0	Write disabled (Initial value)
1	Write enabled

Flash write enable bit

0	When a low level is input to the FWE pin (hardware protection state)
1	When a high level is input to the FWE pin

Note: * The initial value is H'00 in modes 5, 6, and 7 (on-chip flash memory enabled). In modes 1, 3, and 4 (on-chip flash memory disabled), this register cannot be modified and is always re as H'FF.

H8/3048F-ONE	Include this register
H8/3048B mask ROM version H8/3048F H8/3048ZTAT H8/3048 mask ROM version H8/3047 mask ROM version H8/3045 mask ROM version H8/3044 mask ROM version	Not include this register

Flash memory error

0	Flash memory is operating normally. Flash memory program/era protection (error protection) is disabled. (Initial val
1	This indicates that an error has occurred during flash memory programming/erasing. Flash memory program/erase protection (error protection) is enabled.

Note: Bits 6 to 0 are reserved bits but are readable/writable.

H8/3048F-ONE	Include this register
H8/3048B mask ROM version H8/3048F H8/3048ZTAT H8/3048 mask ROM version H8/3047 mask ROM version H8/3045 mask ROM version H8/3044 mask ROM version	Not include this register

Large block 7 to 0

0	Block LB7 to LB0 is not selected (Initial va
1	Block LB7 to LB0 is selected

Note: * The initial value is H'00 in modes 5, 6, and 7 (on-chip flash memory enabled
1, 2, 3, and 4 (on-chip flash memory disabled), this register cannot be modif
always read as H'FF.

H8/3048F	Include this register
H8/3048B mask ROM version H8/3048F-ONE H8/3048ZTAT H8/3048 mask ROM version H8/3047 mask ROM version H8/3045 mask ROM version H8/3044 mask ROM version	Not include this register

Erase block specification bits (1)

0	Erase protection state
1	Erasable state

Note: * The initial value is H'00 in modes 5, 6, and 7 (on-chip ROM enabled). In m
1, 2, 3, and 4 (on-chip ROM disabled), this register cannot be modified an
always read as H'00.

H8/3048F-ONE	Include this register
H8/3048B mask ROM version H8/3048F H8/3048ZTAT H8/3048 mask ROM version H8/3047 mask ROM version H8/3045 mask ROM version H8/3044 mask ROM version	Not include this register

Small block 7 to 0

0	Block SB7 to SB0 is not selected (Initial va
1	Block SB7 to SB0 is selected

Note: * The initial value is H'00 in modes 5, 6, and 7 (on-chip flash memory enabled
1, 2, 3, and 4 (on-chip flash memory disabled), this register cannot be modif
always read as H'FF.

H8/3048F	Include this register
H8/3048B mask ROM version H8/3048F-ONE H8/3048ZTAT H8/3048 mask ROM version H8/3047 mask ROM version H8/3045 mask ROM version H8/3044 mask ROM version	Not include this register

5 to 7	Read/Write	—	—	—	—	R/W	R/W	R/W

Reserved bits

RAM select, RAM2, RAM1

Bit 3	Bit 2	Bit 1	RAM Area	RAM Emulation
RAMS	RAM2	RAM1		
0	0/1	0/1	H'FFF000 to H'FFF3FF	No emulation
1	0	0	H'000000 to H'0003FF	Mapping RAM
		1	H'000400 to H'0007FF	
	1	0	H'000800 to H'000BFF	
		1	H'000C00 to H'000FFF	

Note: Bits 7 to 4 are reserved and cannot be modified.
If data is written to these bits, normal operation is not guaranteed.
Bit 0 is a reserved bit but is readable/writable.

H8/3048F-ONE	Include this register
H8/3048B mask ROM version H8/3048F H8/3048ZTAT H8/3048 mask ROM version H8/3047 mask ROM version H8/3045 mask ROM version H8/3044 mask ROM version	Not include this register

RAM select, RAM 2 to RAM 0

Bit 3	Bit 2	Bit 1	Bit 0	
RAMS	RAM 2	RAM 1	RAM 0	RAM Area
0	1/0	1/0	1/0	H'FFF000 to H'
1	0	0	0	H'01F000 to H'0
			1	H'01F200 to H'0
		1	0	H'01F400 to H'0
			1	H'01F600 to H'0
	1	0	0	H'01F800 to H'0
			1	H'01FA00 to H'0
		1	0	H'01FC00 to H'
			1	H'01FE00 to H'0

Flash memory error

0	Flash memory is not write/erase-protected (is not in error protect mode)	(Initial value)
1	Flash memory is write/erase-protected (is in error protect mode)	

H8/3048F	Include this register
H8/3048B mask ROM version H8/3048F-ONE H8/3048ZTAT H8/3048 mask ROM version H8/3047 mask ROM version H8/3045 mask ROM version H8/3044 mask ROM version	Not include this register

D/A standby enable

0	D/A output is disabled in software standby mode (Initia
1	D/A output is enabled in software standby mode

DIVCR—Division Control Register **H'5D** **Syste**

Bit	7	6	5	7	3	2	1
	—	—	—	—	—	—	DIV1
Initial value	1	1	1	1	1	1	0
Read/Write	—	—	—	—	—	—	R/W

Divide 1 and 0

Bit 1	Bit 0	Frequency
DIV1	DIV0	Division R
0	0	1/1 (Initia
	1	1/2
1	0	1/4
	1	1/8

Module standby 0

0	A/D converter operates norm (I
1	A/D converter is in standby st

Module standby 1

0	Refresh controller operates normally (
1	Refresh controller is in standby state

Module standby 2

0	DMAC operates normally (Initial value)
1	DMAC is in standby state

Module standby 3

0	SCI1 operates normally (Initial value)
1	SCI1 is in standby state

Module standby 4

0	SCI0 operates normally (Initial value)
1	SCI0 is in standby state

Module standby 5

0	ITU operates normally (Initial value)
1	ITU is in standby state

ϕ clock stop

0	ϕ clock output is enabled (Initial value)
1	ϕ clock output is disabled

Chip select 7 to 4 enable

Bit n	
CSnE	Description
0	Output of chip select signal CSn is disabled (Initial value)
1	Output of chip select signal CSn is enabled

(n = 7 to 4)

TSTR—Timer Start Register H'60 ITU (all

Bit	7	6	5	4	3	2	1	
	—	—	—	STR4	STR3	STR2	STR1	S
Initial value	1	1	1	0	0	0	0	
Read/Write	—	—	—	R/W	R/W	R/W	R/W	

Counter start 0

0	TCNT0 is halted
1	TCNT0 is count

Counter start 1

0	TCNT1 is halted
1	TCNT1 is counting

Counter start 2

0	TCNT2 is halted
1	TCNT2 is counting

Counter start 3

0	TCNT3 is halted
1	TCNT3 is counting

Counter start 4

0	TCNT4 is halted
1	TCNT4 is counting

Timer sync 0

0	TCNT0 operates indep
1	TCNT0 is synchronize

Timer sync 1

0	TCNT1 operates independent
1	TCNT1 is synchronized

Timer sync 2

0	TCNT2 operates independently
1	TCNT2 is synchronized

Timer sync 3

0	TCNT3 operates independently
1	TCNT3 is synchronized

Timer sync 4

0	TCNT4 operates independently
1	TCNT4 is synchronized

PWM mode 0

0	Channel 0 operates normally
1	Channel 0 operates in PWM

PWM mode 1

0	Channel 1 operates normally
1	Channel 1 operates in PWM mode

PWM mode 2

0	Channel 2 operates normally
1	Channel 2 operates in PWM mode

PWM mode 3

0	Channel 3 operates normally
1	Channel 3 operates in PWM mode

PWM mode 4

0	Channel 4 operates normally
1	Channel 4 operates in PWM mode

Flag direction

0	OVF is set to 1 in TSR2 when TCNT2 overflows or underflows
1	OVF is set to 1 in TSR2 when TCNT2 overflows

Phase counting mode flag

0	Channel 2 operates normally
1	Channel 2 operates in phase counting mode

Buffer mode A3

0	GRA3 operates no
1	GRA3 is buffered

Buffer mode B3

0	GRB3 operates normally
1	GRB3 is buffered by BRB

Buffer mode A4

0	GRA4 operates normally
1	GRA4 is buffered by BRA4

Buffer mode B4

0	GRB4 operates normally
1	GRB4 is buffered by BRB4

Combination mode 1 and 0

Bit 5	Bit 4	
CMD1	CMD0	Operating Mode of Channels 3 and 4
0	0	Channels 3 and 4 operate normally
	1	
1	0	Channels 3 and 4 operate together in complementary PWM m
	1	Channels 3 and 4 operate together in reset-synchronized PW

Timer prescaler 2 to 0

Bit 2	Bit 1	Bit 0	
TPSC2	TPSC1	TPSC0	TCNT Clock Source
0	0	0	Internal clock: ϕ
		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	
CKEG1	CKEG0	Counted Edges of External Clock
0	0	Rising edges counted
	1	Falling edges counted
1	—	Both edges counted

Counter clear 1 and 0

Bit 6	Bit 5	
CCLR1	CCLR0	TCNT Clear Source
0	0	TCNT is not cleared
	1	TCNT is cleared by GRA compare match or input
1	0	TCNT is cleared by GRB compare match or input
	1	Synchronous clear: TCNT is cleared in synchroniz with other synchronized timers

I/O control A2 to A0

Bit 2	Bit 1	Bit 0		
IOA2	IOA1	IOA0	GRA Function	
0	0	0	GRA is an output compare register	No output at compare match
		1		0 output at GRA compare mat
	1	0		1 output at GRA compare mat
		1		Output toggles at GRA compar
1	0	0	GRA is an input capture register	GRA captures rising edge of in
		1		GRA captures falling edge of in
	1	0		GRA captures both edges of in
		1		

I/O control B2 to B0

Bit 6	Bit 5	Bit 4		
IOB2	IOB1	IOB0	GRB Function	
0	0	0	GRB is an output compare register	No output at compare match
		1		0 output at GRB compare match
	1	0		1 output at GRB compare match
		1		Output toggles at GRB compare mat
1	0	0	GRB is an input capture register	GRB captures rising edge of input
		1		GRB captures falling edge of input
	1	0		GRB captures both edges of input
		1		

Input capture/compare match interrupt enab

0	IMIA interrupt requested by IMFA flag is dis
1	IMIA interrupt requested by IMFA flag is en

Input capture/compare match interrupt enable B

0	IMIB interrupt requested by IMFB flag is disabled
1	IMIB interrupt requested by IMFB flag is enabled

Overflow interrupt enable

0	OVI interrupt requested by OVF flag is disabled
1	OVI interrupt requested by OVF flag is enabled

Input capture/compare match flag A

0	[Clearing condition] Read IMFA when IMFA = 1, then write 0 in IMFA
1	[Setting conditions] TCNT = GRA when GRA functions as an output cor register. TCNT value is transferred to GRA by an input captu signal, when GRA functions as an input capture reg

Input capture/compare match flag B

0	[Clearing condition] Read IMFB when IMFB = 1, then write 0 in IMFB
1	[Setting conditions] TCNT = GRB when GRB functions as an output compare register. TCNT value is transferred to GRB by an input capture signal, when GRB functions as an input capture register.

Overflow flag

0	[Clearing condition] Read OVF when OVF = 1, then write 0 in OVF
1	[Setting condition] TCNT overflowed from H'FFFF to H'0000 or underflowed from H'0000 to H'FFFF

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

GRA0 H/L—General Register A0 H/L **H'6A, H'6B**

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'6C, H'6D**

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'6E**

Bit	7	6	5	4	3	2	1
	—	CCLR1	CCLR0	CKEG1	CKEG0	TPSC2	TPSC1
Initial value	1	0	0	0	0	0	0
Read/Write	—	R/W	R/W	R/W	R/W	R/W	R/W

Note: Bit functions are the same as for ITU0.

TIER1—Timer Interrupt Enable Register 1 **H'70**

Bit	7	6	5	4	3	2	1
	—	—	—	—	—	OVIE	IMIEB
Initial value	1	1	1	1	1	0	0
Read/Write	—	—	—	—	—	R/W	R/W

Note: Bit functions are the same as for ITU0.

TSR1—Timer Status Register 1 **H'71**

Bit	7	6	5	4	3	2	1
	—	—	—	—	—	OVF	IMFB
Initial value	1	1	1	1	1	0	0
Read/Write	—	—	—	—	—	R/(W)*	R/(W)*

Notes: Bit functions are the same as for ITU0.
* Only 0 can be written, to clear the flag.

TCNT1 H/L—Timer Counter 1 H/L **H'72, H'73**

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 ITU0.

GRB1 H/L—General Register B1 H/L **H'76, H'77**

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 ITU0.

TCR2—Timer Control Register 2 **H'78**

Bit	7	6	5	4	3	2	1	
	—	CCLR1	CCLR0	CKEG1	CKEG0	TPSC2	TPSC1	T
Initial value	1	0	0	0	0	0	0	
Read/Write	—	R/W	R/W	R/W	R/W	R/W	R/W	

Notes: 1. Bit functions are the same as for ITU0.
2. When channel 2 is used in phase counting mode, the counter clock source s
bits TPSC2 to TPSC0 is ignored.

TIER2—Timer Interrupt Enable Register 2 **H'7A**

Bit	7	6	5	4	3	2	1
	—	—	—	—	—	OVIE	IMIEB
Initial value	1	1	1	1	1	0	0
Read/Write	—	—	—	—	—	R/W	R/W

Note: Bit functions are the same as for ITU0.

TSR2—Timer Status Register 2 **H'7B**

Bit	7	6	5	4	3	2	1
	—	—	—	—	—	OVF	IMFB
Initial value	1	1	1	1	1	0	0
Read/Write	—	—	—	—	—	R/(W)*	R/(W)*

The function is the sar

Overflow flag

0	[Clearing condition] Read OVF when OVF = 1, then write 0 in C
1	[Setting condition] The TCNT value overflows (from H'FFFF to or underflows (from H'0000 to H'FFFF)

Notes: Bit functions are the same as for ITU0.
* Only 0 can be written, to clear the flag.

Other modes: up-counter

GRA2 H/L—General Register A2 H/L **H'7E, H'7F**

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 ITU0.

GRB2 H/L—General Register B2 H/L **H'80, H'81**

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 ITU0.

TIOR3—Timer I/O Control Register 3 **H'83**

Bit	7	6	5	4	3	2	1
	—	IOB2	IOB1	IOB0	—	IOA2	IOA1
Initial value	1	0	0	0	1	0	0
Read/Write	—	R/W	R/W	R/W	—	R/W	R/W

Note: Bit functions are the same as for ITU0.

TIER3—Timer Interrupt Enable Register 3 **H'84**

Bit	7	6	5	4	3	2	1
	—	—	—	—	—	OVIE	IMIEB
Initial value	1	1	1	1	1	0	0
Read/Write	—	—	—	—	—	R/W	R/W

Note: Bit functions are the same as for ITU0.

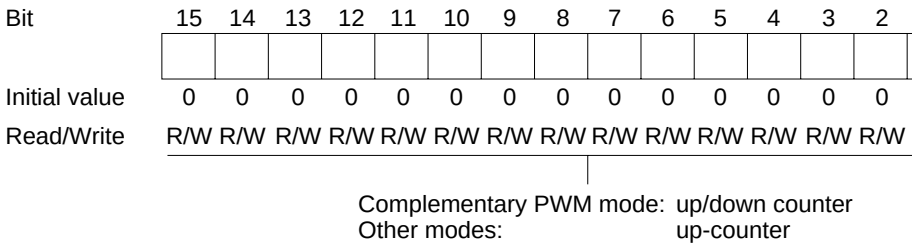
Bit function same as fo

Overflow flag

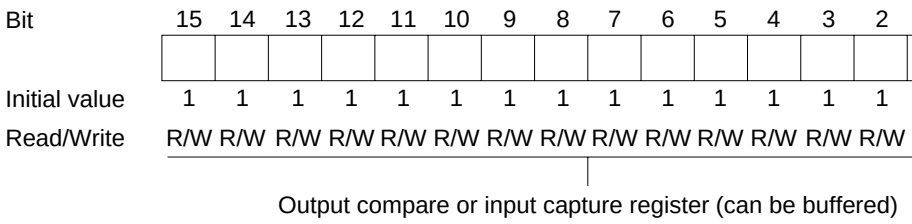
0	[Clearing condition] Read OVF when OVF = 1, then write 1 in OVF
1	[Setting condition] TCNT overflowed from H'FFFF to H'0000 or underflow H'0000 to H'FFFF

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

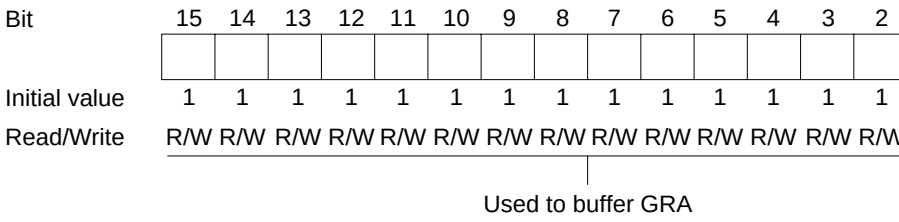
TCNT3 H/L—Timer Counter 3 H/L **H'86, H'87**



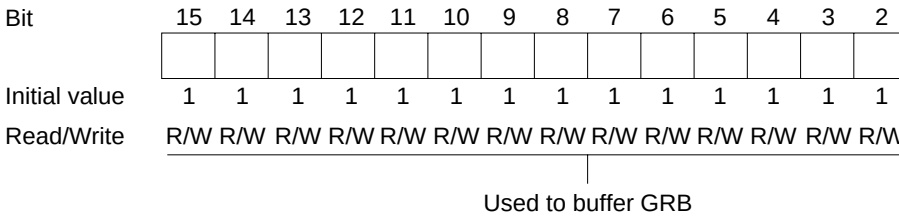
GRA3 H/L—General Register A3 H/L **H'88, H'89**



BRA3 H/L—Buffer Register A3 H/L **H'8C, H'8D**



BRB3 H/L—Buffer Register B3 H/L **H'8E, H'8F**



Master enable TIOCA3

0	$TIOCA_3$ output is disabled regardless of TIOR3, TMDR, and TFCR s
1	$TIOCA_3$ is enabled for output according to TIOR3, TMDR, and TFCF

Master enable TIOCA4

0	$TIOCA_4$ output is disabled regardless of TIOR4, TMDR, and TFCR set
1	$TIOCA_4$ is enabled for output according to TIOR4, TMDR, and TFCR s

Master enable TIOCB4

0	$TIOCB_4$ output is disabled regardless of TIOR4 and TFCR settings
1	$TIOCB_4$ is enabled for output according to TIOR4 and TFCR settings

Master enable TIOCB3

0	$TIOCB_3$ output is disabled regardless of TIOR3 and TFCR settings
1	$TIOCB_3$ is enabled for output according to TIOR3 and TFCR settings

Master enable TOCXA4

0	$TOCXA_4$ output is disabled regardless of TFCR settings
1	$TOCXA_4$ is enabled for output according to TFCR settings

Master enable TOCXB4

0	$TOCXB_4$ output is disabled regardless of TFCR settings
1	$TOCXB_4$ is enabled for output according to TFCR settings

Output level select 3

0	$TIOCB_3$, $TOCXA_4$, and $TOCXB_4$ outputs are inverte
1	$TIOCB_3$, $TOCXA_4$, and $TOCXB_4$ outputs are not inv

Output level select 4

0	$TIOCA_3$, $TIOCA_4$, and $TIOCB_4$ outputs are inverted
1	$TIOCA_3$, $TIOCA_4$, and $TIOCB_4$ outputs are not inverted

External trigger disable

0	Input capture A in channel 1 is used as an external trigger signal in reset-synchronized PWM mode and complementary PWM mode*
1	External triggering is disabled

Note: * When an external trigger occurs, bits 5 to 0 in TOER are cleared to 0, disab output.

TIOR4—Timer I/O Control Register 4 **H'93**

Bit	7	6	5	4	3	2	1
	—	IOB2	IOB1	IOB0	—	IOA2	IOA1
Initial value	1	0	0	0	1	0	0
Read/Write	—	R/W	R/W	R/W	—	R/W	R/W

Note: Bit functions are the same as for ITU0.

TIER4—Timer Interrupt Enable Register 4 **H'94**

Bit	7	6	5	4	3	2	1
	—	—	—	—	—	OVIE	IMIEB
Initial value	1	1	1	1	1	0	0
Read/Write	—	—	—	—	—	R/W	R/W

Note: Bit functions are the same as for ITU0.

TSR4—Timer Status Register 4 **H'95**

Bit	7	6	5	4	3	2	1
	—	—	—	—	—	OVF	IMFB
Initial value	1	1	1	1	1	0	0
Read/Write	—	—	—	—	—	R/(W)*	R/(W)*

Notes: Bit functions are the same as for ITU0.
* Only 0 can be written, to clear the flag.

GRA4 H/L—General Register A4 H/L **H'98, H'99**

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 ITU3.

GRB4 H/L—General Register B4 H/L **H'9A, H'9B**

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 ITU3.

BRA4 H/L—Buffer Register A4 H/L **H'9C, H'9D**

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 ITU3.

Bit	7	6	5	4	3	2	1
	—	—	—	—	G3NOV	G2NOV	G1NOV
Initial value	1	1	1	1	0	0	0
Read/Write	—	—	—	—	R/W	R/W	R/W

Group 0 non-overlap

0	Normal TPC output in group 0 Output values change at compare match A in the selected ITU cl
1	Non-overlapping TPC output in group 0, controlled by compare n A and B in the selected ITU channel

Group 1 non-overlap

0	Normal TPC output in group 1 Output values change at compare match A in the selected ITU chan
1	Non-overlapping TPC output in group 1, controlled by compare mat A and B in the selected ITU channel

Group 2 non-overlap

0	Normal TPC output in group 2 Output values change at compare match A in the selected ITU channel
1	Non-overlapping TPC output in group 2, controlled by compare match A and B in the selected ITU channel

Group 3 non-overlap

0	Normal TPC output in group 3 Output values change at compare match A in the selected ITU channel
1	Non-overlapping TPC output in group 3, controlled by compare match A and B in the selected ITU channel

Group 0 compare match select 1 and 0

Bit 1	Bit 0	
G0CMS1	G0CMS0	ITU Channel Selected as Output Trigger
0	0	TPC output group 0 (TP_3 to TP_0) is triggered by compare match in I
	1	TPC output group 0 (TP_3 to TP_0) is triggered by compare match in I
1	0	TPC output group 0 (TP_3 to TP_0) is triggered by compare match in I
	1	TPC output group 0 (TP_3 to TP_0) is triggered by compare match in I

Group 1 compare match select 1 and 0

Bit 3	Bit 2	
G1CMS1	G1CMS0	ITU Channel Selected as Output Trigger
0	0	TPC output group 1 (TP_7 to TP_4) is triggered by compare match in ITU
	1	TPC output group 1 (TP_7 to TP_4) is triggered by compare match in ITU
1	0	TPC output group 1 (TP_7 to TP_4) is triggered by compare match in ITU
	1	TPC output group 1 (TP_7 to TP_4) is triggered by compare match in ITU

Group 2 compare match select 1 and 0

Bit 5	Bit 4	
G2CMS1	G2CMS0	ITU Channel Selected as Output Trigger
0	0	TPC output group 2 (TP_{11} to TP_8) is triggered by compare match in ITU
	1	TPC output group 2 (TP_{11} to TP_8) is triggered by compare match in ITU
1	0	TPC output group 2 (TP_{11} to TP_8) is triggered by compare match in ITU
	1	TPC output group 2 (TP_{11} to TP_8) is triggered by compare match in ITU

Group 3 compare match select 1 and 0

Bit 7	Bit 6	
G3CMS1	G3CMS0	ITU Channel Selected as Output Trigger
0	0	TPC output group 3 (TP_{15} to TP_{12}) is triggered by compare match in ITU c
	1	TPC output group 3 (TP_{15} to TP_{12}) is triggered by compare match in ITU c
1	0	TPC output group 3 (TP_{15} to TP_{12}) is triggered by compare match in ITU c
	1	TPC output group 3 (TP_{15} to TP_{12}) is triggered by compare match in ITU c

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
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'A3

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 P
1	TPC outputs TP_7 to TP_0 are enabled (NDR7 to NDR0 are transferred to PA_7 to PA_0)

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

Store the next output data for TPC output group 3

Store the next output da TPC output group 2

— Address H'FFA6

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'FFA4

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'FFA6

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 da TPC output group 2

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

Store the next output data for TPC output group 1

Store the next output dat TPC output group 0

— Address H'FFA7

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'FFA5

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'FFA7

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 dat TPC output group 0

Timer enable

0	Timer disabled • TCNT is initialized to H'00 and halted
1	Timer enabled • TCNT is counting • CPU interrupt requests are enabled

Clock select 2 to 0

0	0	0
		1
	1	0
		1
1	0	0
		1
	1	0
		1

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.

RSTCSR—Reset Control/Status Register **H'AB (read), H'AA (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 input at $\overline{\text{RES}}$ pin • When WRST = 1, write 0 after reading WRST flag
1	[Setting condition] TCNT overflow generates a reset signal

Note: * Only 0 can be written in bit 7, to clear the flag.

H8/3048F-ONE	Not include this register
H8/3048B mask ROM version H8/3048F H8/3048ZTAT H8/3048 mask ROM version H8/3047 mask ROM version H8/3045 mask ROM version H8/3044 mask ROM version	Include this register

Watchdog timer reset

0	[Clearing conditions] • Reset signal input at $\overline{\text{RES}}$ pin • When WRST = 1, write 0 after reading WRST flag
1	[Setting condition] TCNT overflow generates a reset signal

Notes: 1. Only 0 can be written in bit 7, to clear the flag.
2. Bit 6 must not be set to 1; in a write, 0 must always be written in this bit.

H8/3048F-ONE	Include this register
H8/3048B mask ROM version H8/3048F H8/3048ZTAT H8/3048 mask ROM version H8/3047 mask ROM version H8/3045 mask ROM version H8/3044 mask ROM version	Not include this register

Refresh cycle enable

0	Refresh cycles are disabled
1	Refresh cycles are enabled fo

Refresh pin enable

0	Refresh signal output at the $\overline{\text{RFSH}}$ pin is d
1	Refresh signal output at the $\overline{\text{RFSH}}$ pin is e

Address multiplex mode select

0	8-bit column mode
1	9-bit column mode

Strobe mode select

0	2 $\overline{\text{WE}}$ mode
1	2 $\overline{\text{CAS}}$ mode

PSRAM enable, DRAM enable

Bit 6	Bit 5	
PSRAME	DRAME	RAM Interface
0	0	Can be used as an interval timer (DRAM and PSRAM cannot be directly connected)
	1	DRAM can be directly connected
1	0	PSRAM can be directly connected
	1	Illegal setting

Self-refresh mode

0	DRAM or PSRAM self-refresh is disabled in software standby mode
1	DRAM or PSRAM self-refresh is enabled in software standby mode

Bit 5	Bit 4	Bit 3	
CKS2	CKS1	CKS0	Counter Clock Sourc
0	0	0	Clock input is disabl
		1	ϕ/2
	1	0	ϕ/8
		1	ϕ/32
1	0	0	ϕ/128
		1	ϕ/512
	1	0	ϕ/2048
		1	ϕ/4096

Compare match interrupt enable

0	The CMI interrupt requested by CMF is disabled
1	The CMI interrupt requested by CMF is enabled

Compare match flag

0	[Clearing condition] Read CMF when CMF = 1, then write 0 in CMF
1	[Setting condition] RTCNT = RTCOR

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

RTCOR—Refresh Time Constant Register **H'AF** **Refresh**

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

Interval at which RTCNT and compare match are set

Clock select 1 and 0

Bit 1	Bit 0	
CKS1	CKS0	Clock
0	0	φ cloc
	1	φ/4 cl
1	0	φ/16 c
	1	φ/64 c

Multiprocessor mode

0	Multiprocessor function disabled
1	Multiprocessor format selected

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
(when using a serial communication interface)

0	Asynchronous mode
1	Synchronous mode

GSM mode (when using a smart card interface)

0	Regular smart card interface operation
1	GSM mode smart card interface operation

Note: * The function of this bit differs for the normal serial communication interface an
for the smart card interface. Its function is switched with the SMIF bit in SCMR

Clock enable 1 and 0

Bit 1	Bit 0		
CKE1	CKE0	Clock Selection and Output	
0	0	Asynchronous mode	Internal clock, SCK pin available for ge
		Synchronous mode	Internal clock, SCK pin used for serial c
	1	Asynchronous mode	Internal clock, SCK pin used for clock c
		Synchronous mode	Internal clock, SCK pin used for serial c
1	0	Asynchronous mode	External clock, SCK pin used for clock
		Synchronous mode	External clock, SCK pin used for serial
	1	Asynchronous mode	External clock, SCK pin used for clock
		Synchronous mode	External clock, SCK pin used for serial

Transmit-end interrupt enable

0	Transmit-end interrupt requests (TEI) are disabled
1	Transmit-end interrupt requests (TEI) are enabled

Multiprocessor interrupt enable

0	Multiprocessor interrupts are disabled (normal receive operation)
1	Multiprocessor interrupts are enabled

Receive enable

0	Receiving is disabled
1	Receiving is enabled

Transmit enable

0	Transmitting is disabled
1	Transmitting is 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

Multiprocessor bit

0	Multiprocessor bit value in receive data is 0
1	Multiprocessor bit value in receive data is 1

Multiprocessor bit tra

0	Multiprocessor bit va transmit data is 0
1	Multiprocessor bit va transmit data is 1

Parity error

0	[Clearing conditions] Reset or transition to standby mode. Read PER when PER = 1, then write 0 in PER.
1	[Setting condition] Parity error: (parity of receive data does not match parity setting O/$\overline{E}$ bit in SMR)

Transmit end

0	[Clearing conditions] Read TDRE when TDRE = 1, then write 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 cleared to 0. TDRE is 1 when last bit of 1-byte serial is transmitted.

Framing error (for SCI0)

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 inte

0	[Clearing conditions] Reset or transition to standby mode. Read ERS when ERS = 1, then write 0 in
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 ORER.
1	[Setting condition] Overrun error (reception of next serial da ends when RDRF = 1)

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

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	—

Smart card interface mode select

0	Smart card interface function is disabled (Initial
1	Smart card interface function is enabled

Smart card data invert

0	Unmodified TDR contents are transmitted (Initial value) Received data is stored unmodified in RDR
1	Inverted TDR contents are transmitted Received data are inverted before storage in RDR

Smart card data transfer direction

0	TDR contents are transmitted LSB-first (Initial value) Received data is stored LSB-first in RDR
1	TDR contents are transmitted MSB-first Received data is stored MSB-first in RDR

BRR—Bit Rate Register **H'B9**

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'BA**

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.

SSR—Serial Status Register **H'BC**

Bit	7	6	5	4	3	2	1
	TDRE	RDRF	ORER	FER	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

Notes: Bit functions are the same as for SCI0.
* Only 0 can be written, to clear the flag.

RDR—Receive Data Register **H'BD**

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.

Port 1 input/output select

0	Generic input pin
1	Generic output pin

P2DDR—Port 2 Data Direction Register H'C1

Bit		7	6	5	4	3	2	1
		$P2_7$DDR	$P2_6$DDR	$P2_5$DDR	$P2_4$DDR	$P2_3$DDR	$P2_2$DDR	$P2_1$DD
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 input/output select

0	Generic input pin
1	Generic output pin

P1DR—Port 1 Data Register H'C2

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

P3DDR—Port 3 Data Direction Register **H'C4**

Bit	7	6	5	4	3	2	1
	$P3_7DDR$	$P3_6DDR$	$P3_5DDR$	$P3_4DDR$	$P3_3DDR$	$P3_2DDR$	$P3_1DDR$
Initial value	0	0	0	0	0	0	0
Read/Write	W	W	W	W	W	W	W

Port 3 input/output select

0	Generic input pin
1	Generic output pin

P4DDR—Port 4 Data Direction Register **H'C5**

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 input/output select

0	Generic input pin
1	Generic output pin

P4DR—Port 4 Data Register **H'C7**

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

P5DDR—Port 5 Data Direction Register **H'C8**

Bit		7	6	5	4	3	2	1
		—	—	—	—	$P5_3DDR$	$P5_2DDR$	$P5_1DD$
Modes 1 to 4	Initial value	1	1	1	1	1	1	1
	Read/Write	—	—	—	—	—	—	—
Modes 5 to 7	Initial value	1	1	1	1	0	0	0
	Read/Write	—	—	—	—	W	W	W

Port 5 input/output

0	Generic input
1	Generic output

0	Generic input
1	Generic output

P5DR—Port 5 Data Register **H'CA**

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'CB**

Bit	7	6	5	4	3	2	1
	—	$P6_6$	$P6_5$	$P6_4$	$P6_3$	$P6_2$	$P6_1$
Initial value	1	0	0	0	0	0	0
Read/Write	—	R/W	R/W	R/W	R/W	R/W	R/W

Data for port 6 pins

Port 8 input/output select

0	Generic input
1	$\overline{CS}$ output

Port 8 input/o

0	Generic in
1	Generic ou

P7DR—Port 7 Data Register H'CE

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

Read the pin levels for port 7

Note: * Determined by pins $P7_7$ to $P7_0$.

P8DR—Port 8 Data Register H'CF

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

Data for port 8 pins

0	Generic input
1	Generic output

PADDR—Port A Data Direction Register **H'D1**

Bit		7	6	5	4	3	2	1
		PA_7DDR	PA_6DDR	PA_5DDR	PA_4DDR	PA_3DDR	PA_2DDR	PA_1DDR
Modes 3, 4, 6	Initial value	1	0	0	0	0	0	0
	Read/Write	—	W	W	W	W	W	W
Modes 1, 2, 5, 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

P9DR—Port 9 Data Register **H'D2**

Bit	7	6	5	4	3	2	1
	—	—	$P9_5$	$P9_4$	$P9_3$	$P9_2$	$P9_1$
Initial value	1	1	0	0	0	0	0
Read/Write	—	—	R/W	R/W	R/W	R/W	R/W

Data for port 9 pins

PBDDR—Port B Data Direction Register **H'D4**

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 input/output select

0	Generic input
1	Generic output

PBDR—Port B Data Register **H'D6**

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

Data for port B pins

0	Input pull-up transistor is off
1	Input pull-up transistor is on

Note: Valid when the corresponding P2DDR bit is cleared to 0 (designating generic in

P4PCR—Port 4 Input Pull-Up MOS Control Register H'DA

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 MOS control 7 to 0

0	Input pull-up transistor is off
1	Input pull-up transistor is on

Note: Valid when the corresponding P4DDR bit is cleared to 0 (designating generic in

0	Input pull-up transistor is of
1	Input pull-up transistor is on

Note: Valid when the corresponding P5DDR bit is cleared to 0 (designating generic in

DADR0—D/A Data Register 0 **H'DC**

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

D/A conversion data

DADR1—D/A Data Register 1 **H'DD**

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

D/A conversion data

D/A enable

Bit 7	Bit 6	Bit 5	
DAOE1	DAOE0	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

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

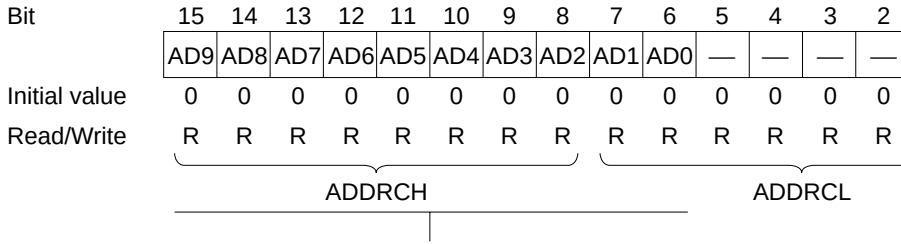
ADDRA H/L—A/D Data Register A H/L — H'E0, H'E1

Bit	15	14	13	12	11	10	9	8	7	6	5	4	3	2
	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
Read/Write	R	R	R	R	R	R	R	R	R	R	R	R	R	R
	ADDRAH								ADDRAL					

A/D conversion data
10-bit data giving an A/D conversion result

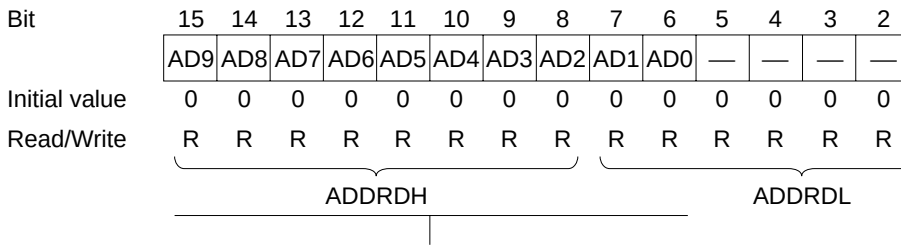
A/D conversion data
10-bit data giving an
A/D conversion result

ADDRC H/L—A/D Data Register C H/L — H'E4, H'E5



A/D conversion data
10-bit data giving an
A/D conversion result

ADDRD H/L—A/D Data Register D H/L — H'E6, H'E7



A/D conversion data
10-bit data giving an
A/D conversion result

Clock select

0	Conversion time = 266 states (maximum)
1	Conversion time = 134 states (maximum)

Channel select 2 to 0

Group Selection	Channel Selection		Description	
CH2	CH1	CH0	Single Mode	Scan
0	0	0	AN_0	AN_0
		1	AN_1	AN_0,
	1	0	AN_2	AN_0
		1	AN_3	AN_0
1	0	0	AN_4	AN_4
		1	AN_5	AN_4,
	1	0	AN_6	AN_4
		1	AN_7	AN_4

Scan mode

0	Single mode
1	Scan mode

A/D start

0	A/D conversion is stopped
1	Single mode: A/D conversion starts; ADST is automatically cleared to 0 whe conversion ends Scan mode: A/D conversion starts and continues, cycling among the select channels, until ADST is cleared to 0 by software, by a reset, o 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 condition] Read ADF while ADF = 1, then write 0 in ADF
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 flag.

0	A/D conversion cannot be externally triggered
1	A/D conversion starts at the fall of the external trigger signal ($\overline{AD}$

H8/3048F-ONE	Not include this register
H8/3048F H8/3048B mask ROM version H8/3048ZTAT H8/3048 mask ROM version H8/3047 mask ROM version H8/3045 mask ROM version H8/3044 mask ROM version	Include this register

ADCR—A/D Control Register H'E9

Bit	7	6	5	4	3	2	1
	TRGE	—	—	—	—	—	—
Initial value	0	1	1	1	1	1	1
Read/Write	R/W	—	—	—	—	—	—

Trigger enable

0	A/D conversion cannot be externally triggered
1	A/D conversion starts at the fall of the external trigger signal ($\overline{AD}$

Note: * Bit 0 must not be set to 1; in a write, 0 must always be written in this bit.

H8/3048F-ONE	Include this register
H8/3048F H8/3048B mask ROM version H8/3048ZTAT H8/3048 mask ROM version H8/3047 mask ROM version H8/3045 mask ROM version H8/3044 mask ROM version	Not include this register

Area 7 to 0 bus width control

Bits 7 to 0	
ABW7 to ABW0	Bus Width of Access Area
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'ED Bus

Bit	7	6	5	4	3	2	1
	AST7	AST6	AST5	AST4	AST3	AST2	AST1
Initial value	1	1	1	1	1	1	1
Read/Write	R/W	R/W	R/W	R/W	R/W	R/W	R/W

Area 7 to 0 access state control

Bits 7 to 0	
AST7 to AST0	Number of States in Access Cycle
0	Areas 7 to 0 are two-state access areas
1	Areas 7 to 0 are three-state access areas

Wait mode select 1 and 0

Bit 3	Bit 2	
WMS1	WMS0	Wait Mode
0	0	Programmable wait mode
	1	No wait states inserted by wait-state controller
1	0	Pin wait mode 1
	1	Pin auto-wait mode

Wait count 1 and 0

Bit 1	Bit 0	
WC1	WC0	Number of Wait
0	0	No wait states in wait-state contro
	1	1 state inserted
1	0	2 states inserted
	1	3 states inserted

WCER—Wait-State Controller Enable Register **H'EF** **Bus**

Bit	7	6	5	4	3	2	1
	WCE7	WCE6	WCE5	WCE4	WCE3	WCE2	WCE1
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

Wait-state controller enable 7 to 0

0	Wait-state control is disabled (pin wait mode
1	Wait-state control is enabled

Bit 2	Bit 1	Bit 0	
MD_2	MD_1	MD_0	Operatin
0	0	0	—
		1	Mode 1
	1	0	Mode 2
		1	Mode 3
1	0	0	Mode 4
		1	Mode 5
	1	0	Mode 6
		1	Mode 7

Note: * Determined by the state of the mode pins (MD_2 to MD_0).

RAM enable

0	On-chip RAM
1	On-chip RAM

NMI edge select

0	An interrupt is requested at the falling e
1	An interrupt is requested at the rising e

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

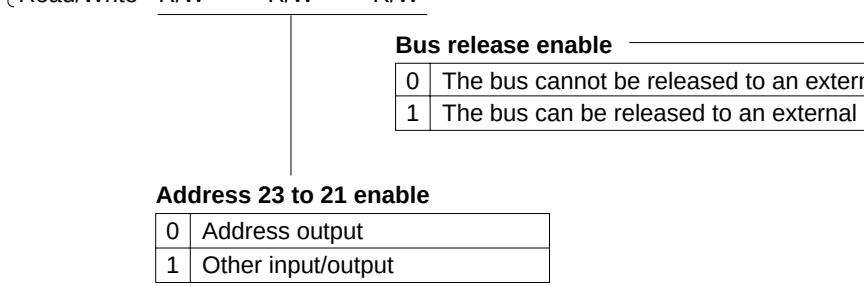
Standby timer select 2 to 0

Bit 6	Bit 5	Bit 4	Standby Timer	
STS2	STS1	STS0	H8/3048F-ONE H8/3048B mask ROM version	*
0	0	0	Waiting time = 8,192 states	Waiting time = 8,1
		1	Waiting time = 16,384 states	Waiting time = 16,
	1	0	Waiting time = 32,768 states	Waiting time = 32,
		1	Waiting time = 65,536 states	Waiting time = 65,
1	0	0	Waiting time = 131,072 states	Waiting time = 131
		1	Waiting time = 262,144 states	Waiting time = 1,0
	1	0	Waiting time = 1,024 states	Illegal setting
		1	Illegal setting	Illegal setting

Note: * H8/3048F
H8/3048ZTAT
H8/3048 mask ROM version
H8/3047 mask ROM version
H8/3045 mask ROM version
H8/3044 mask ROM version

Software standby

0	SLEEP instruction causes transition to sleep mode
1	SLEEP instruction causes transition to software standby mode



ISCR—IRQ Sense Control Register **H'F4 Interrupt**

Bit	7	6	5	4	3	2	1
	—	—	IRQ5SC	IRQ4SC	IRQ3SC	IRQ2SC	IRQ1SC
Initial value	0	0	0	0	0	0	0
Read/Write	R/W	R/W	R/W	R/W	R/W	R/W	R/W

IRQ_5 to IRQ_0 sense control

0	Interrupts are requested when $\overline{IRQ_5}$ to $\overline{IRQ_0}$ inputs a
1	Interrupts are requested by falling-edge input at $\overline{IRQ}$

0	IRQ_5 to IRQ_0 interrupts are disabled
1	IRQ_5 to IRQ_0 interrupts are enabled

ISR—IRQ Status Register — H'F6 — Interrupt

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)*	

IRQ_5 to IRQ_0 flags

Bits 5 to 0	
IRQ5F to IRQ0F	Setting and Clearing Conditions
0	[Clearing conditions] Read IRQnF when IRQnF = 1, then write 0 in IRQ IRQnSC = 0, $\overline{IRQn}$ input is high, and interrupt exc handling is carried out. IRQnSC = 1 and IRQn interrupt exception handlin carried out.
1	[Setting conditions] IRQnSC = 0 and $\overline{IRQn}$ input is low. IRQnSC = 1 and a falling edge is generated in the

(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

	Bit 7: IPRA7	Bit 6: IPRA6	Bit 5: IPRA5	Bit 4: IPRA4	Bit 3: IPRA3	Bit 2: IPRA2	Bit 1: IPRA1
Interrupt source	IRQ_0	IRQ_1	IRQ_2, IRQ_3	IRQ_4, IRQ_5	WDT, Refresh Controller	ITU channel 0	ITU channel 1

IPRB—Interrupt Priority Register B **H'F9** **Interrupt**

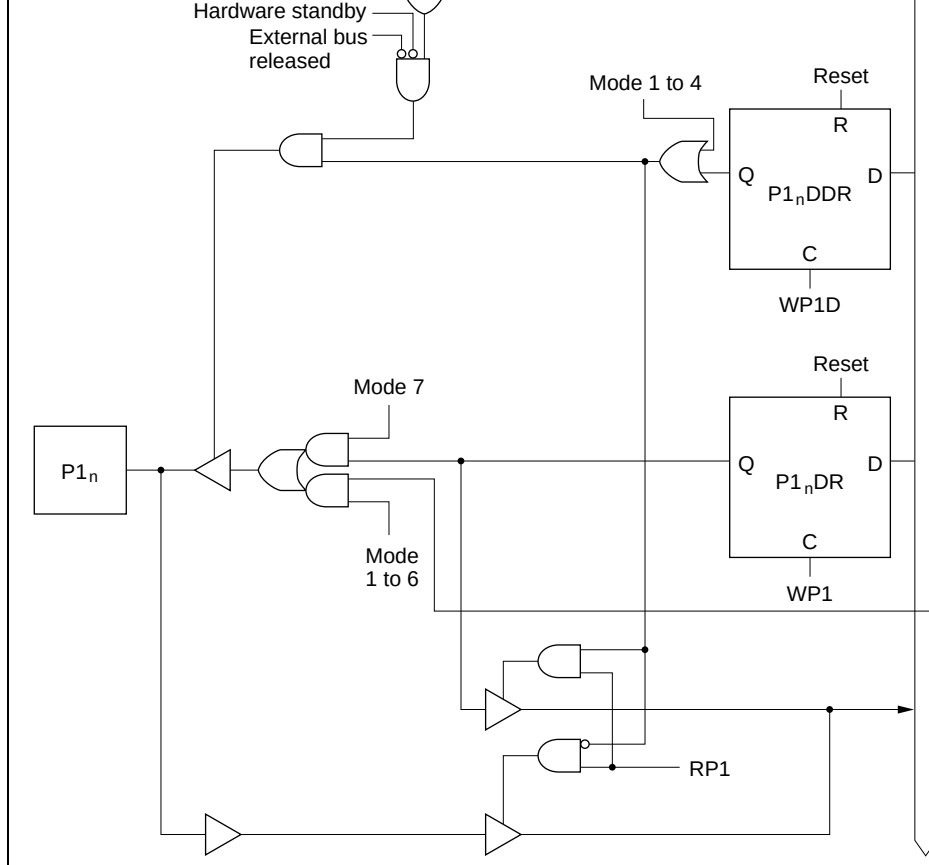
Bit	7	6	5	4	3	2	1
	IPRB7	IPRB6	IPRB5	—	IPRB3	IPRB2	IPRB1
Initial value	0	0	0	0	0	0	0
Read/Write	R/W	R/W	R/W	R/W	R/W	R/W	R/W

Priority level B7 to B5, B3 to B 1

0	Priority level 0 (low priority)
1	Priority level 1 (high priority)

- Interrupt sources controlled by each bit

	Bit 7: IPRB7	Bit 6: IPRB6	Bit 5: IPRB5	Bit 4: —	Bit 3: IPRB3	Bit 2: IPRB2	Bit 1: IPRB1
Interrupt source	ITU channel 3	ITU channel 4	DMAC	—	SCI channel 0	SCI channel 1	A/D converte



Legend:
WP1D: Write to P1DDR
WP1: Write to port 1
RP1: Read port 1

Note: n = 0 to 7

Figure C.1 Port 1 Block Diagram (Pins $P1_0$ to $P1_7$)

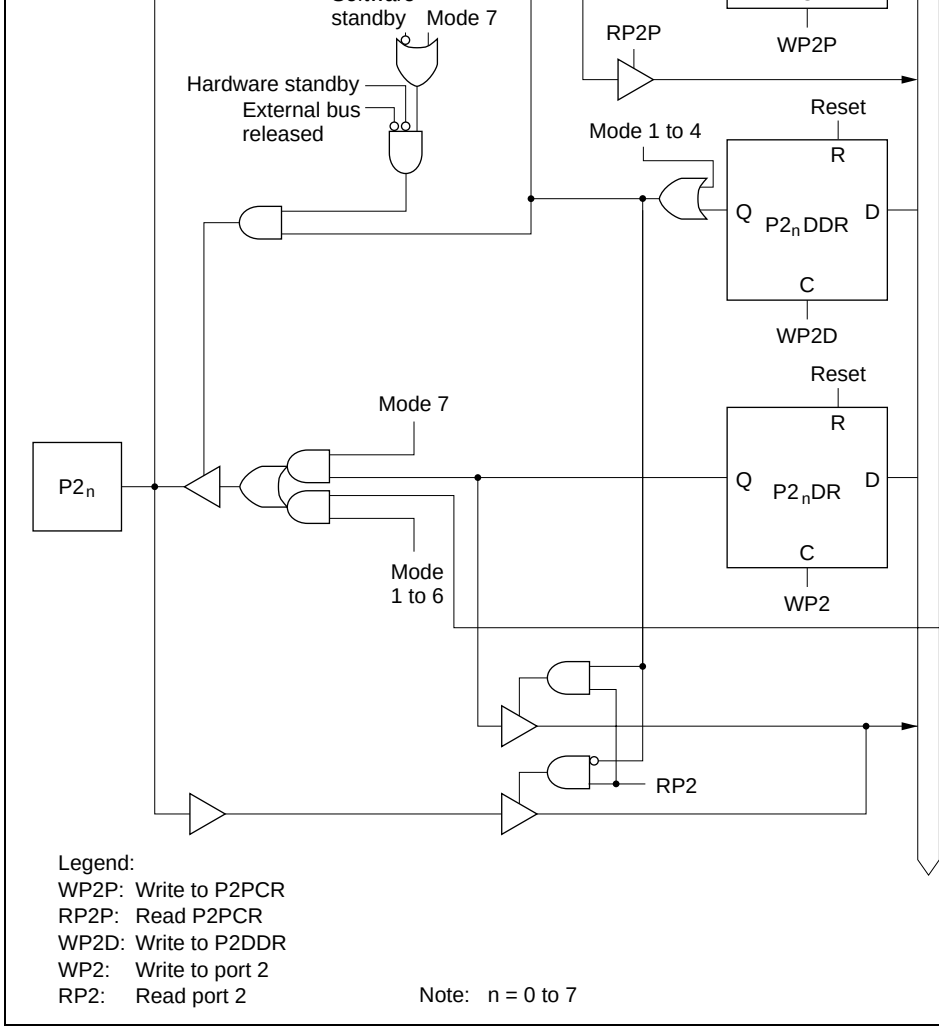


Figure C.2 Port 2 Block Diagram (Pins $P2_0$ to $P2_7$)

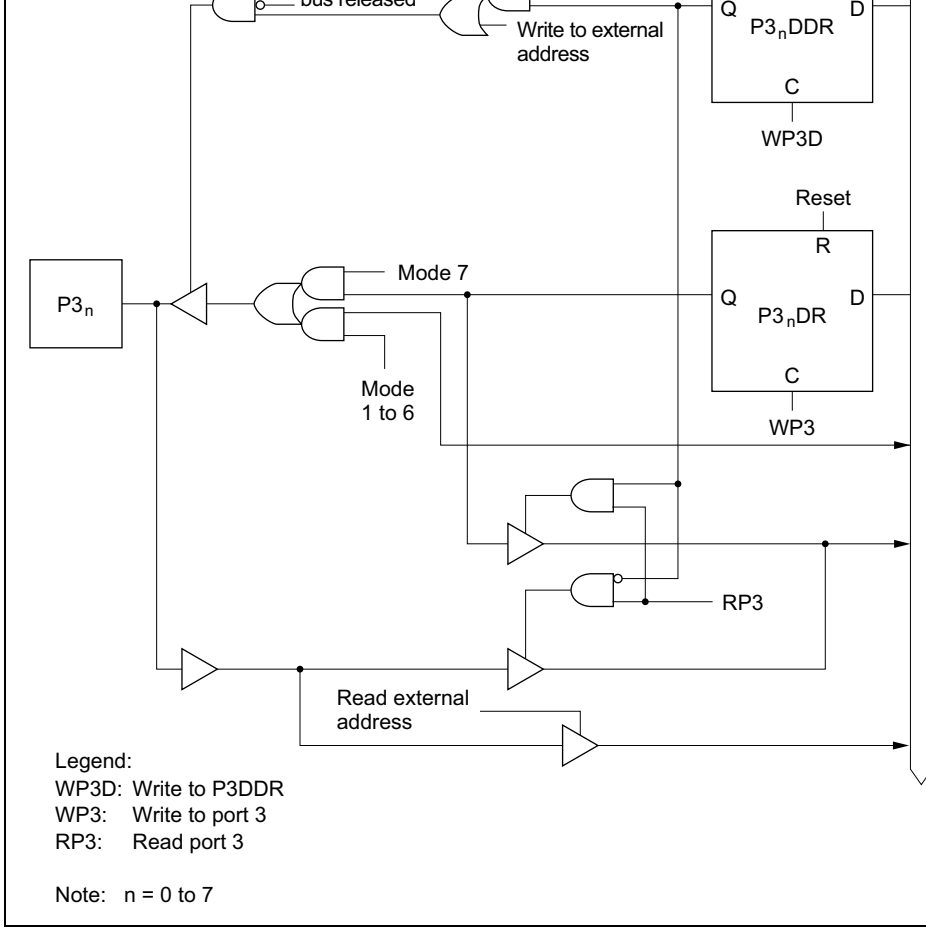


Figure C.3 Port 3 Block Diagram (Pins $P3_0$ to $P3_7$)

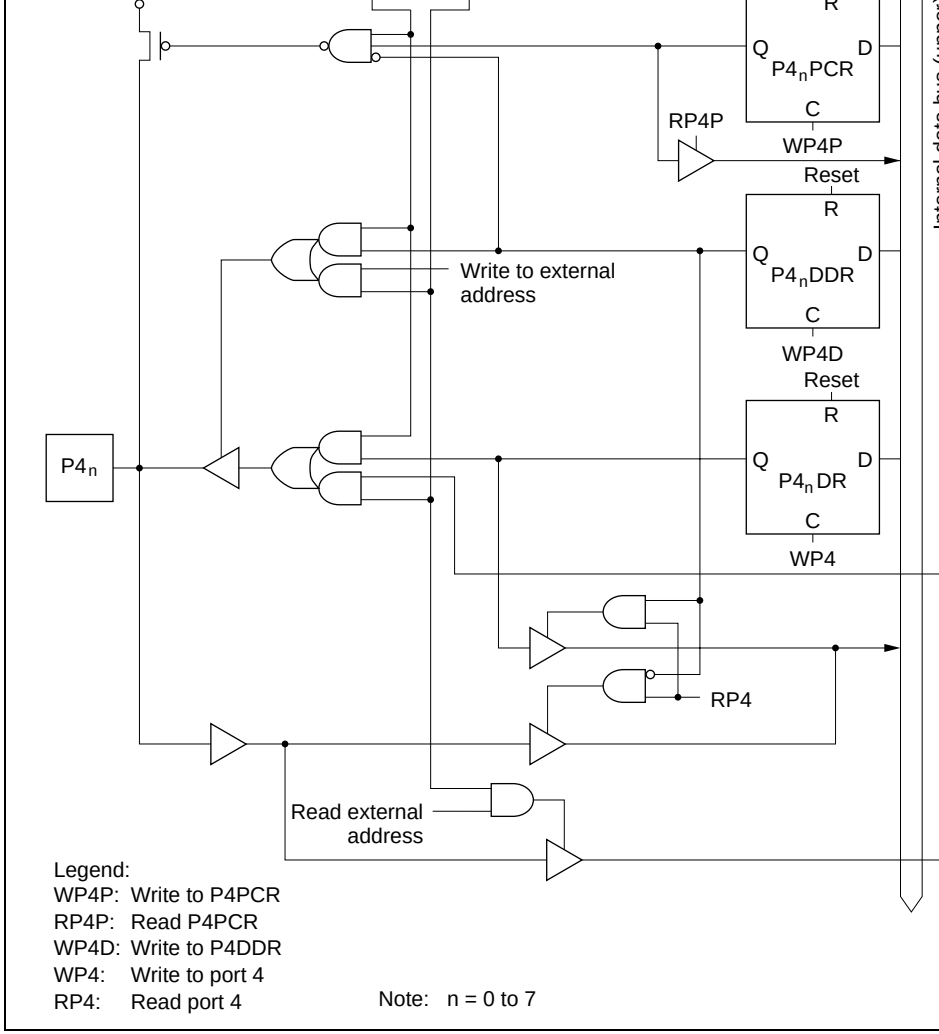
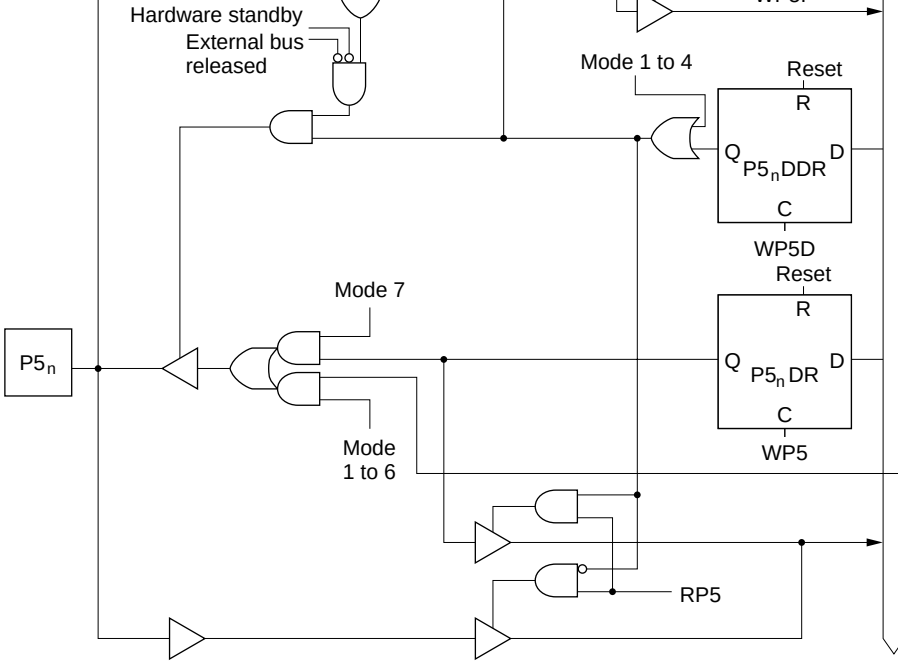


Figure C.4 Port 4 Block Diagram (Pins $P4_0$ to $P4_7$)



Legend:
WP5P: Write to P5PCR
RP5P: Read P5PCR
WP5D: Write to P5DDR
WP5: Write to port 5
RP5: Read port 5

Note: n = 0 to 3

Figure C.5 Port 5 Block Diagram (Pins $P5_0$ to $P5_3$)

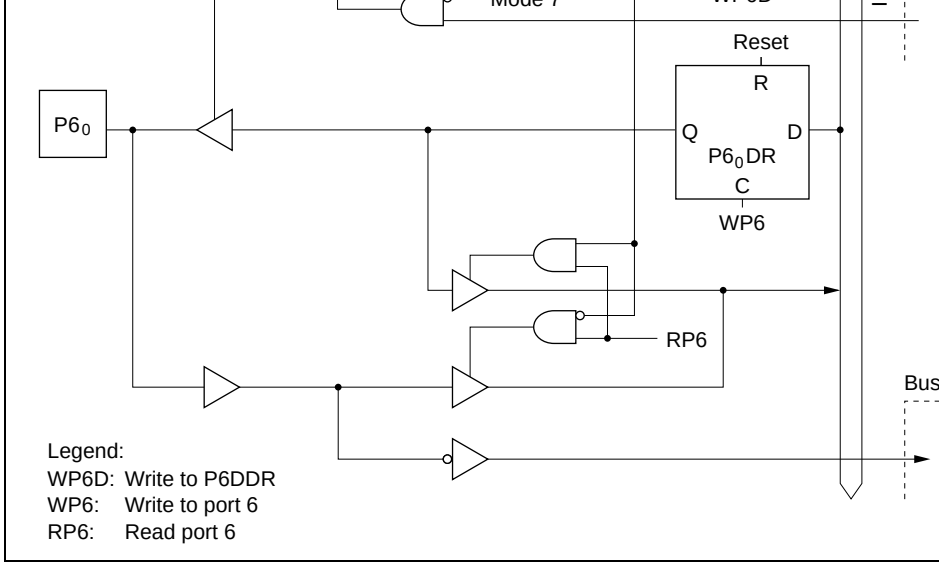


Figure C.6 (a) Port 6 Block Diagram (Pin P6$_0$)

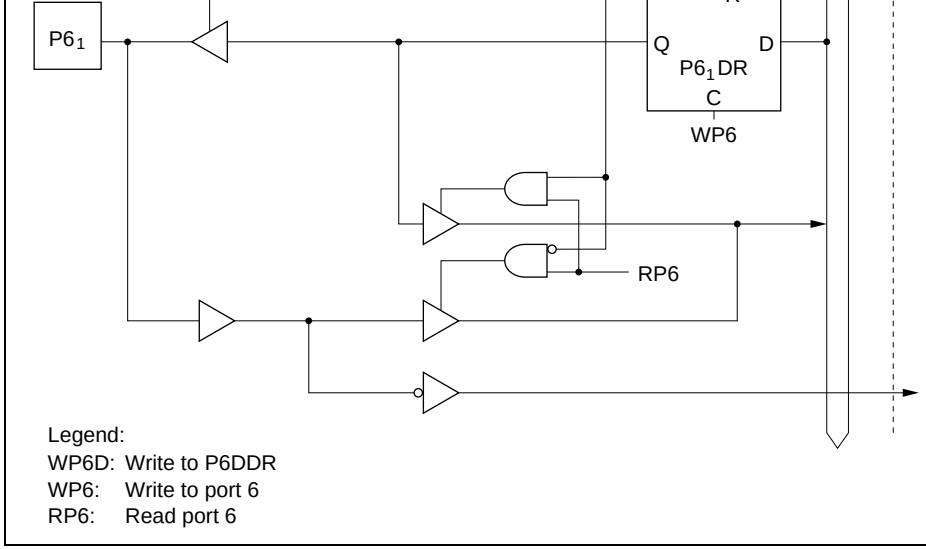


Figure C.6 (b) Port 6 Block Diagram (Pin P6$_1$)

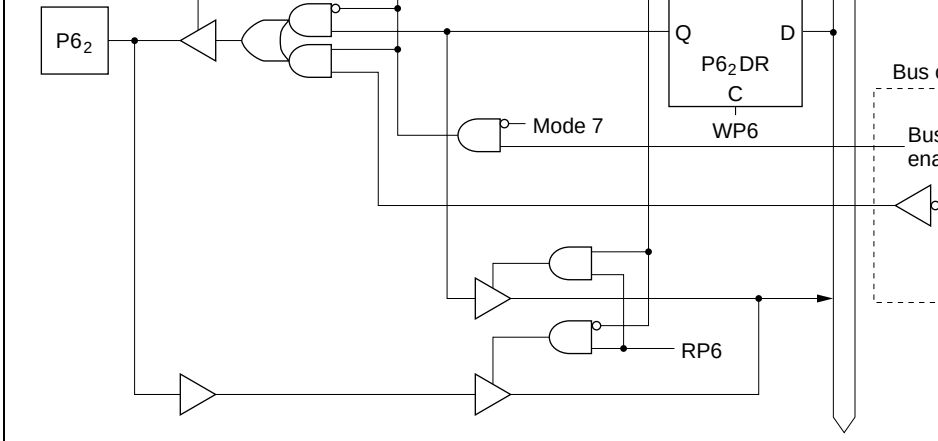
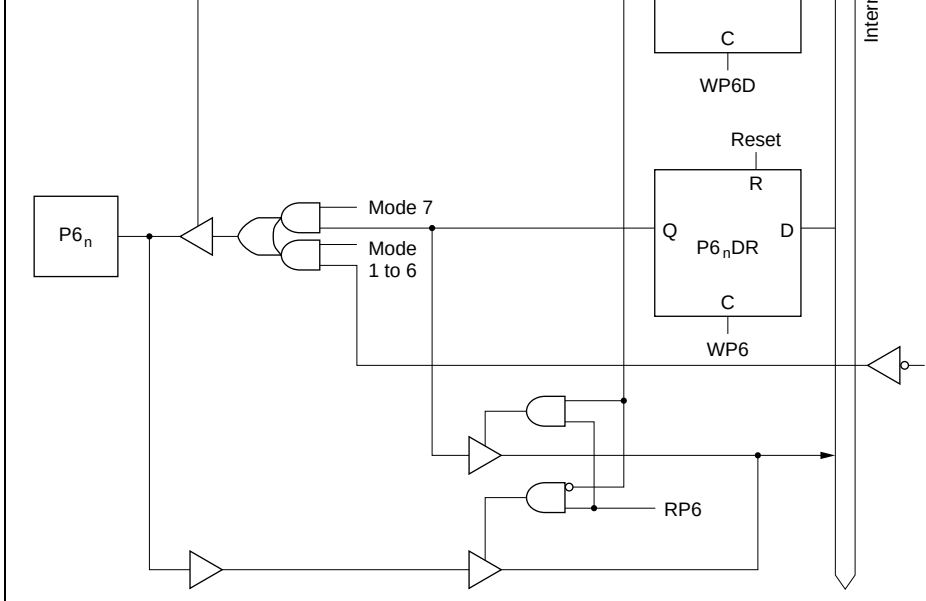


Figure C.6 (c) Port 6 Block Diagram (Pin P6$_2$)



Legend:
WP6D: Write to P6DDR
WP6: Write to port 6
RP6: Read port 6

Note: n = 6 to 3

Figure C.6 (d) Port 6 Block Diagram (Pins $P6_6$ to $P6_3$)

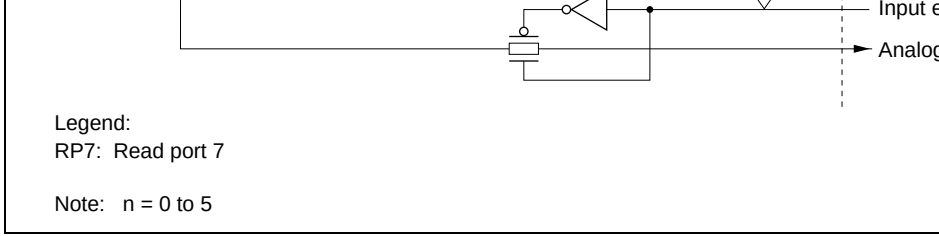


Figure C.7 (a) Port 7 Block Diagram (Pins $P7_0$ to $P7_5$)

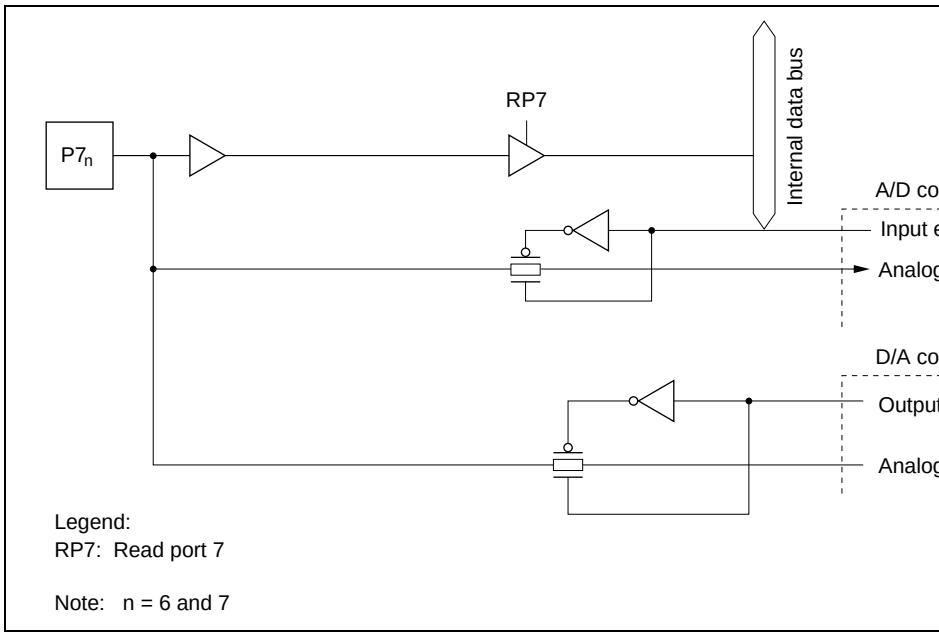


Figure C.7 (b) Port 7 Block Diagram (Pins $P7_6$ and $P7_7$)

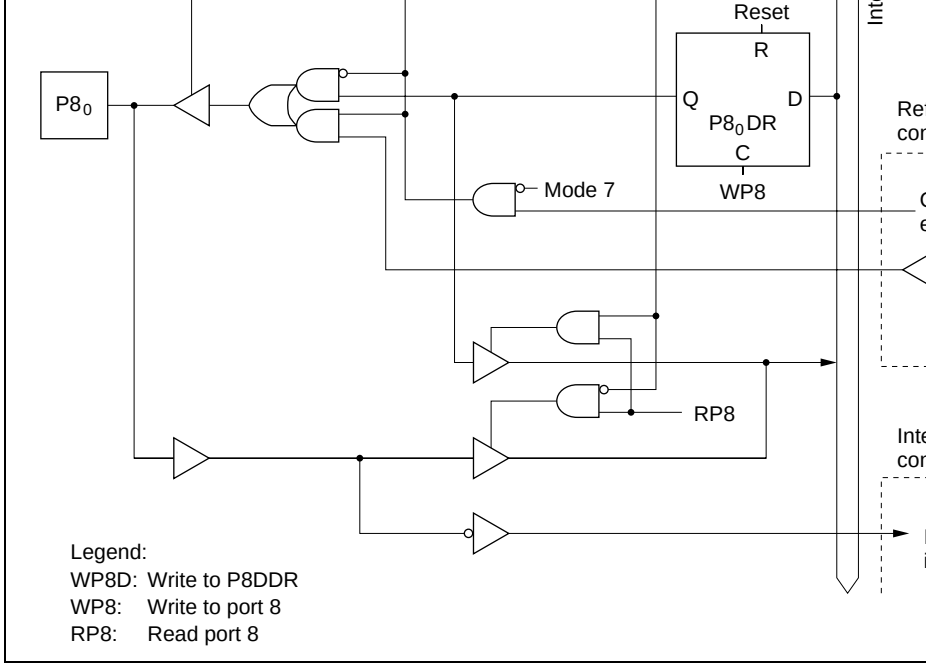


Figure C.8 (a) Port 8 Block Diagram (Pin P8$_0$)

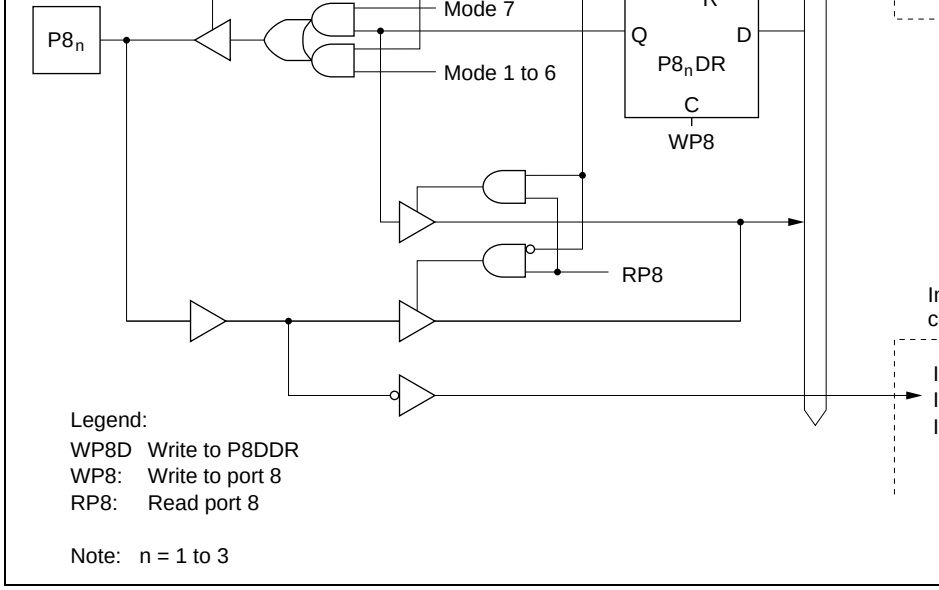


Figure C.8 (b) Port 8 Block Diagram (Pins $P8_1$ to $P8_3$)

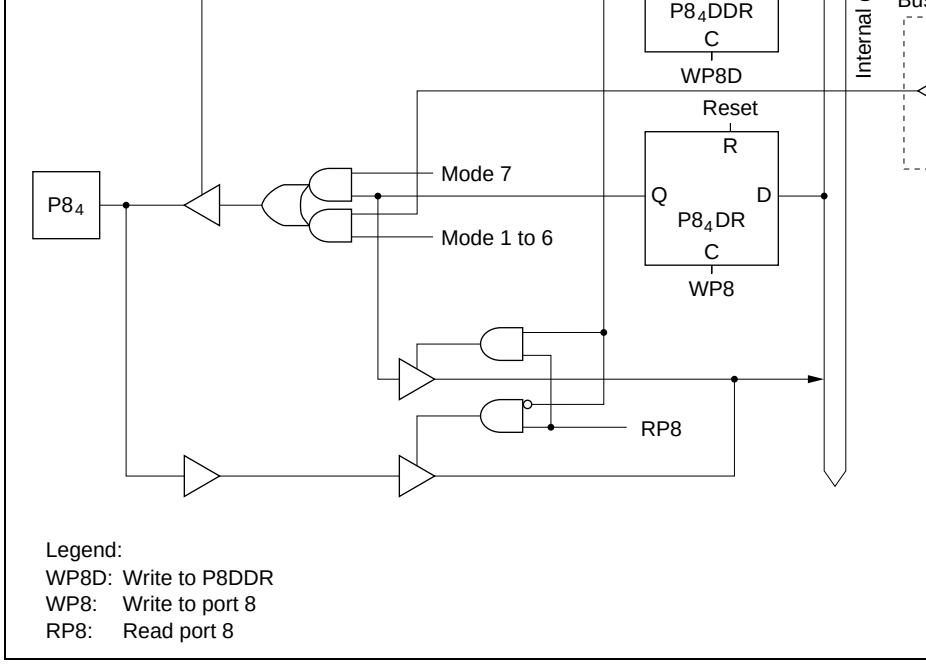


Figure C.8 (c) Port 8 Block Diagram (Pin P8$_4$)

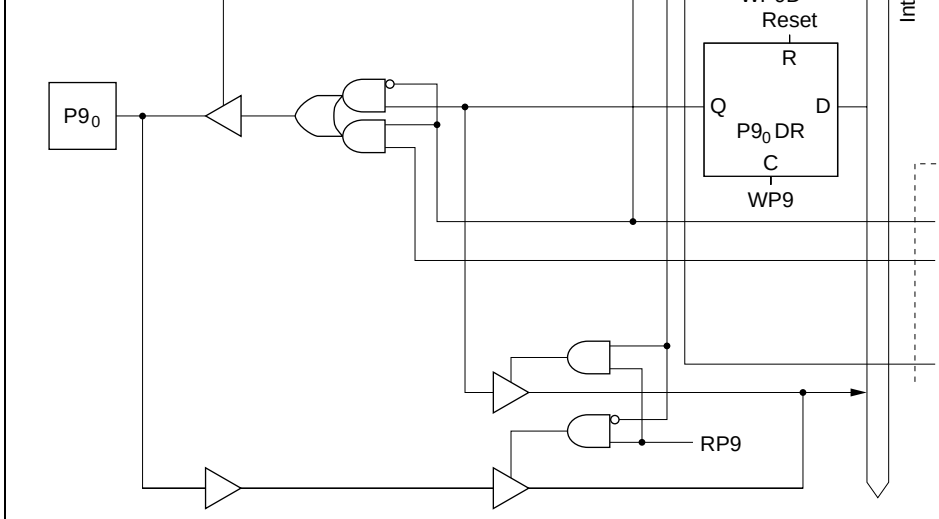
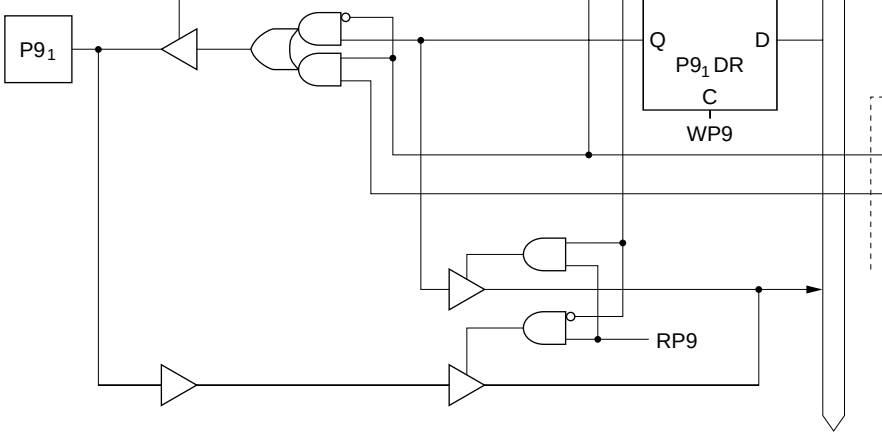
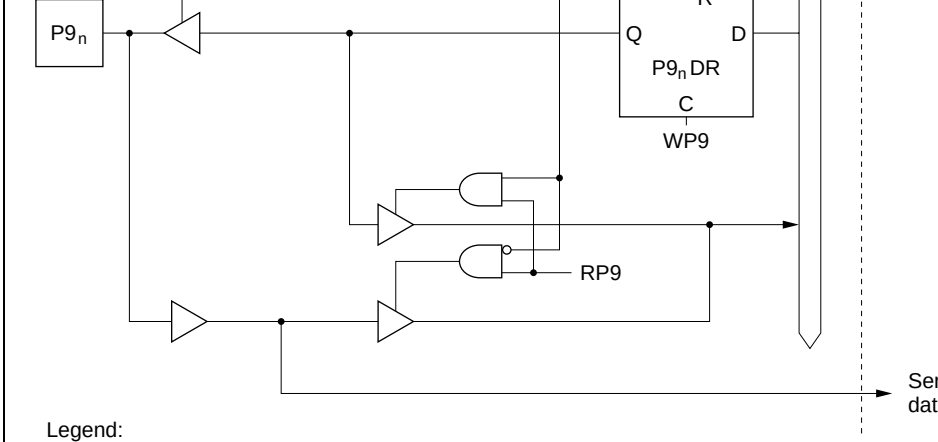


Figure C.9 (a) Port 9 Block Diagram (Pin P9$_0$)



Legend:
WP9D: Write to P9DDR
WP9: Write to port 9
RP9: Read port 9

Figure C.9 (b) Port 9 Block Diagram (Pin P9$_1$)



Legend:
WP9D: Write to P9DDR
WP9: Write to port 9
RP9: Read port 9

Note: n = 2 and 3

Figure C.9 (c) Port 9 Block Diagram (Pins $P9_2$ and $P9_3$)

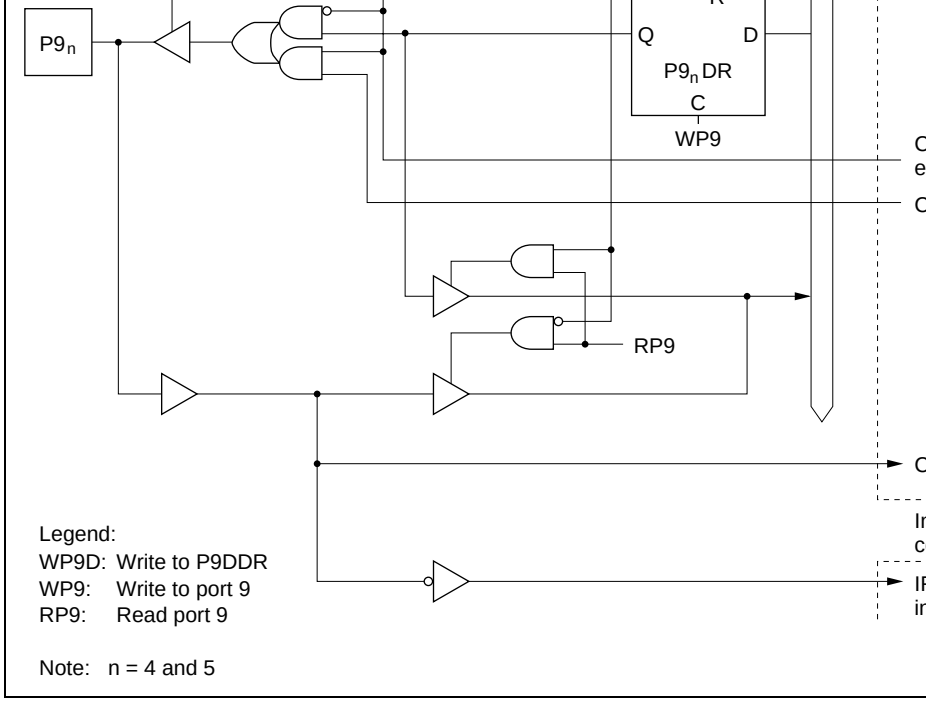
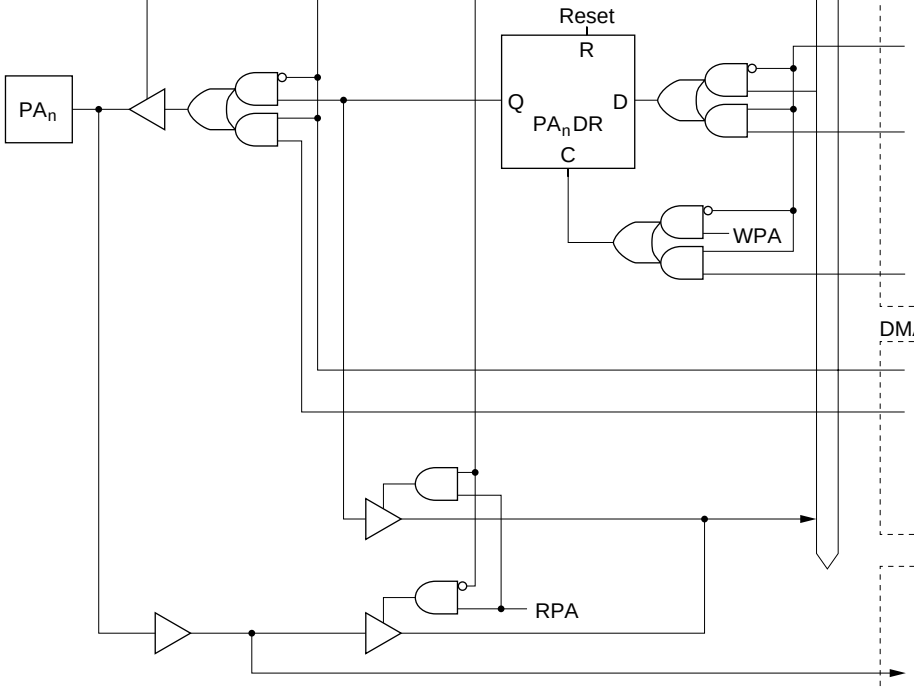


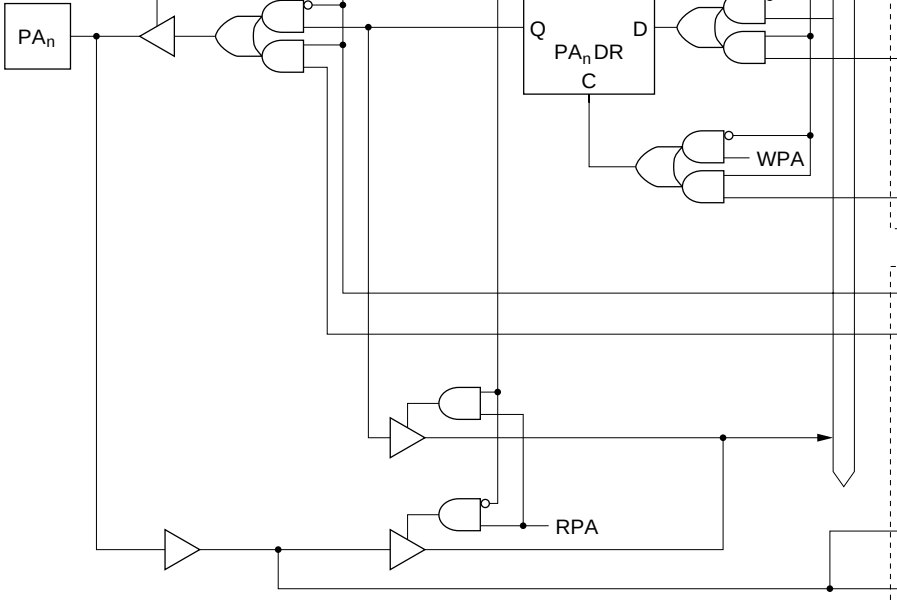
Figure C.9 (d) Port 9 Block Diagram (Pins $P9_4$ and $P9_5$)



Legend:
WPAD: Write to PADDR
WPA: Write to port A
RPA: Read port A

Note: n = 0 and 1

Figure C.10 (a) Port A Block Diagram (Pins PA_0 and PA_1)



Legend:
WPAD: Write to PADDR
WPA: Write to port A
RPA: Read port A

Note: n = 2 and 3

Figure C.10 (b) Port A Block Diagram (Pins PA_2 and PA_3)

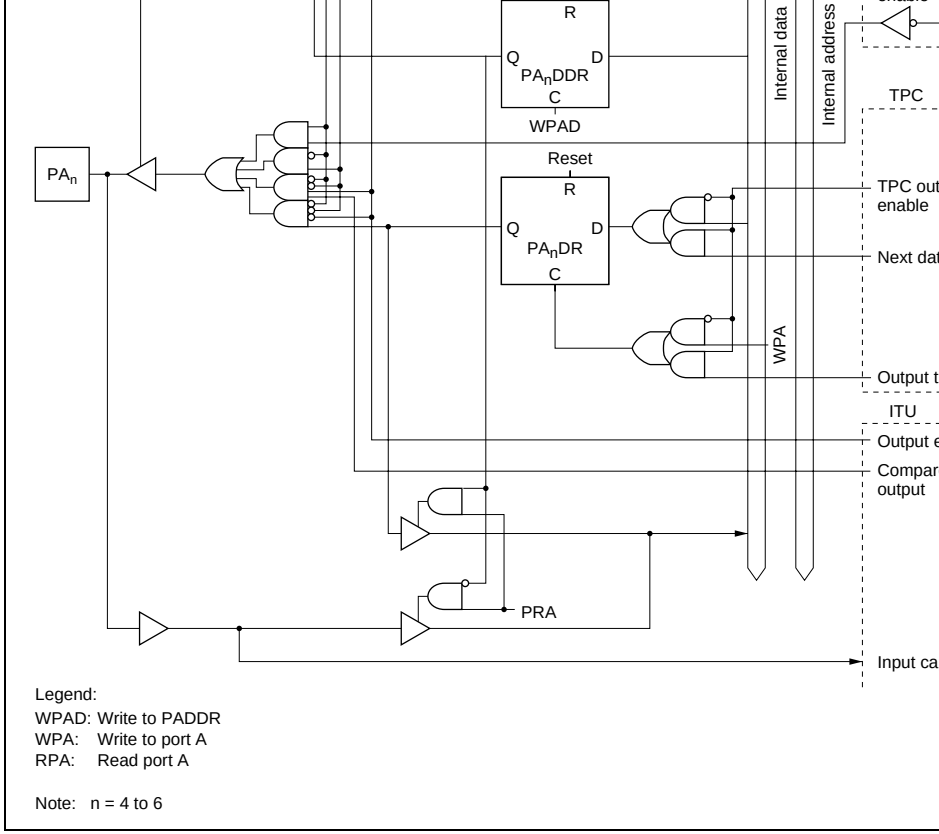


Figure C.10 (c) Port A Block Diagram (Pins PA_4 to PA_6)

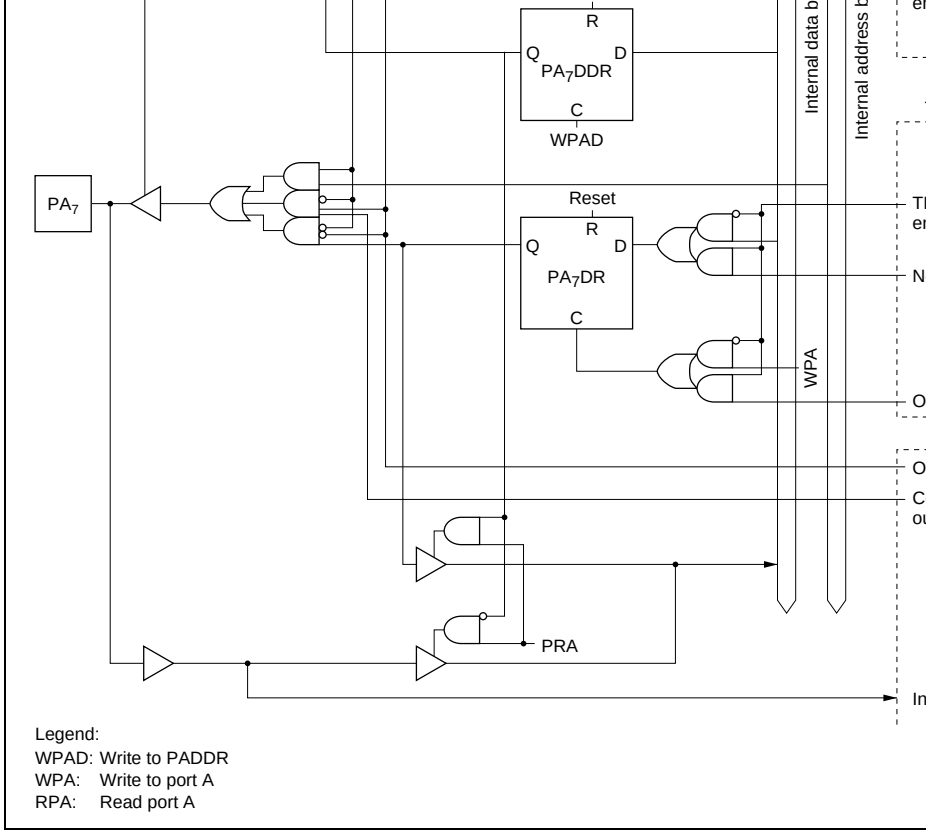


Figure C.10 (d) Port A Block Diagram (Pin PA_7)

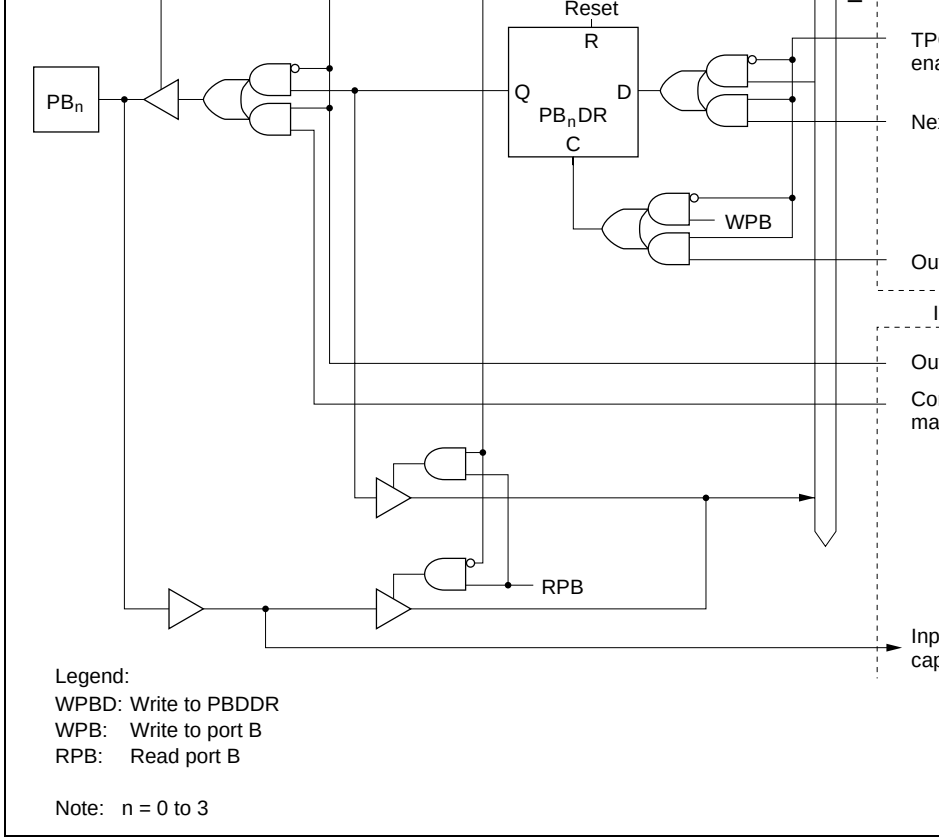
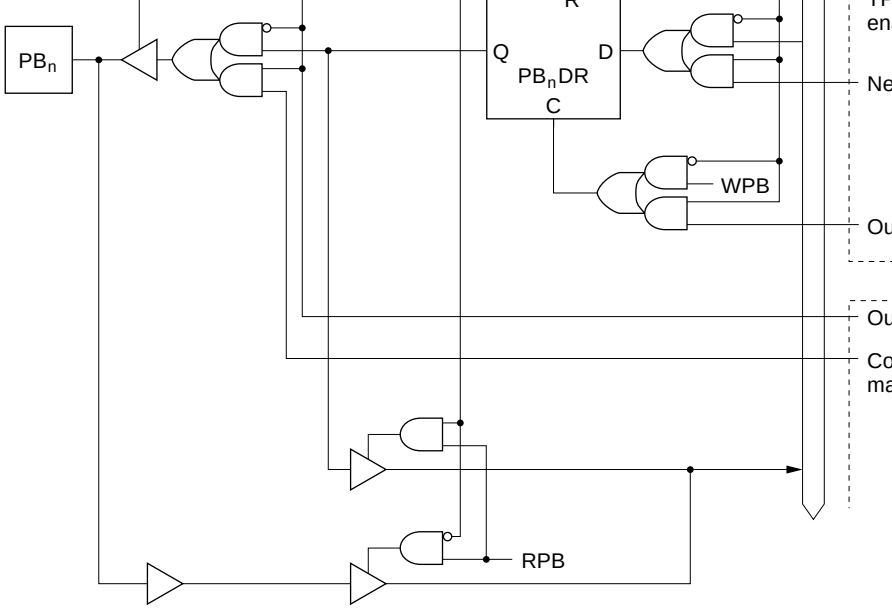


Figure C.11 (a) Port B Block Diagram (Pins PB_0 to PB_3)



Legend:
WPBD: Write to PBDDR
WPB: Write to port B
RPB: Read port B

Note: n = 4 and 5

Figure C.11 (b) Port B Block Diagram (Pins PB$_4$ and PB$_5$)

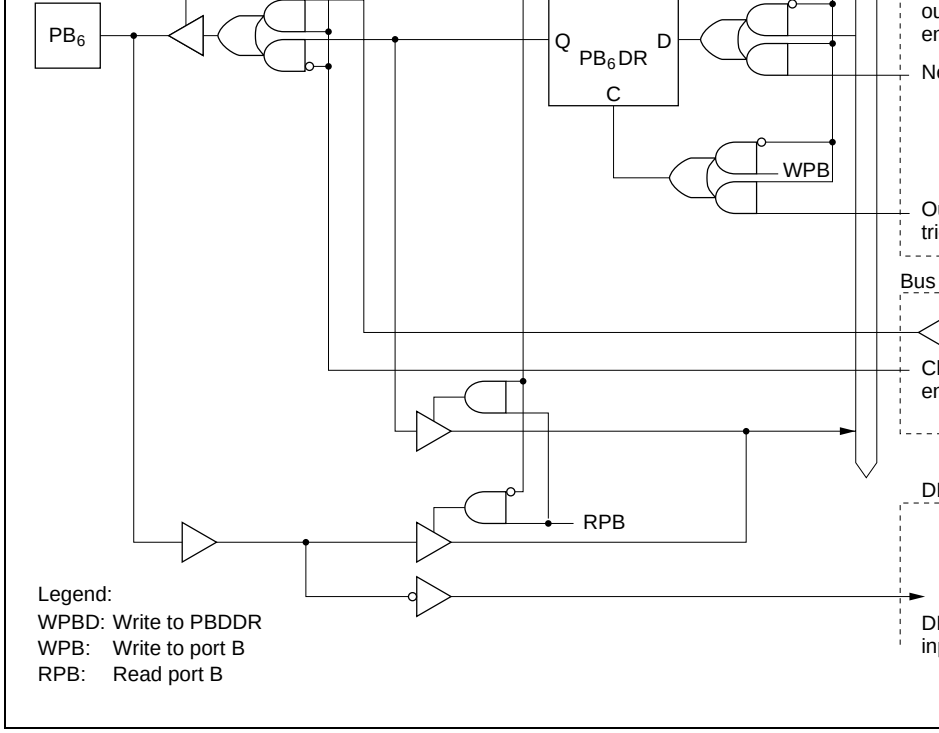


Figure C.11 (c) Port B Block Diagram (Pin PB_6)

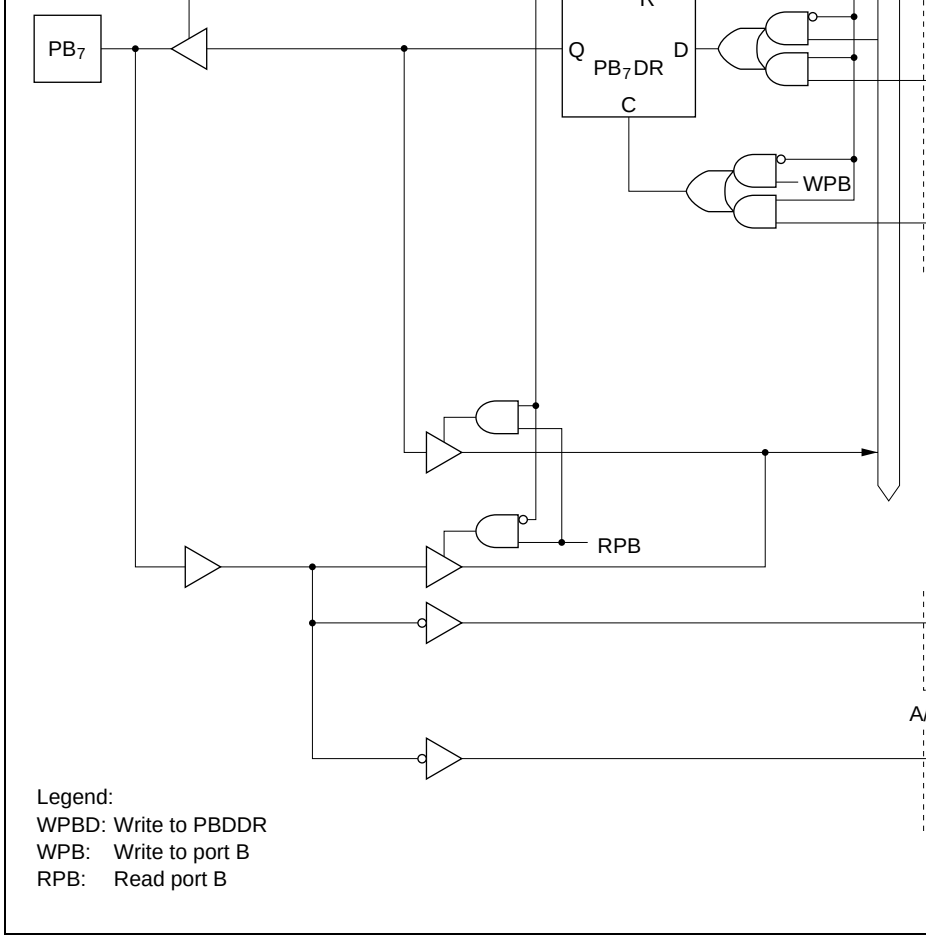


Figure C.11 (d) Port B Block Diagram (Pin PB_7)

Pin Name	Mode		Reset	Mode	Mode	Mode	S
ϕ	—		Clock output	T	H	Clock output	C
$\overline{\text{RESO}}$*2	—		T*2	T	T	T	$\overline{\text{R}}$
$P1_7$ to $P1_0$	1 to 4		L	T	T	T	A
	5, 6		T	T	keep	T	In (D
					T	T	A (D
	7		T	T	keep	—	I/
$P2_7$ to $P2_0$	1 to 4		L	T	T	T	A
	5, 6		T	T	keep	T	In (D
					T	T	A (D
	7		T	T	keep	—	I/
$P3_7$ to $P3_0$	1 to 6		T	T	T	T	D
	7		T	T	keep	—	I/
$P4_7$ to $P4_0$	1 to 6	8-bit bus	T	T	keep	keep	I/
		16-bit bus	T	T	T	T	D
	7		T	T	keep	—	I/
$P5_3$ to $P5_0$	1 to 4		L	T	T	T	A
	5, 6		T	T	keep	T	In (D
					T	T	A (D
	7		T	T	keep	—	I/

				T (BRLE = 1)		
	7	T	T	keep	—	
$P6_2$	1 to 6	T	T	keep (BRLE = 0) H (BRLE = 1)	L	
	7	T	T	keep	—	
$P6_6$ to $P6_3$	1 to 6	H[*3]	T	T	T	
	7	T	T	keep	—	
$P7_7$ to $P7_0$	1 to 7	T	T	T	T[*1]	
$P8_0$	1 to 6	T	T	keep (RFSHE = 0) $\overline{\text{RFSH}}$ (RFSHE = 1)	keep (RFSHE = 0) H (RFSHE = 1)	
	7	T	T	keep	—	
$P8_3$ to $P8_1$	1 to 6	T	T	T (DDR = 0) H (DDR = 1)	keep (DDR = 0) H (DDR = 1)	
	7	T	T	keep	—	
$P8_4$	1 to 6	L	T	T (DDR = 0) L (DDR = 1)	keep (DDR = 0) H (DDR = 1)	
	7	T	T	keep	—	

				output)	output)	(a
				keep (otherwise)	keep (otherwise)	o I/ (c
	1, 2, 5, 7	T*4	T	keep	keep*1	I/
PA_7	3, 4, 6	L*4	T	T	T	A
	1, 2, 5, 7	T*4	T	keep	keep*1	I/
PB_7, PB_5 to PB_0	1 to 7	T	T	keep	keep*1	I/
PB_6	3, 4, 6	T	T	H ($\overline{CS}$ output)	H ($\overline{CS}$ output)	$\overline{C}$ ($\overline{C}$
				keep (otherwise)	keep (otherwise)	I/ (c
	1, 2, 5, 7	T	T	keep	keep*1	I/

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 bit

Notes: 1. The bus cannot be released in mode 7.

2. Output is low only for reset by WDT overflow.
   This $\overline{RESO}$ output function is only for the mask ROM, ZTAT, and flash memo
   power supply).

3. During direct power supply, oscillation damping time is “H” or “T”.

4. During direct power supply, oscillation damping time differs between “H”, “L”

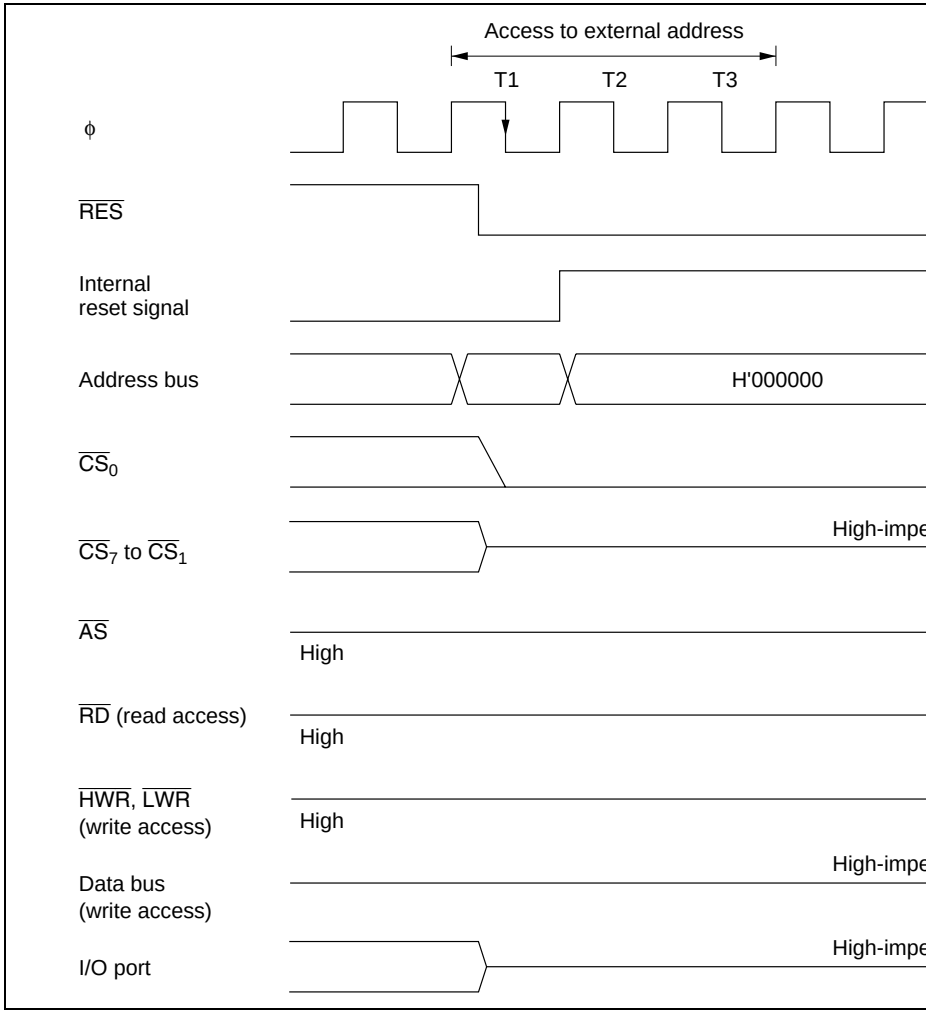


Figure D.1 Reset during Memory Access (Reset during T1 State)

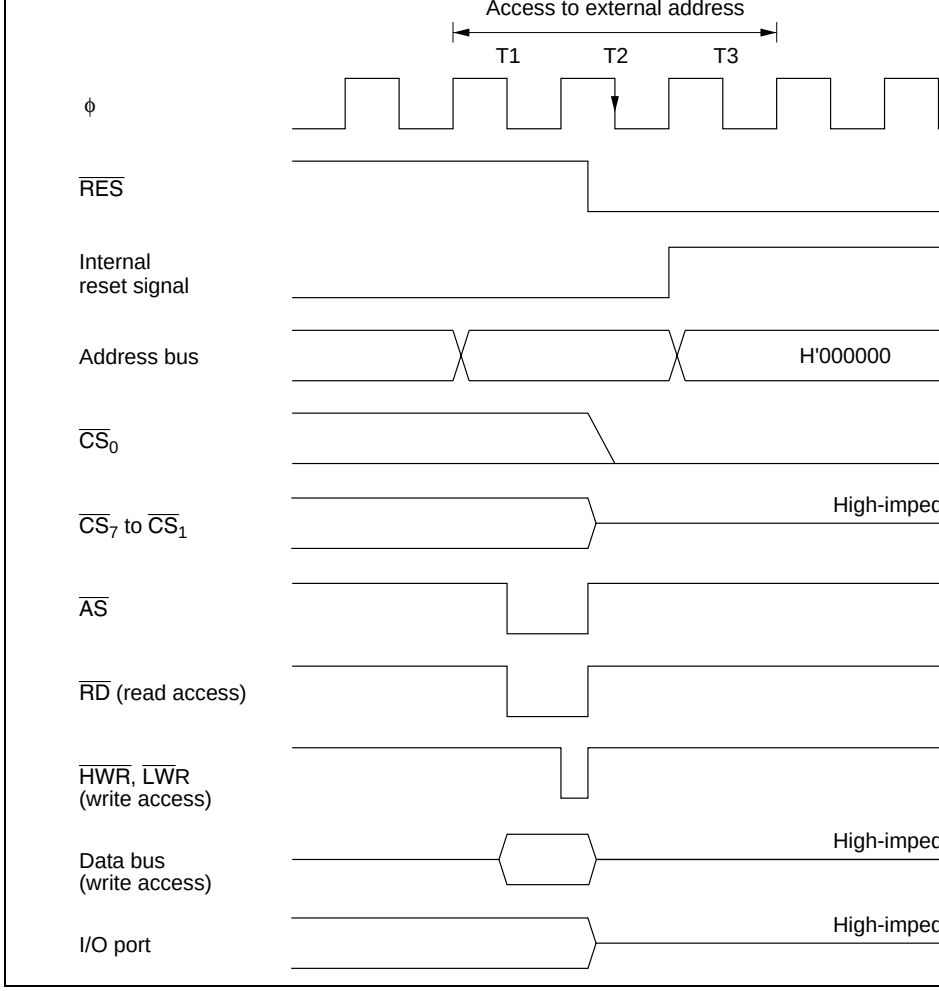


Figure D.2 Reset during Memory Access (Reset during T2 State)

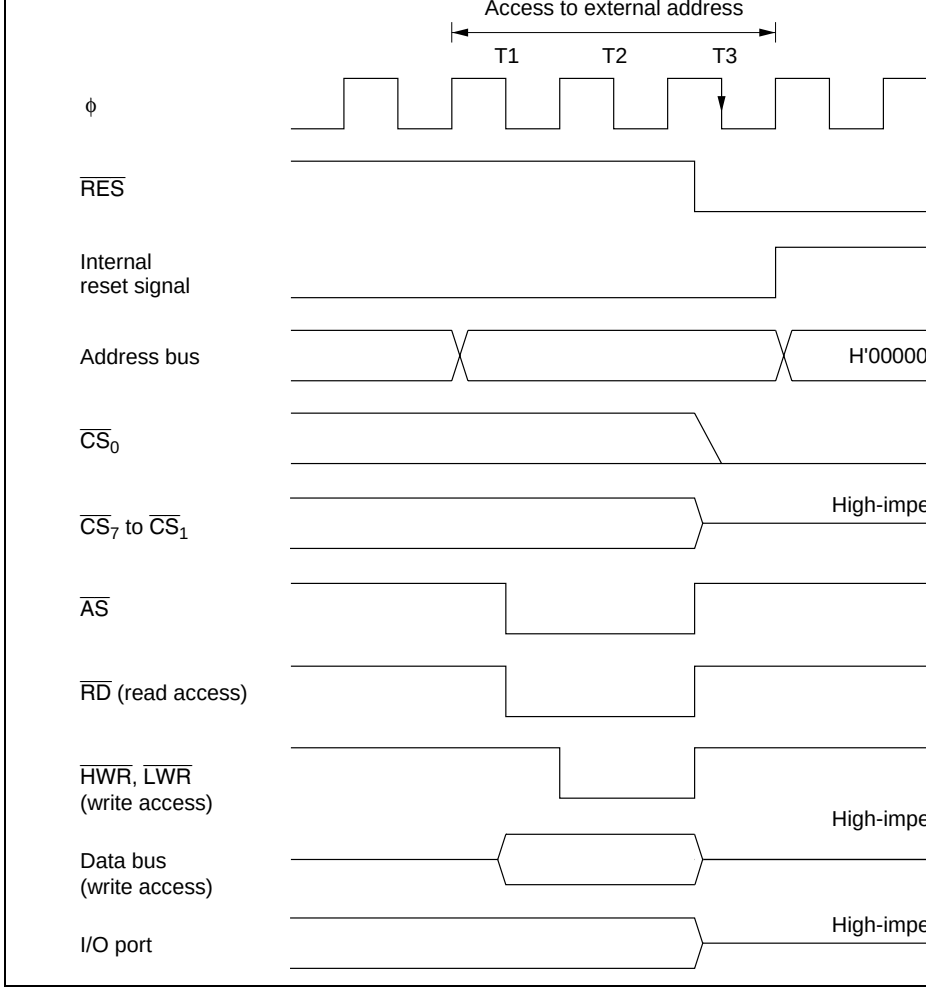
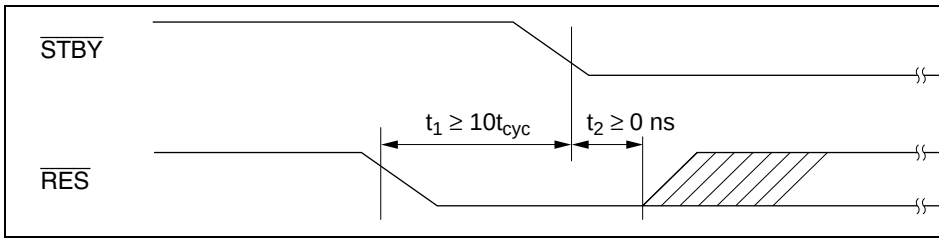


Figure D.3 Reset during Memory Access (Reset during T3 State)

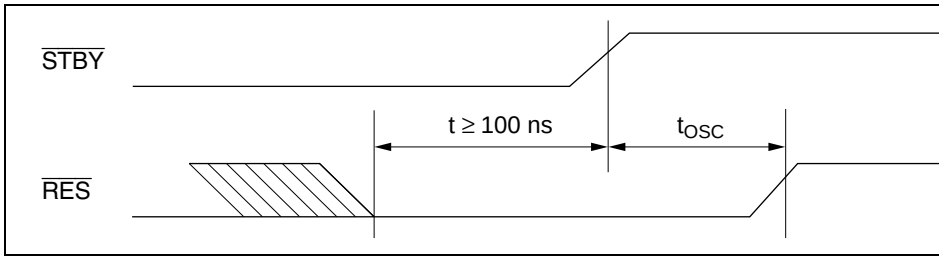
until STBY goes low (minimum delay from STBY low to RES high: 0 ns).



(2) To retain RAM contents with the RAME bit cleared to 0 in SYSCR, or when RAM do not need to be retained, $\overline{\text{RES}}$ does not have to be driven low as in (1).

Timing of Recovery from Hardware Standby Mode

Drive the $\overline{\text{RES}}$ signal low approximately 100 ns before $\overline{\text{STBY}}$ goes high.



	ROM version		HD6433048BF		100-pin QFP
		3 V version	HD6433048BVTE		100-pin TQP
			HD6433048BVF		100-pin QFP
H8/3048 F-ONE	Flash memory version (single power supply)	5 V version	HD64F3048BTE	64F3048TE	100-pin TQP
			HD64F3048BF	64F3048F	100-pin QFP
		3 V version	HD64F3048BVTE	64F3048VTE	100-pin TQP
			HD64F3048BVF	64F3048VF	100-pin QFP
H8/3048F	Flash memory version (dual power supply)	5 V version	HD64F3048TF	HD64F3048TF	100-pin TQP
			HD64F3048F	HD64F3048F	100-pin QFP
		3 V version	HD64F3048VTF	HD64F3048VTF	100-pin TQP
			HD64F3048VF	HD64F3048VF	100-pin QFP
H8/3048 ZTAT	PROM version	5 V version	HD6473048TF	HD6473048TF	100-pin TQP
			HD6473048F	HD6473048F	100-pin QFP
		3 V version	HD6473048VTF	HD6473048VTF	100-pin TQP
			HD6473048VF	HD6473048VF	100-pin QFP
H8/3048	Mask ROM version	5 V version	HD6433048TF	HD6433048(***)TF	100-pin TQP
			HD6433048F	HD6433048(***)F	100-pin QFP
		3 V version	HD6433048VTF	HD6433048(***)VTF	100-pin TQP
			HD6433048VF	HD6433048(***)VF	100-pin QFP
H8/3047	Mask ROM version	5 V version	HD6433047TF	HD6433047(***)TF	100-pin TQP
			HD6433047F	HD6433047(***)F	100-pin QFP
		3 V version	HD6433047VTF	HD6433047(***)VTF	100-pin TQP
			HD6433047VF	HD6433047(***)VF	100-pin QFP

	ROM version		HD6433044F	HD6433044(***)F	100-pin QFP (
		3 V version	HD6433044VTF	HD6433044(***)VTF	100-pin TQFP
			HD6433044VF	HD6433044(***)VF	100-pin QFP (

Note: (***) in mask ROM versions is the ROM code.

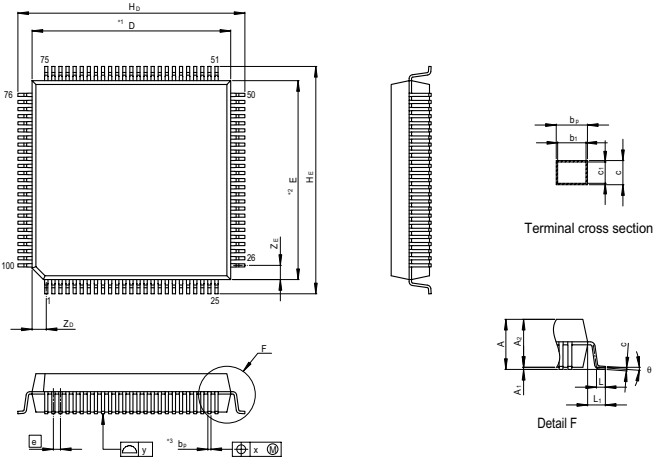
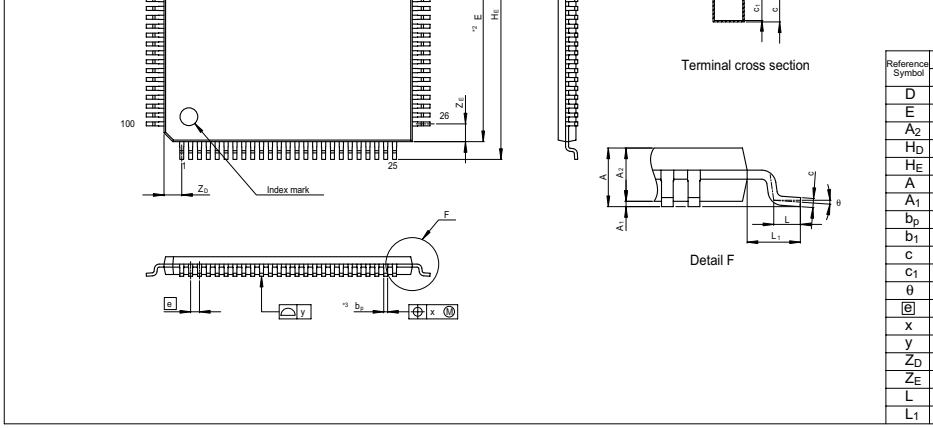


Figure G.1 Package Dimensions (FP-100B)



Reference Symbol
D
E
A_2
H_D
H_E
A
A_1
b_p
b_1
c
c_1
θ
e
x
y
Z_D
Z_E
L
L_1

Figure G.2 Package Dimensions (TFP-100B)

Renesas 8-Bit Single-Chip Microcomputer
Hardware Manual
H8/3048B Group

Publication Date: 1st Edition, August 2002
Rev.3.00, September 27, 2006
Published by: Sales Strategic Planning Div.
Renesas Technology Corp.
Edited by: Customer Support Department
Global Strategic Communication Div.
Renesas Solutions Corp.

RENESAS SALES OFFICES

Refer to "**http://www.renesas.com/en/network**" for the latest and detailed information.

Renesas Technology America, Inc.
450 Holger Way, San Jose, CA 95134-1368, U.S.A
Tel: <1> (408) 382-7500, Fax: <1> (408) 382-7501

Renesas Technology Europe Limited
Dukes Meadow, Millboard Road, Bourne End, Buckinghamshire, SL8 5FH, U.K.
Tel: <44> (1628) 585-100, Fax: <44> (1628) 585-900

Renesas Technology (Shanghai) Co., Ltd.
Unit 204, 205, AZIACenter, No.1233 Lujiazui Ring Rd, Pudong District, Shanghai, China 200120
Tel: <86> (21) 5877-1818, Fax: <86> (21) 6887-7898

Renesas Technology Hong Kong Ltd.
7th Floor, North Tower, World Finance Centre, Harbour City, 1 Canton Road, Tsimshatsui, Kowloon, Hong Kong
Tel: <852> 2265-6688, Fax: <852> 2730-6071

Renesas Technology Taiwan Co., Ltd.
10th Floor, No.99, Fushing North Road, Taipei, Taiwan
Tel: <886> (2) 2715-2888, Fax: <886> (2) 2713-2999

Renesas Technology Singapore Pte. Ltd.
1 Harbour Front Avenue, #06-10, Keppel Bay Tower, Singapore 098632
Tel: <65> 6213-0200, Fax: <65> 6278-8001

Renesas Technology Korea Co., Ltd.
Kukje Center Bldg. 18th Fl., 191, 2-ka, Hangang-ro, Yongsan-ku, Seoul 140-702, Korea
Tel: <82> (2) 796-3115, Fax: <82> (2) 796-2145

Renesas Technology Malaysia Sdn. Bhd
Unit 906, Block B, Menara Amcorp, Amcorp Trade Centre, No.18, Jalan Persiaran Barat, 46050 Petaling Jaya, Selangor Darul Ehs
Tel: <603> 7955-9390, Fax: <603> 7955-9510

H8/3048B Group
Hardware Manual

Renesas Electronics Corporation
1753, Shimonumabe, Nakahara-ku, Kawasaki-shi, Kanagawa 211-8668 Japan

REJ09B0325-0300

X-ON Electronics

Largest Supplier of Electrical and Electronic Components

Click to view similar products for Microprocessors - MPU ***category:***

Click to view products by Renesas ***manufacturer:***

Other Similar products are found below :

MC68302EH20C MC7457RX1000LC MC7457RX1267LC MC7457VG1267LC A2C00010998 A A2C52004004 R5F117BCGNA#20 R5F52106BDLA#U0 R5S72690W266BG#U0 ADJ3400IAA5DOE MPC8245TVV266D MPC8245TZU300D MPC8260ACVVMHBB MPC8323ECVRAFDCA MPC8536ECVJAVLA BOXNUC5PGYH0AJ 20-668-0024 P1010NSN5DFB P2010NSN2MHC P2020NXE2HHC P5020NSE7QMB P5020NSE7TNB P5020NSE7VNB LS1020ASN7KQB LS1020AXN7HNB LS1020AXN7KQB A2C00010729 A A2C00039344 T1022NSE7MQB T1022NXN7PQB T1023NSE7MQA T1024NXE7PQA T1042NSE7MQB T1042NSN7MQB T1042NXN7WQB T2080NSE8TTB T2080NSN8PTB T2080NXE8TTB T2081NXN8TTB R5F101AFASP#V0 MC68302CEH20C TS68040MF33A MPC8260ACVVMIBB MPC8280CZUUPEA MPC8313ECVRAFFC MPC8313ECVRAGDC MPC8313EVRADDC MPC8313EVRAFFC MPC8313VRADDC MPC8314CVRAGDA