To our customers,

Old Company Name in Catalogs and Other Documents

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

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

April 1st, 2010
Renesas Electronics Corporation

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

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

Notice

1. All information included in this document is current as of the date this document is issued. Such information, however, is subject to change without any prior notice. Before purchasing or using any Renesas Electronics products listed herein, please confirm the latest product information with a Renesas Electronics sales office. Also, please pay regular and careful attention to additional and different information to be disclosed by Renesas Electronics such as that disclosed through our website.
2. Renesas Electronics does not assume any liability for infringement of patents, copyrights, or other intellectual property rights of third parties by or arising from the use of Renesas Electronics products or technical information described in this document. No license, express, implied or otherwise, is granted hereby under any patents, copyrights or other intellectual property rights of Renesas Electronics or others.
3. You should not alter, modify, copy, or otherwise misappropriate any Renesas Electronics product, whether in whole or in part.
4. Descriptions of circuits, software and other related information in this document are provided only to illustrate the operation of semiconductor products and application examples. You are fully responsible for the incorporation of these circuits, software, and information in the design of your equipment. Renesas Electronics assumes no responsibility for any losses incurred by you or third parties arising from the use of these circuits, software, or information.
5. When exporting the products or technology described in this document, you should comply with the applicable export control laws and regulations and follow the procedures required by such laws and regulations. You should not use Renesas Electronics products or the technology described in this document for any purpose relating to military applications or use by the military, including but not limited to the development of weapons of mass destruction. Renesas Electronics products and technology may not be used for or incorporated into any products or systems whose manufacture, use, or sale is prohibited under any applicable domestic or foreign laws or regulations.
6. Renesas Electronics has used reasonable care in preparing the information included in this document, but Renesas Electronics does not warrant that such information is error free. Renesas Electronics assumes no liability whatsoever for any damages incurred by you resulting from errors in or omissions from the information included herein.
7. Renesas Electronics products are classified according to the following three quality grades: "Standard", "High Quality", and "Specific". The recommended applications for each Renesas Electronics product depends on the product's quality grade, as indicated below. You must check the quality grade of each Renesas Electronics product before using it in a particular application. You may not use any Renesas Electronics product for any application categorized as "Specific" without the prior written consent of Renesas Electronics. Further, you may not use any Renesas Electronics product for any application for which it is not intended without the prior written consent of Renesas Electronics. Renesas Electronics shall not be in any way liable for any damages or losses incurred by you or third parties arising from the use of any Renesas Electronics product for an application categorized as "Specific" or for which the product is not intended where you have failed to obtain the prior written consent of Renesas Electronics. The quality grade of each Renesas Electronics product is "Standard" unless otherwise expressly specified in a Renesas Electronics data sheets or data books, etc.

   "Standard": Computers; office equipment; communications equipment; test and measurement equipment; audio and visual equipment; home electronic appliances; machine tools; personal electronic equipment; and industrial robots.

   "High Quality": Transportation equipment (automobiles, trains, ships, etc.); traffic control systems; anti-disaster systems; anti-crime systems; safety equipment; and medical equipment not specifically designed for life support.

   "Specific": Aircraft; aerospace equipment; submersible repeaters; nuclear reactor control systems; medical equipment or systems for life support (e.g. artificial life support devices or systems), surgical implantations, or healthcare intervention (e.g. excision, etc.), and any other applications or purposes that pose a direct threat to human life.
8. You should use the Renesas Electronics products described in this document within the range specified by Renesas Electronics, especially with respect to the maximum rating, operating supply voltage range, movement power voltage range, heat radiation characteristics, installation and other product characteristics. Renesas Electronics shall have no liability for malfunctions or damages arising out of the use of Renesas Electronics products beyond such specified ranges.
9. Although Renesas Electronics endeavors to improve the quality and reliability of its products, semiconductor products have specific characteristics such as the occurrence of failure at a certain rate and malfunctions under certain use conditions. Further, Renesas Electronics products are not subject to radiation resistance design. Please be sure to implement safety measures to guard them against the possibility of physical injury, and injury or damage caused by fire in the event of the failure of a Renesas Electronics product, such as safety design for hardware and software including but not limited to redundancy, fire control and malfunction prevention, appropriate treatment for aging degradation or any other appropriate measures. Because the evaluation of microcomputer software alone is very difficult, please evaluate the safety of the final products or system manufactured by you.
10. Please contact a Renesas Electronics sales office for details as to environmental matters such as the environmental compatibility of each Renesas Electronics product. Please use Renesas Electronics products in compliance with all applicable laws and regulations that regulate the inclusion or use of controlled substances, including without limitation, the EU RoHS Directive. Renesas Electronics assumes no liability for damages or losses occurring as a result of your noncompliance with applicable laws and regulations.
11. This document may not be reproduced or duplicated, in any form, in whole or in part, without prior written consent of Renesas Electronics.
12. Please contact a Renesas Electronics sales office if you have any questions regarding the information contained in this document or Renesas Electronics products, or if you have any other inquiries.

(Note 1) "Renesas Electronics" as used in this document means Renesas Electronics Corporation and also includes its majority-owned subsidiaries.

(Note 2) "Renesas Electronics product(s)" means any product developed or manufactured by or for Renesas Electronics.

User's Manual

16

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

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

H8/3039 Group, H8/3039F-ZTAT™

Hardware Manual

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

H8/3039	HD64F3039F	H8/3037	HD6433037F
	HD64F3039TE		HD6433037TE
	HD64F3039VF		HD6433037VF
	HD64F3039VTE		HD6433037VTE
	HD6433039F	H8/3036	HD6433036F
	HD6433039TE		HD6433036TE
	HD6433039VF		HD6433036VF
	HD6433039VTE		HD6433036VTE
H8/3038	HD6433038F		
	HD6433038TE		
	HD6433038VF		
	HD6433038VTE		

Renesas Electronics
www.renesas.com

Rev.3.00 2007.03

Notes regarding these materials

1. This document is provided for reference purposes only so that Renesas customers may select the appropriate Renesas products for their use. Renesas neither makes warranties or representations with respect to the accuracy or completeness of the information contained in this document nor grants any license to any intellectual property rights or any other rights of Renesas or any third party with respect to the information in this document.
2. Renesas shall have no liability for damages or infringement of any intellectual property or other rights arising out of the use of any information in this document, including, but not limited to, product data, diagrams, charts, programs, algorithms, and application circuit examples.
3. You should not use the products or the technology described in this document for the purpose of military applications such as the development of weapons of mass destruction or for the purpose of any other military use. When exporting the products or technology described herein, you should follow the applicable export control laws and regulations, and procedures required by such laws and regulations.
4. All information included in this document such as product data, diagrams, charts, programs, algorithms, and application circuit examples, 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 products listed in this document, please confirm the latest product information with a Renesas sales office. Also, please pay regular and careful attention to additional and different information to be disclosed by Renesas such as that disclosed through our website. (http://www.renesas.com)
5. Renesas has used reasonable care in compiling the information included in this document, but Renesas assumes no liability whatsoever for any damages incurred as a result of errors or omissions in the information included in this document.
6. When using or otherwise relying on the information in this document, you should evaluate the information in light of the total system before deciding about the applicability of such information to the intended application. Renesas makes no representations, warranties or guaranties regarding the suitability of its products for any particular application and specifically disclaims any liability arising out of the application and use of the information in this document or Renesas products.
7. With the exception of products specified by Renesas as suitable for automobile applications, Renesas products are not designed, manufactured or tested for applications or otherwise in systems the failure or malfunction of which may cause a direct threat to human life or create a risk of human injury or which require especially high quality and reliability such as safety systems, or equipment or systems for transportation and traffic, healthcare, combustion control, aerospace and aeronautics, nuclear power, or undersea communication transmission. If you are considering the use of our products for such purposes, please contact a Renesas sales office beforehand. Renesas shall have no liability for damages arising out of the uses set forth above.
8. Notwithstanding the preceding paragraph, you should not use Renesas products for the purposes listed below:
   (1) artificial life support devices or systems
   (2) surgical implantations
   (3) healthcare intervention (e.g., excision, administration of medication, etc.)
   (4) any other purposes that pose a direct threat to human life

   Renesas shall have no liability for damages arising out of the uses set forth in the above and purchasers who elect to use Renesas products in any of the foregoing applications shall indemnify and hold harmless Renesas Technology Corp., its affiliated companies and their officers, directors, and employees against any and all damages arising out of such applications.
9. You should use the products described herein within the range specified by Renesas, especially with respect to the maximum rating, operating supply voltage range, movement power voltage range, heat radiation characteristics, installation and other product characteristics. Renesas shall have no liability for malfunctions or damages arising out of the use of Renesas products beyond such specified ranges.
10. Although Renesas endeavors to improve the quality and reliability of its products, IC products have specific characteristics such as the occurrence of failure at a certain rate and malfunctions under certain use conditions. Please be sure to implement safety measures to guard against the possibility of physical injury, and injury or damage caused by fire in the event of the failure of a Renesas 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 applicable measures. Among others, since the evaluation of microcomputer software alone is very difficult, please evaluate the safety of the final products or system manufactured by you.
11. In case Renesas products listed in this document are detached from the products to which the Renesas products are attached or affixed, the risk of accident such as swallowing by infants and small children is very high. You should implement safety measures so that Renesas products may not be easily detached from your products. Renesas shall have no liability for damages arising out of such detachment.
12. This document may not be reproduced or duplicated, in any form, in whole or in part, without prior written approval from Renesas.
13. Please contact a Renesas sales office if you have any questions regarding the information contained in this document, Renesas semiconductor products, or if you have any other inquiries.

General Precautions in the Handling of MPU/MCU Products

The following usage notes are applicable to all MPU/MCU products from Renesas. For detailed usage notes on the products covered by this manual, refer to the relevant sections of the manual. If the descriptions under General Precautions in the Handling of MPU/MCU Products and in the body of the manual differ from each other, the description in the body of the manual takes precedence.

1. Handling of Unused Pins

   Handle unused pins in accord with the directions given under Handling of Unused Pins in the manual.

   — The input pins of CMOS products are generally in the high-impedance state. In operation with an unused pin in the open-circuit state, extra electromagnetic noise is induced in the vicinity of LSI, an associated shoot-through current flows internally, and malfunctions may occur due to the false recognition of the pin state as an input signal. Unused pins should be handled as described under Handling of Unused Pins in the manual.

2. Processing at Power-on

   The state of the product is undefined at the moment when power is supplied.

   — The states of internal circuits in the LSI are indeterminate and the states of register settings and pins are undefined at the moment when power is supplied.

   In a finished product where the reset signal is applied to the external reset pin, the states of pins are not guaranteed from the moment when power is supplied until the reset process is completed.

   In a similar way, the states of pins in a product that is reset by an on-chip power-on reset function are not guaranteed from the moment when power is supplied until the power reaches the level at which resetting has been specified.

3. Prohibition of Access to Reserved Addresses

   Access to reserved addresses is prohibited.

   — The reserved addresses are provided for the possible future expansion of functions. Do not access these addresses; the correct operation of LSI is not guaranteed if they are accessed.

4. Clock Signals

   After applying a reset, only release the reset line after the operating clock signal has become stable. When switching the clock signal during program execution, wait until the target clock signal has stabilized.

   — When the clock signal is generated with an external resonator (or from an external oscillator) during a reset, ensure that the reset line is only released after full stabilization of the clock signal. Moreover, when switching to a clock signal produced with an external resonator (or by an external oscillator) while program execution is in progress, wait until the target clock signal is stable.

5. Differences between Products

   Before changing from one product to another, i.e. to one with a different type number, confirm that the change will not lead to problems.

   — The characteristics of MPU/MCU in the same group but having different type numbers may differ because of the differences in internal memory capacity and layout pattern. When changing to products of different type numbers, implement a system-evaluation test for each of the products.

Preface

The H8/3039 Group comprises high-performance single-chip microcomputers (MCUs) that integrate system supporting functions together with an H8/300H CPU core.

The H8/300H CPU has a 32-bit internal architecture with sixteen 16-bit general registers, and a concise, optimized instruction set designed for speed. It can address a 16-Mbyte linear address space.

The on-chip system supporting functions include ROM, RAM, a 16-bit integrated timer unit (ITU), a programmable timing pattern controller (TPC), a watchdog timer (WDT), a serial communication interface (SCI), an A/D converter, I/O ports, and other facilities. Of the two SCI channels, one has been expanded to support the ISO/IEC 7816-3 smart card interface. Functions have also been added to reduce power consumption in battery-powered applications: individual modules can be placed in standby, and the frequency of the system clock supplied to the chip can be divided down under software control.

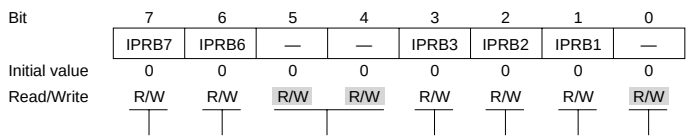
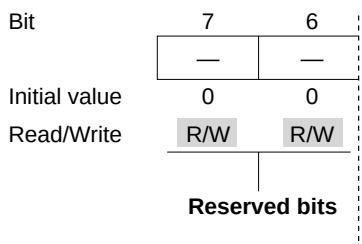
The five MCU operating modes offer a choice of expanded mode, single-chip mode, and address space size, enabling the H8/3039 Group to adapt quickly and flexibly to a variety of conditions.

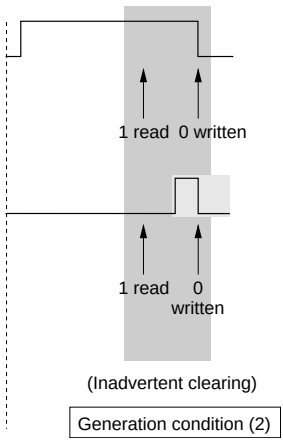
In addition to its mask-ROM versions, the H8/3039 Group has an F-ZTAT™ version with user programmable on-chip flash memory that can be programmed on-board. These versions enable users to respond quickly and flexibly to changing application specifications.

This manual describes the H8/3039 Group hardware. For details of the instruction set, refer to the H8/300H Series Software Manual.

Note: F-ZTAT is a trademark of Renesas Technology Corp.

Main Revisions for This Edition

Item	Page	Revision (See Manual for Details)
All	—	• Notification of change in company name amended (Before) Hitachi, Ltd. → (After) Renesas Technology Corp. • Product naming convention amended (Before) H8/3039 Series → (After) H8/3039 Group
2.3 Address Space Figure 2.2 Memory Map	20	Figure amended H'0000 H'FFFF  1. Normal mode (64-Kbyte mode)
5.2.2 Interrupt Priority Registers A and B (IPRA, IPRB) Interrupt Priority Register B (IPRB)	91	Description amended 
5.2.3 IRQ Status Register (ISR)	93	Description amended Bits 5, 4, 1 and 0—IRQ_5, IRQ_4, IRQ_1 and IRQ_0 Flags (IRQ5F, IRQ4F, IRQ1F, and IRQ0F): These bits indicate the status of IRQ_5, IRQ_4, IRQ_1, and IRQ_0 interrupt requests.
5.2.4 IRQ Enable Register (IER)	94	Description amended  Bits 5, 4, 1, and 0—IRQ_5, IRQ_4, IRQ_1, and IRQ_0 Enable (IRQ5E, IRQ4E, IRQ1E, IRQ0E): These bits enable or disable IRQ5, IRQ4, IRQ1, IRQ0 interrupts.

Item	Page	Revision (See Manual for Details)
5.3.3 Interrupt Vector Table Table 5.3 Interrupt Sources, Vector Addresses, and Priority	98	Table amended WOVI (interval timer)
5.5.4 Usage Notes Figure 5.9 IRQnF Flag when Interrupt Exception Handling is not Executed	109	Figure amended 
6.4.2 Precautions on Setting ASTCR and ABWCR*	131	Description amended Modes 5 and 7 ASTCR0 = 0 ABWCR = H'FC
11.2.8 Bit Rate Register (BRR)	349	Description added The baud rate generator is controlled separately for the individual channels, so different values may be set for each.
Table 11.3 Examples of Bit Rates and BRR Settings in Asynchronous Mode	351	Table amended **ϕ (MHz)** / **12** **Bit Rate (bits/s)** / **n** / **N** / **Error (%)** 300 / 2 / 77 / 0.16
11.3.4 Synchronous Operation Clock	376	Description amended An internal clock generated by the on-chip baud rate generator or an external clock input from the SCK pin can be selected by clearing or setting the CKE1 and CKE0 bits in SCR and the C/$\overline{A}$ bit in SMR. See table 11.9.

Item	Page	Revision (See Manual for Details)
16.2.1 Connecting a Crystal Resonator Table 16.2 Crystal Resonator Parameters	500	Preliminary deleted
18.1.3 AC Characteristics Table 18.5 Control Signal Timing	532	Table amended
18.1.4 A/D Conversion Characteristics	535	Newly added
18.2.2 DC Characteristics Table 18.10 Permissible Output Currents	541	Table amended
A.1 Instruction List 8. Block transfer instructions	576	Table amended
A.3 Number of States Required for Execution Table A.4 Number of Cycles per Instruction	584	Table amended

Item	Symbol	Condition A 8 MHz Min	Condition A 8 MHz Max	Condition B 10 MHz Min	Condition B 10 MHz Max	Condition C 18 MHz Min	Condition C 18 MHz Max	Unit	Test Conditions
$\overline{RES}$ setup time	t_{RESS}	200	—	200	—	200	—	ns	Figure 18.10
$\overline{RES}$ pulse width	t_{RESW}	10	—	10	—	10	—	tcyc	
Mode programming setup time (MD_2, MD_1, MD_0)	t_{MDS}	200	—	200	—	200	—	ns	

Item	
Permissible output low current (total)	Total of 27 pins including ports 1, 2, 5 and B

Mnemonic	Operand Size	Operation
EEPMOV. W	—	if R4 ≠ 0 then repeat @R5 → @R6 R5+1 → R5 R6+1 → R6 R4–1 → R4 until R4=0 else next

Instruction	Mnemonic		Word Data Access M	Internal Operation N
BSR	BSR d:16	Normal		2
		Advanced		2

All trademarks and registered trademarks are the property of their respective owners.

Contents

Section 1 Overview

1.1 Overview

The H8/3039 Group comprises microcomputers (MCUs) that integrate system supporting functions together with an H8/300H CPU core featuring an original Renesas Technology architecture.

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

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

The H8/3039 Group consists of four models: the H8/3039 with 128 kbytes of ROM and 4 kbytes of RAM, the H8/3038 with 64 kbytes of ROM and 2 kbytes of RAM, the H8/3037 with 32 kbytes of ROM and 1 kbytes of RAM, and the H8/3036 with 16 kbytes of ROM and 512 bytes of RAM.

The five MCU operating modes offer a choice of expanded mode, single-chip mode and address space size.

In addition to the mask-ROM version of the H8/3039 Group, an F-ZTAT™ version with an on-chip flash memory that can be freely programmed and reprogrammed by the user after the board is installed is also available. This version enables users to respond quickly and flexibly to changing application specifications, growing production volumes, and other conditions.

Table 1.1 summarizes the features of the H8/3039 Group.

Note: F-ZTAT is a trademark of Renesas Technology Corp.

Table 1.1 Features

Feature	Description
CPU	Upward-compatible with the H8/300 CPU at the object-code level General-register machine • Sixteen 16-bit general registers (also useable as sixteen 8-bit registers or eight 32-bit registers) High-speed operation • Maximum clock rate: 18 MHz • Add/subtract: 111 ns • Multiply/divide: 778 ns Two CPU operating modes • Normal mode (64-kbyte address space) • Advanced mode (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 bits × 16 bits) • Signed and unsigned divide instructions (16 bits ÷ 8 bits, 32 bits ÷ 16 bits) • Bit accumulator function • Bit manipulation instructions with register-indirect specification of bit positions

Feature	Description
Memory	H8/3039 • ROM: 128 kbytes • RAM: 4 kbytes H8/3038 • ROM: 64 kbytes • RAM: 2 kbytes H8/3037 • ROM: 32 kbytes • RAM: 1 kbyte H8/3036 • ROM: 16 kbytes • RAM: 512 bytes
Interrupt controller	• Five external interrupt pins: NMI, $\overline{IRQ}_0$, $\overline{IRQ}_1$, $\overline{IRQ}_4$, $\overline{IRQ}_5$ • 25 internal interrupts • Three selectable interrupt priority levels
Bus controller	• Address space can be partitioned into eight areas, with independent bus specifications in each area • Two-state or three-state access selectable for each area • Selection of four wait modes
16-bit integrated timer unit (ITU)	• Five 16-bit timer channels, capable of processing up to 12 pulse outputs or 10 pulse inputs • 16-bit timer counter (channels 0 to 4) • Two multiplexed output compare/input capture pins (channels 0 to 4) • 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)

<table>
<tr><th>Feature</th><th>Description</th></tr>
<tr><td>Programmable timing pattern controller (TPC)</td><td>• Maximum 15-bit pulse output, using ITU as time base
• Up to three 4-bit pulse output groups and one 3-bit pulse output group (or one 15-bit group, one 8-bit group, or one 7-bit group)
• Non-overlap mode available</td></tr>
<tr><td>Watchdog timer (WDT), 1 channel</td><td>• Reset signal can be generated by overflow
• Reset signal can be output externally (However, not available with the F-ZTAT version.)
• Usable as an interval timer</td></tr>
<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
• Can be externally triggered</td></tr>
<tr><td>I/O ports</td><td>• 55 input/output pins
• 8 input-only pins</td></tr>
<tr><td>Operating modes</td><td>Five MCU operating modes
<table>
<tr><th>Mode</th><th>Address Space</th><th>Address Pins</th><th>Bus Width</th></tr>
<tr><td>Mode 1</td><td>1 Mbyte</td><td>A_0 to A_{19}</td><td>8 bits</td></tr>
<tr><td>Mode 3</td><td>16 Mbytes</td><td>A_{23} to A_0</td><td>8 bits</td></tr>
<tr><td>Mode 5</td><td>1 Mbyte</td><td>A_0 to A_{19}</td><td>8 bits</td></tr>
<tr><td>Mode 6</td><td>64 kbytes</td><td>—</td><td>—</td></tr>
<tr><td>Mode 7</td><td>1 Mbyte</td><td>—</td><td>—</td></tr>
</table>
• On-chip ROM is disabled in modes 1 and 3</td></tr>
<tr><td>Power-down state</td><td>• Sleep mode
• Software standby mode
• Hardware standby mode
• Module standby function
• Programmable System clock frequency division</td></tr>
</table>

Feature	Description			
Other features	• On-chip clock oscillator			
Product lineup	**Model (5 V)**	**Model (3 V)***	**Package**	**ROM**
	HD64F3039F	HD64F3039VF	80-pin QFP (FP-80A)	Flash memory
	HD64F3039TE	HD64F3039VTE	80-pin TQFP (TFP-80C)	
	HD6433039F	HD6433039VF	80-pin QFP (FP-80A)	Mask ROM
	HD6433039TE	HD6433039VTE	80-pin TQFP (TFP-80C)	
	HD6433038F	HD6433038VF	80-pin QFP (FP-80A)	Mask ROM
	HD6433038TE	HD6433038VTE	80-pin TQFP (TFP-80C)	
	HD6433037F	HD6433037VF	80-pin QFP (FP-80A)	Mask ROM
	HD6433037TE	HD6433037VTE	80-pin TQFP (TFP-80C)	
	HD6433036F	HD6433036VF	80-pin QFP (FP-80A)	Mask ROM
	HD6433036TE	HD6433036VTE	80-pin TQFP (TFP-80C)	

Note: * There are two 3 V versions: one with V_{cc} = 2.7 V to 5.5 V and ϕ = 2 to 8 MHz, and one with V_{cc} = 3.0 V to 5.5 V and ϕ = 2 to 10 MHz. However, there is only one flash memory version, with V_{cc} = 3.0 to 5.5 V and ϕ = 2 to 10 MHz.

1.2 Block Diagram

Figure 1.1 shows an internal block diagram of the H8/3039 Group.

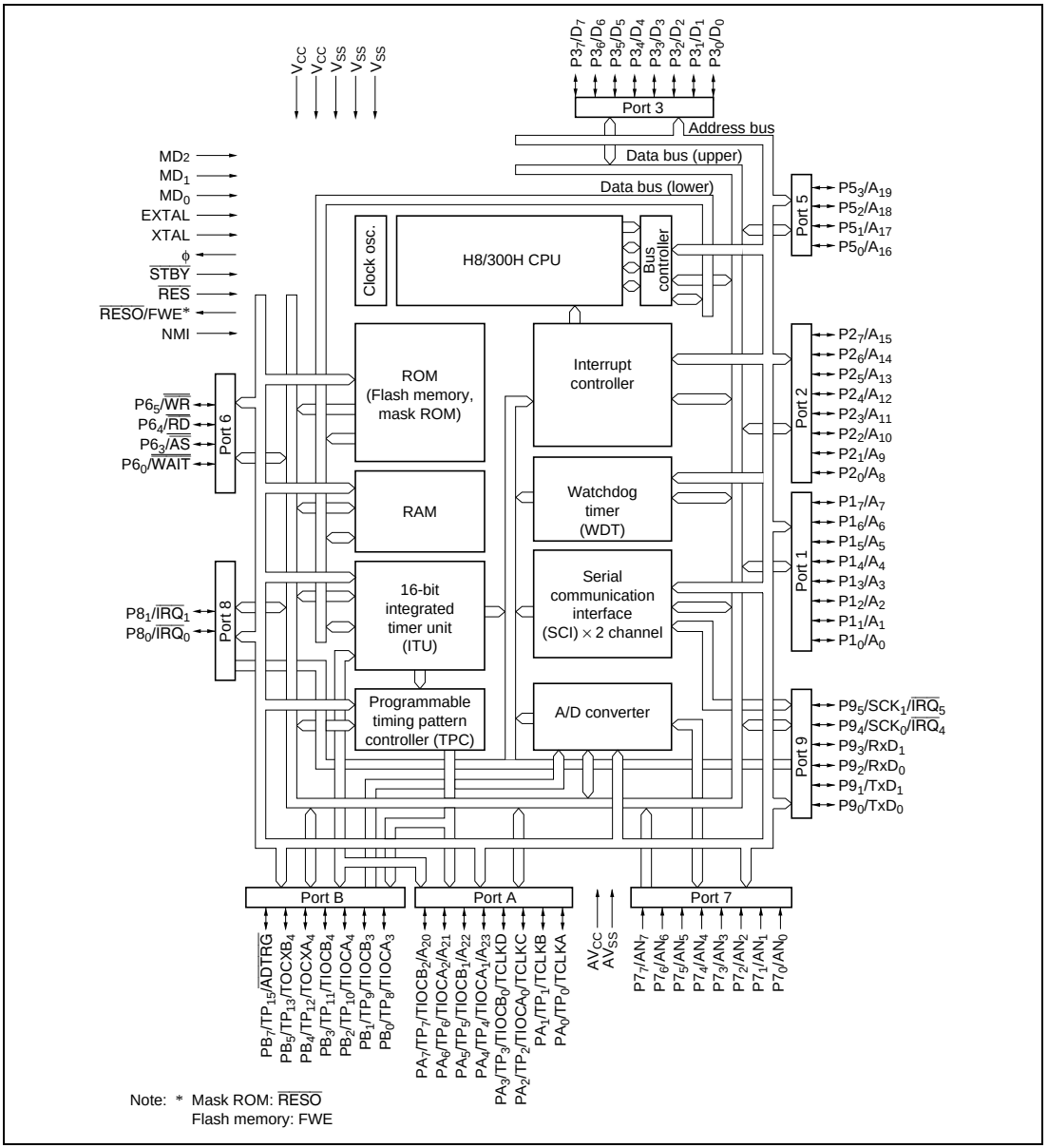


Figure 1.1 Block Diagram

1.3 Pin Description

1.3.1 Pin Arrangement

Figure 1.2 shows the pin arrangement of the H8/3039 Group.

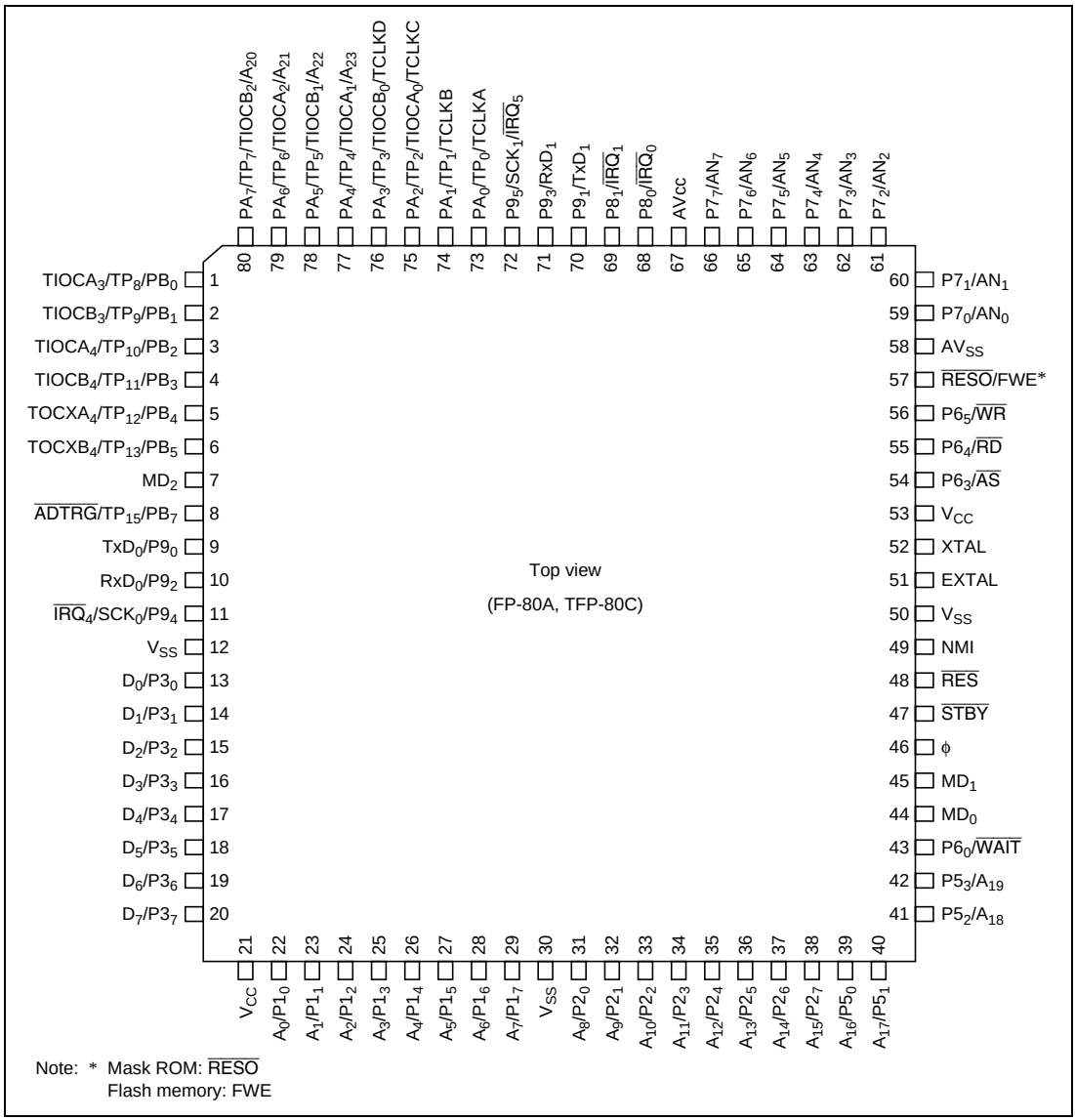


Figure 1.2 Pin Arrangement (FP-80A, TFP-80C Top View)

1.3.2 Pin Functions

Pin Assignments in Each Mode

Table 1.2 lists the FP-80A and TFP-80C pin assignments in each mode.

Table 1.2 FP-80A and TFP-80C Pin Assignments in Each Mode

	Pin Name					
Pin No.	Mode 1	Mode 3	Mode 5	Mode 6	Mode 7	PROM Mode Flash memory
1	$PB_0/TP_8/TIOCA_3$	$PB_0/TP_8/TIOCA_3$	$PB_0/TP_8/TIOCA_3$	$PB_0/TP_8/TIOCA_3$	$PB_0/TP_8/TIOCA_3$	NC
2	$PB_1/TP_9/TIOCB_3$	$PB_1/TP_9/TIOCB_3$	$PB_1/TP_9/TIOCB_3$	$PB_1/TP_9/TIOCB_3$	$PB_1/TP_9/TIOCB_3$	NC
3	$PB_2/TP_{10}/TIOCA_4$	$PB_2/TP_{10}/TIOCA_4$	$PB_2/TP_{10}/TIOCA_4$	$PB_2/TP_{10}/TIOCA_4$	$PB_2/TP_{10}/TIOCA_4$	NC
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
5	$PB_4/TP_{12}/TOCXA_4$	$PB_4/TP_{12}/TOCXA_4$	$PB_4/TP_{12}/TOCXA_4$	$PB_4/TP_{12}/TOCXA_4$	$PB_4/TP_{12}/TOCXA_4$	NC
6	$PB_5/TP_{13}/TOCXB_4$	$PB_5/TP_{13}/TOCXB_4$	$PB_5/TP_{13}/TOCXB_4$	$PB_5/TP_{13}/TOCXB_4$	$PB_5/TP_{13}/TOCXB_4$	NC
7	MD_2	MD_2	MD_2	MD_2	MD_2	V_{SS}
8	$\overline{PB_7/TP_{15}/ADTRG}$	$\overline{PB_7/TP_{15}/ADTRG}$	$\overline{PB_7/TP_{15}/ADTRG}$	$\overline{PB_7/TP_{15}/ADTRG}$	$\overline{PB_7/TP_{15}/ADTRG}$	NC
9	$P9_0/TxD_0$	$P9_0/TxD_0$	$P9_0/TxD_0$	$P9_0/TxD_0$	$P9_0/TxD_0$	NC
10	$P9_2/RxD_0$	$P9_2/RxD_0$	$P9_2/RxD_0$	$P9_2/RxD_0$	$P9_2/RxD_0$	V_{SS}
11	$\overline{P9_4/SCK_0/IRQ_4}$	$\overline{P9_4/SCK_0/IRQ_4}$	$\overline{P9_4/SCK_0/IRQ_4}$	$\overline{P9_4/SCK_0/IRQ_4}$	$\overline{P9_4/SCK_0/IRQ_4}$	NC
12	V_{SS}	V_{SS}	V_{SS}	V_{SS}	V_{SS}	V_{SS}
13	D_0	D_0	D_0	$P3_0$	$P3_0$	I/O_0
14	D_1	D_1	D_1	$P3_1$	$P3_1$	I/O_1
15	D_2	D_2	D_2	$P3_2$	$P3_2$	I/O_2
16	D_3	D_3	D_3	$P3_3$	$P3_3$	I/O_3
17	D_4	D_4	D_4	$P3_4$	$P3_4$	I/O_4

Pin No.	Pin Name					
	Mode 1	Mode 3	Mode 5	Mode 6	Mode 7	PROM Mode Flash memory
18	D_5	D_5	D_5	$P3_5$	$P3_5$	I/O_5
19	D_6	D_6	D_6	$P3_6$	$P3_6$	I/O_6
20	D_7	D_7	D_7	$P3_7$	$P3_7$	I/O_7
21	V_{CC}	V_{CC}	V_{CC}	V_{CC}	V_{CC}	V_{CC}
22	A_0	A_0	$P1_0/A_0$	$P1_0$	$P1_0$	A_0
23	A_1	A_1	$P1_1/A_1$	$P1_1$	$P1_1$	A_1
24	A_2	A_2	$P1_2/A_2$	$P1_2$	$P1_2$	A_2
25	A_3	A_3	$P1_3/A_3$	$P1_3$	$P1_3$	A_3
26	A_4	A_4	$P1_4/A_4$	$P1_4$	$P1_4$	A_4
27	A_5	A_5	$P1_5/A_5$	$P1_5$	$P1_5$	A_5
28	A_6	A_6	$P1_6/A_6$	$P1_6$	$P1_6$	A_6
29	A_7	A_7	$P1_7/A_7$	$P1_7$	$P1_7$	A_7
30	V_{SS}	V_{SS}	V_{SS}	V_{SS}	V_{SS}	V_{SS}
31	A_8	A_8	$P2_0/A_8$	$P2_0$	$P2_0$	A_8
32	A_9	A_9	$P2_1/A_9$	$P2_1$	$P2_1$	A_9
33	A_{10}	A_{10}	$P2_2/A_{10}$	$P2_2$	$P2_2$	A_{10}
34	A_{11}	A_{11}	$P2_3/A_{11}$	$P2_3$	$P2_3$	A_{11}
35	A_{12}	A_{12}	$P2_4/A_{12}$	$P2_4$	$P2_4$	A_{12}
36	A_{13}	A_{13}	$P2_5/A_{13}$	$P2_5$	$P2_5$	A_{13}
37	A_{14}	A_{14}	$P2_6/A_{14}$	$P2_6$	$P2_6$	A_{14}
38	A_{15}	A_{15}	$P2_7/A_{15}$	$P2_7$	$P2_7$	A_{15}
39	A_{16}	A_{16}	$P5_0/A_{16}$	$P5_0$	$P5_0$	A_{16}
40	A_{17}	A_{17}	$P5_1/A_{17}$	$P5_1$	$P5_1$	V_{SS}
41	A_{18}	A_{18}	$P5_2/A_{18}$	$P5_2$	$P5_2$	V_{SS}
42	A_{19}	A_{19}	$P5_3/A_{19}$	$P5_3$	$P5_3$	V_{SS}
43	$P6_0/\overline{WAIT}$	$P6_0/\overline{WAIT}$	$P6_0/\overline{WAIT}$	$P6_0$	$P6_0$	NC
44	MD_0	MD_0	MD_0	MD_0	MD_0	V_{SS}
45	MD_1	MD_1	MD_1	MD_1	MD_1	V_{SS}
46	ϕ	ϕ	ϕ	ϕ	ϕ	NC

Pin No.	Pin Name					
	Mode 1	Mode 3	Mode 5	Mode 6	Mode 7	PROM Mode Flash memory
47	$\overline{STBY}$	$\overline{STBY}$	$\overline{STBY}$	$\overline{STBY}$	$\overline{STBY}$	V_{CC}
48	$\overline{RES}$	$\overline{RES}$	$\overline{RES}$	$\overline{RES}$	$\overline{RES}$	$\overline{RES}$
49	NMI	NMI	NMI	NMI	NMI	V_{CC}
50	V_{SS}	V_{SS}	V_{SS}	V_{SS}	V_{SS}	V_{SS}
51	EXTAL	EXTAL	EXTAL	EXTAL	EXTAL	EXTAL
52	XTAL	XTAL	XTAL	XTAL	XTAL	XTAL
53	V_{CC}	V_{CC}	V_{CC}	V_{CC}	V_{CC}	V_{CC}
54	$\overline{AS}$	$\overline{AS}$	$\overline{AS}$	$P6_3$	$P6_3$	NC
55	$\overline{RD}$	$\overline{RD}$	$\overline{RD}$	$P6_4$	$P6_4$	NC
56	$\overline{WR}$	$\overline{WR}$	$\overline{WR}$	$P6_5$	$P6_5$	V_{CC}
57	$\overline{RESO}$/ FWE*	$\overline{RESO}$/ FWE*	$\overline{RESO}$/ FWE*	$\overline{RESO}$/ FWE*	$\overline{RESO}$/ FWE*	FWE
58	AV_{SS}	AV_{SS}	AV_{SS}	AV_{SS}	AV_{SS}	V_{SS}
59	$P7_0/AN_0$	$P7_0/AN_0$	$P7_0/AN_0$	$P7_0/AN_0$	$P7_0/AN_0$	NC
60	$P7_1/AN_1$	$P7_1/AN_1$	$P7_1/AN_1$	$P7_1/AN_1$	$P7_1/AN_1$	NC
61	$P7_2/AN_2$	$P7_2/AN_2$	$P7_2/AN_2$	$P7_2/AN_2$	$P7_2/AN_2$	NC
62	$P7_3/AN_3$	$P7_3/AN_3$	$P7_3/AN_3$	$P7_3/AN_3$	$P7_3/AN_3$	NC
63	$P7_4/AN_4$	$P7_4/AN_4$	$P7_4/AN_4$	$P7_4/AN_4$	$P7_4/AN_4$	NC
64	$P7_5/AN_5$	$P7_5/AN_5$	$P7_5/AN_5$	$P7_5/AN_5$	$P7_5/AN_5$	NC
65	$P7_6/AN_6$	$P7_6/AN_6$	$P7_6/AN_6$	$P7_6/AN_6$	$P7_6/AN_6$	NC
66	$P7_7/AN_7$	$P7_7/AN_7$	$P7_7/AN_7$	$P7_7/AN_7$	$P7_7/AN_7$	NC
67	AV_{CC}	AV_{CC}	AV_{CC}	AV_{CC}	AV_{CC}	V_{CC}
68	$P8_0/\overline{IRQ}_0$	$P8_0/\overline{IRQ}_0$	$P8_0/\overline{IRQ}_0$	$P8_0/\overline{IRQ}_0$	$P8_0/\overline{IRQ}_0$	V_{SS}
69	$P8_1/\overline{IRQ}_1$	$P8_1/\overline{IRQ}_1$	$P8_1/\overline{IRQ}_1$	$P8_1/\overline{IRQ}_1$	$P8_1/\overline{IRQ}_1$	V_{SS}
70	$P9_1/TxD_1$	$P9_1/TxD_1$	$P9_1/TxD_1$	$P9_1/TxD_1$	$P9_1/TxD_1$	NC
71	$P9_3/RxD_1$	$P9_3/RxD_1$	$P9_3/RxD_1$	$P9_3/RxD_1$	$P9_3/RxD_1$	NC
72	$P9_5/SCK_1/\overline{IRQ}_5$	$P9_5/SCK_1/\overline{IRQ}_5$	$P9_5/SCK_1/\overline{IRQ}_5$	$P9_5/SCK_1/\overline{IRQ}_5$	$P9_5/SCK_1/\overline{IRQ}_5$	V_{CC}
73	PA_0/TP_0/ TCLKA	PA_0/TP_0/ TCLKA	PA_0/TP_0/ TCLKA	PA_0/TP_0/ TCLKA	PA_0/TP_0/ TCLKA	$\overline{CE}$

Pin No.	Pin Name					
	Mode 1	Mode 3	Mode 5	Mode 6	Mode 7	PROM Mode Flash memory
74	PA_1/TP_1/ TCLKB	PA_1/TP_1/ TCLKB	PA_1/TP_1/ TCLKB	PA_1/TP_1/ TCLKB	PA_1/TP_1/ TCLKB	$\overline{OE}$
75	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	$\overline{WE}$
76	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
77	PA_4/TP_4/ $TIOCA_1$	PA_4/TP_4/ $TIOCA_1/A_{23}$	PA_4/TP_4/ $TIOCA_1$	PA_4/TP_4/ $TIOCA_1$	PA_4/TP_4/ $TIOCA_1$	NC
78	PA_5/TP_5/ $TIOCB_1$	PA_5/TP_5/ $TIOCB_1/A_{22}$	PA_5/TP_5/ $TIOCB_1$	PA_5/TP_5/ $TIOCB_1$	PA_5/TP_5/ $TIOCB_1$	NC
79	PA_6/TP_6/ $TIOCA_2$	PA_6/TP_6/ $TIOCA_2/A_{21}$	PA_6/TP_6/ $TIOCA_2$	PA_6/TP_6/ $TIOCA_2$	PA_6/TP_6/ $TIOCA_2$	NC
80	PA_7/TP_7/ $TIOCB_2$	A_{20}	PA_7/TP_7/ $TIOCB_2$	PA_7/TP_7/ $TIOCB_2$	PA_7/TP_7/ $TIOCB_2$	NC

Notes: Pins marked NC should be left unconnected.

For details about PROM mode see section 15, ROM.

* Mask ROM: $\overline{RESO}$
Flash Memory: FWE

1.4 Pin Functions

Table 1.3 summarizes the pin functions.

Table 1.3 Pin Functions

Type	Symbol	Pin No.	I/O	Name and Function
Power	V_{CC}	21, 53	Input	**Power:** For connection to the power supply. Connect all V_{CC} pins to the system power supply.
	V_{SS}	12, 30, 50	Input	**Ground:** For connection to ground (0 V). Connect all V_{SS} pins to the 0-V system power supply.
Clock	XTAL	52	Input	For connection to a crystal resonator For examples of crystal resonator and external clock input, see section 16, Clock Pulse Generator.
	EXTAL	51	Input	For connection to a crystal resonator or input of an external clock signal. For examples of crystal resonator and external clock input, see section 16, Clock Pulse Generator.
	ϕ	46	Output	System clock: Supplies the system clock to external devices
Operating mode control	MD_2, MD_1, MD_0	7, 45, 44	Input	**Mode 2 to mode 0:** For setting the operating mode, as follows. These pins should not be changed during operation.

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

Type	Symbol	Pin No.	I/O	Name and Function
System control	$\overline{RES}$	48	Input	**Reset input:** When driven low, this pin resets the chip
	$\overline{RESO}$/ FWE	57	Output/ Input	**Reset output** (Mask ROM version): Outputs WDT-generated reset signal to an external device. **Write enable signal** (F-ZTAT version): Flash memory write control signal.
	$\overline{STBY}$	47	Input	**Standby:** When driven low, this pin forces a transition to hardware standby mode
Interrupts	NMI	49	Input	**Nonmaskable interrupt:** Requests a nonmaskable interrupt
	$\overline{IRQ}_5$, $\overline{IRQ}_4$ $\overline{IRQ}_1$, $\overline{IRQ}_0$	72, 11, 69, 68	Input	**Interrupt request 5, 4, 1, 0:** Maskable interrupt request pins
Address bus	A_{23} to A_{20}, A_{19} to A_8, A_7 to A_0	77 to 80, 42 to 31, 29 to 22	Output	**Address bus:** Outputs address signals
Data bus	D_7 to D_0	20 to 13	Input/ output	**Data bus:** Bidirectional data bus
Bus control	$\overline{AS}$	54	Output	**Address strobe:** Goes low to indicate valid address output on the address bus
	$\overline{RD}$	55	Output	**Read:** Goes low to indicate reading from the external address space.
	$\overline{WR}$	56	Output	**Write:** Goes low to indicate writing to the external address space indicates valid data on the data bus.
	$\overline{WAIT}$	43	Input	**Wait:** Requests insertion of wait states in bus cycles during access to the external address space.
16-bit integrated timer unit (ITU)	TCLKD to TCLKA	76 to 73	Input	**Clock input A to D:** External clock inputs
	$TIOCA_4$ to $TIOCA_0$	3, 1, 79, 77, 75	Input/ Output	**Input capture/output compare A4 to A0:** GRA4 to GRA0 output compare or input capture, or PWM output
	$TIOCB_4$ to $TIOCB_0$	4, 2, 80, 78, 76	Input/ output	**Input capture/output compare B4 to B0** GRB4 to GRB0 output compare or input capture, or PWM output
	$TOCXA_4$	5	Output	**Output compare XA4:** PWM output
	$TOCXB_4$	6	Output	**Output compare XB4:** PWM output

Type	Symbol	Pin No.	I/O	Name and Function
Programmable timing pattern controller (TPC)	TP_{15}, TP_{13} to TP_0	8, 6 to 1 80 to 73	Output	**TPC output 15, 13 to 0 :** Pulse output
Serial communication interface (SCI)	TxD_1, TxD_0	70, 9	Output	**Transmit data**:(channels 0 and 1): SCI data output
	RxD_1, RxD_0	71, 10	Input	**Receive data:**(channels 0 and 1): SCI data input
	SCK_1, SCK_0	72, 11	Input/ output	**Serial clock:**(channels 0 and 1): SCI clock input/output
A/D converter	AN_7 to AN_0	66 to 59	Input	**Analog 7 to 0:** Analog input pins
	$\overline{ADTRG}$	8	Input	**A/D trigger:** External trigger input for starting A/D conversion
	AV_{CC}	67	Input	Power supply pin and reference voltage input pin for the A/D converter. Connect to the system power supply when not using the A/D converter.
	AV_{SS}	58	Input	Ground pin for the A/D converter. Connect to system power-supply (0 V).
I/O ports	$P1_7$ to $P1_0$	29 to 22	Input/ output	**Port 1:** Eight input/output pins. The direction of each pin can be selected in the port 1 data direction register (P1DDR).
	$P2_7$ to $P2_0$	38 to 31	Input/ output	**Port 2:** Eight input/output pins. The direction of each pin can be selected in the port 2 data direction register (P2DDR).
	$P3_7$ to $P3_0$	20 to 13	Input/ output	**Port 3:** Eight input/output pins. The direction of each pin can be selected in the port 3 data direction register (P3DDR).
	$P5_3$ to $P5_0$	42 to 39	Input/ output	**Port 5:** Four input/output pins. The direction of each pin can be selected in the port 5 data direction register (P5DDR).
	$P6_5$ to $P6_3$, $P6_0$	56 to 54, 43	Input/ output	**Port 6:** Four input/output pins. The direction of each pin can be selected in the port 6 data direction register (P6DDR).
	$P7_7$ to $P7_0$	66 to 59	Input	**Port 7:** Eight input pins
	$P8_1$, $P8_0$	69, 68	Input/ output	**Port 8:** Two input/output pins. The direction of each pin can be selected in the port 8 data direction register (P8DDR).

Type	Symbol	Pin No.	I/O	Name and Function
I/O ports	$P9_5$ to $P9_0$	72, 11 71, 10 70, 9	Input/ output	**Port 9:** Six input/output pins. The direction of each pin can be selected in the port 9 data direction register (P9DDR).
	PA_7 to PA_0	80 to 73	Input/ output	**Port A:** Eight input/output pins. The direction of each pin can be selected in the port A data direction register (PADDR).
	PB_7, PB_5 to PB_0	8, 6to1	Input/ output	**Port B:** Seven input/output pins. The direction of each pin can be selected in the port B data direction register (PBDDR).

Section 2 CPU

2.1 Overview

The H8/300H CPU is a high-speed central processing unit with an internal 32-bit architecture that is upward-compatible with the H8/300 CPU. The H8/300H CPU has sixteen 16-bit general registers, can address a 16-Mbyte linear address space, and is ideal for realtime control.

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 without alteration
- General-register architecture
  Sixteen 16-bit general registers (also usable as sixteen 8-bit registers or eight 32-bit registers)
- Sixty-two basic instructions
  - 8/16/32-bit arithmetic and logic instructions
  - Multiply and divide instructions
  - Powerful bit-manipulation instructions
- Eight addressing modes
  - Register direct [Rn]
  - Register indirect [@ERn]
  - Register indirect with displacement [@(d:16, ERn) or @(d:24, ERn)]
  - Register indirect with post-increment or pre-decrement [@ERn+ or @–ERn]
  - Absolute address [@aa:8, @aa:16, or @aa:24]
  - Immediate [#xx:8, #xx:16, or #xx:32]
  - Program-counter relative [@(d:8, PC) or @(d:16, PC)]
  - Memory indirect [@@aa:8]
- 16-Mbyte linear address space
- High-speed operation
  - All frequently-used instructions execute in two to four states
  - Maximum clock frequency: 18 MHz
  - 8/16/32-bit register-register add/subtract: 111 ns
  - 8 × 8-bit register-register multiply: 778 ns
  - 16 ÷ 8-bit register-register divide: 778 ns

— 16 × 16-bit register-register multiply: 1222 ns
— 32 ÷ 16-bit register-register divide: 1222 ns

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

2.1.2 Differences from H8/300 CPU

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

- More general registers
  Eight 16-bit registers have been added.
- Expanded address space
  — Advanced mode supports a maximum 16-Mbyte address space.
  — Normal mode supports the same 64-kbyte address space as the H8/300 CPU.
- Enhanced addressing
  The addressing modes have been enhanced to make effective use of the 16-Mbyte address 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.

2.2 CPU Operating Modes

The H8/300H CPU has two operating modes: normal and advanced. Normal mode supports a maximum 64-kbyte address space. Advanced mode supports up to 16 Mbytes. See figure 2.1.

Unless specified otherwise, all descriptions in this manual refer to advanced mode.

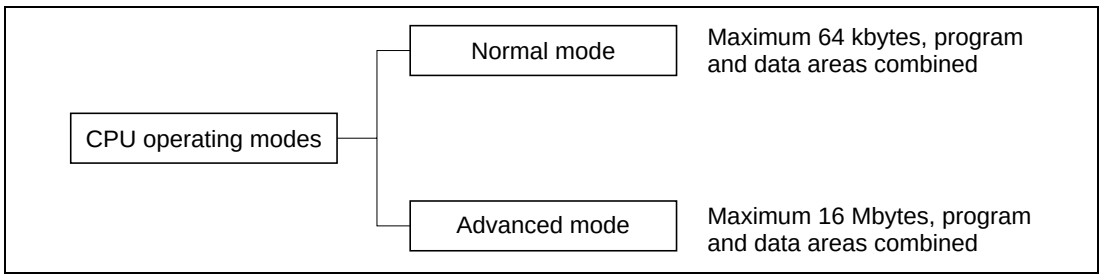


Figure 2.1 CPU Operating Modes

2.3 Address Space

The maximum address space of the H8/300H CPU is 16 Mbytes. This LSI allows selection of a normal mode and advanced mode 1-Mbyte mode or 16-Mbyte mode for the address space depending on the MCU operation mode. Figure 2.2 shows the address ranges of the H8/3039 Group. For further details see section 3.6, Memory Map in Each Operating Mode.

The 1-Mbyte operating mode uses 20-bit addressing. The upper 4 bits of effective addresses are ignored.

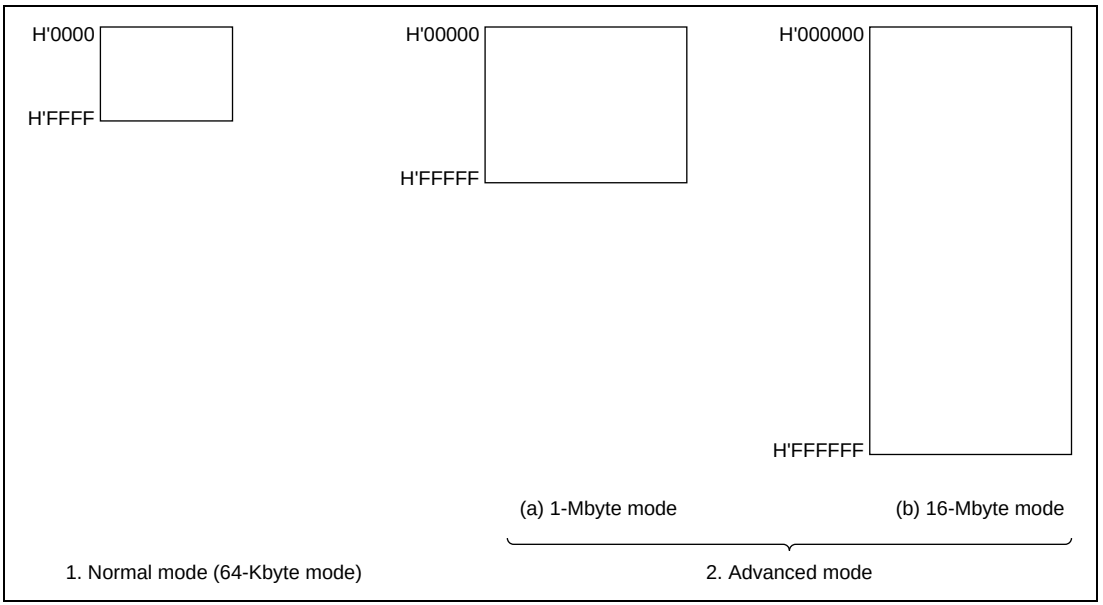


Figure 2.2 Memory Map

2.4 Register Configuration

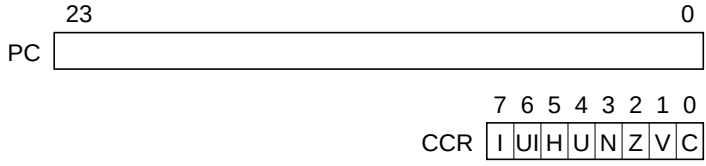
2.4.1 Overview

The H8/300H CPU has the internal registers shown in figure 2.3. There are two types of registers: general registers and control registers.

General Registers (ERn)

	15 0	7 0	7 0
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 Registers

2.4.2 General Registers

The H8/300H CPU has eight 32-bit general registers. These general registers are all functionally alike and can be used without distinction between data registers and address registers. When a general register is used as a data register, it can be accessed as a 32-bit, 16-bit, or 8-bit register. When the general registers are used as 32-bit registers or as address registers, they are designated by the letters ER (ER0 to ER7).

The ER registers divide into 16-bit general registers designated by the letters E (E0 to E7) and R (R0 to R7). These registers are functionally equivalent, providing a maximum sixteen 16-bit 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 R7H) and RL (R0L to R7L). These registers are functionally equivalent, providing a maximum sixteen 8-bit registers.

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

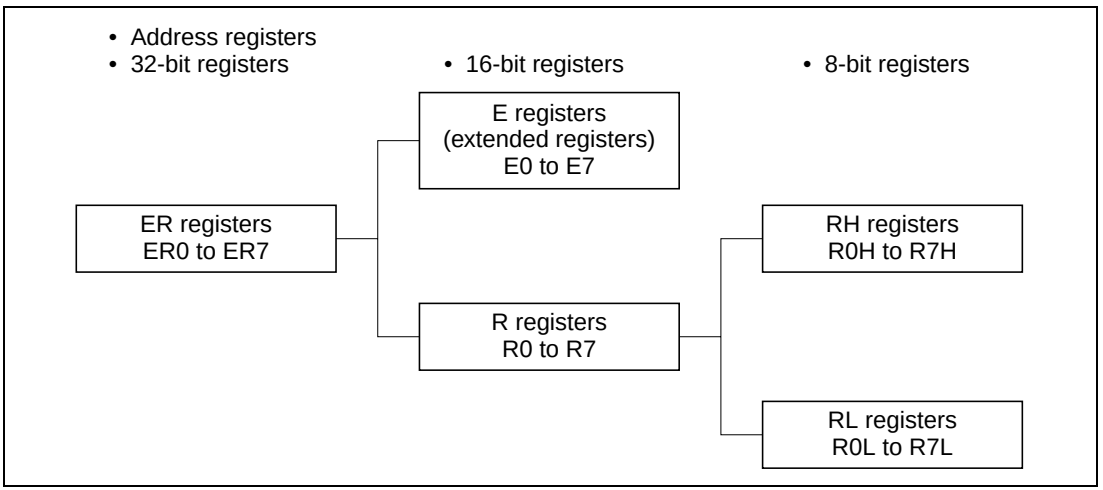


Figure 2.4 Usage of General Registers

General register ER7 has the function of stack pointer (SP) in addition to its general-register function, and is used implicitly in exception handling and subroutine calls. Figure 2.5 shows the stack.

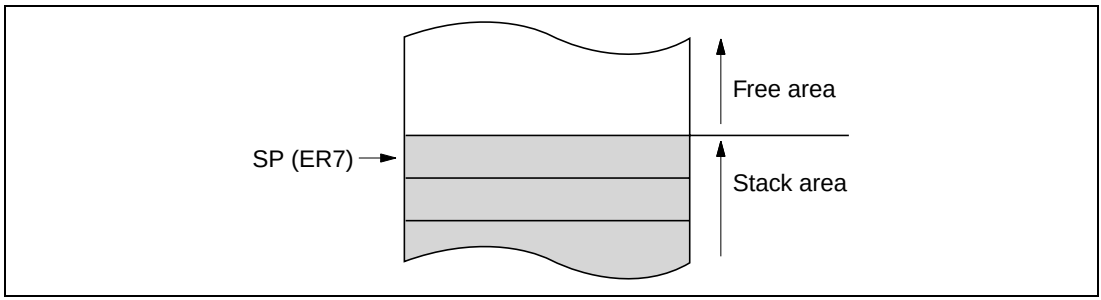


Figure 2.5 Stack

2.4.3 Control Registers

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

Program Counter (PC)

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

Condition Code Register (CCR)

This 8-bit register contains internal CPU status information, including the interrupt mask bit (I) 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. NMI is accepted regardless of the I bit setting. The I bit is set to 1 at the start of an exception-handling sequence.

Bit 6—User Bit or Interrupt Mask Bit (UI): Can be written and read by software using the LDC, STC, ANDC, ORC, and XORC instructions. This bit can also be used as an interrupt 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, CMP.B, or NEG.B instruction is executed, this flag is set to 1 if there is a carry or borrow at bit 3, and cleared to 0 otherwise. When the ADD.W, SUB.W, CMP.W, or NEG.W instruction is executed, the H flag is set to 1 if there is a carry or borrow at bit 11, and cleared to 0 otherwise. When the ADD.L, SUB.L, CMP.L, or NEG.L instruction is executed, the H flag is set to 1 if there is a carry or borrow at bit 27, and cleared to 0 otherwise.

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

Bit 3—Negative Flag (N): Indicates the most significant bit (sign bit) of data.

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

Bit 1—Overflow Flag (V): Set to 1 when an arithmetic overflow occurs, and cleared to 0 at other times.

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

- 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 the LDC, STC, ANDC, ORC, and XORC instructions. The N, Z, V, and C flags are used by conditional branch (Bcc) instructions.

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

2.5 Data Formats

The H8/300H CPU can process 1-bit, 4-bit (BCD), 8-bit (byte), 16-bit (word), and 32-bit (longword) data. Bit-manipulation instructions operate on 1-bit data by accessing bit n (n = 0, 1, 2, …, 7) of byte operand data. The DAA and DAS decimal-adjust instructions treat byte data as two digits of 4-bit BCD data.

2.5.1 General Register Data Formats

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

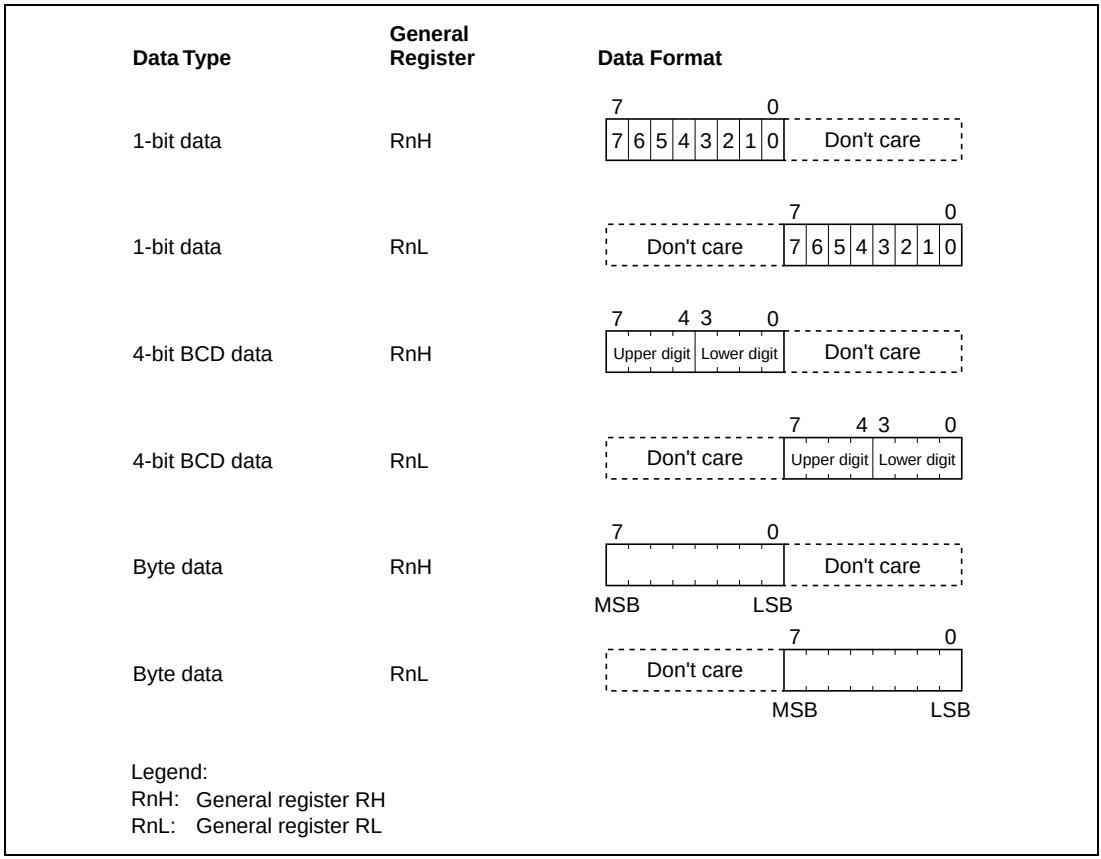


Figure 2.6 General Register Data Formats

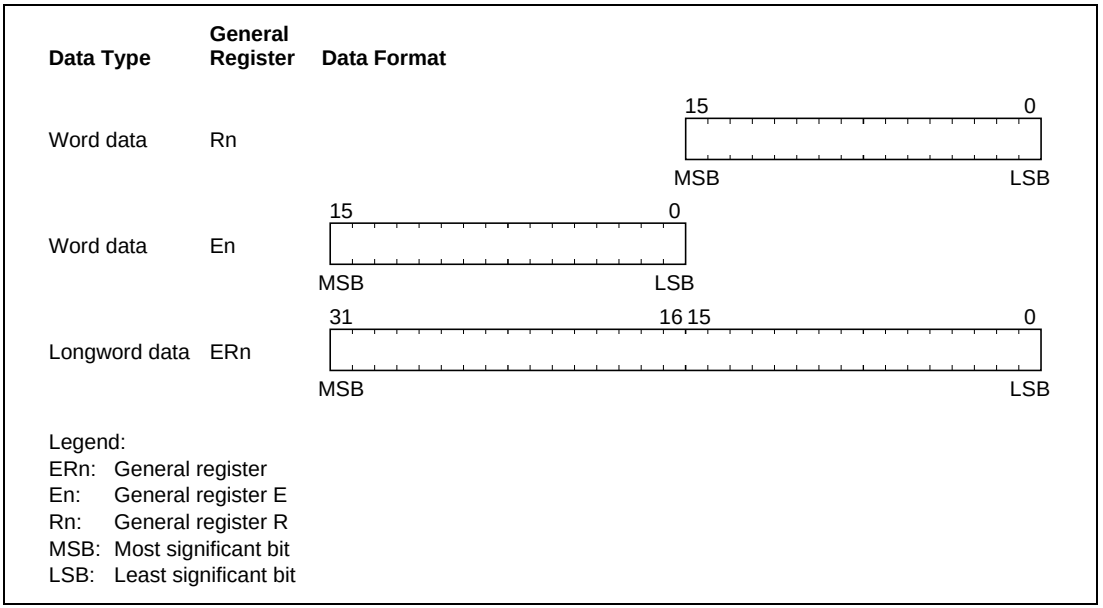


Figure 2.7 General Register Data Formats

2.5.2 Memory Data Formats

Figure 2.8 shows the data formats on memory. The H8/300H CPU can access word data and longword data on memory, but word or longword data must begin at an even address. If an attempt is made to access word or longword data at an odd address, no address error occurs but the least significant bit of the address is regarded as 0, so the access starts at the preceding address. This also applies to instruction fetches.

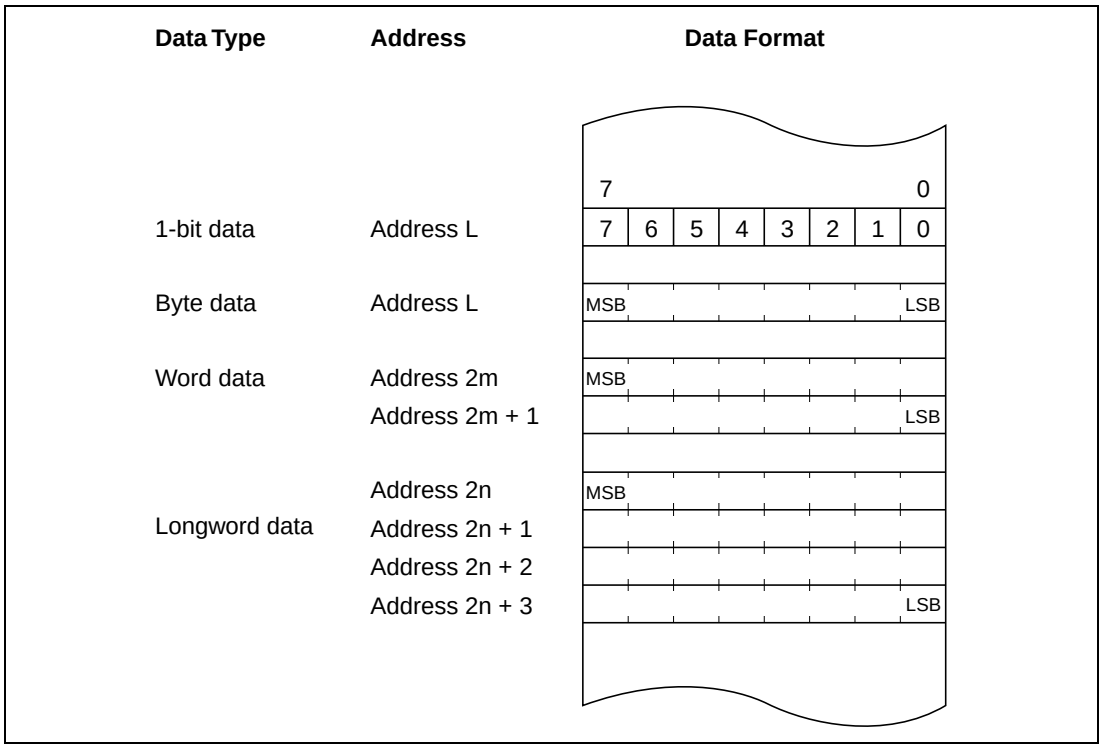


Figure 2.8 Memory Data Formats

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

2.6 Instruction Set

2.6.1 Instruction Set Overview

The H8/300H CPU has 62 types of instructions, which are classified as shown in table 2.1.

Table 2.1 Instruction Classification

Function	Instruction	Types
Data transfer	MOV, PUSH*[1], POP*[1], MOVTPE*[2], MOVFPE*[2]	3
Arithmetic operations	ADD, SUB, ADDX, SUBX, INC, DEC, ADDS, SUBS, DAA, DAS, MULXU, MULXS, DIVXU, DIVXS, CMP, NEG, EXTS, EXTU	18
Logic operations	AND, OR, XOR, NOT	4
Shift operations	SHAL, SHAR, SHLL, SHLR, ROTL, ROTR, ROTXL, ROTXR	8
Bit manipulation	BSET, BCLR, BNOT, BTST, BAND, BIAND, BOR, BIOR, BXOR, BIXOR, BLD, BILD, BST, BIST	14
Branch	Bcc*[3], JMP, BSR, JSR, RTS	5
System control	TRAPA, RTE, SLEEP, LDC, STC, ANDC, ORC, XORC, NOP	9
Block data transfer	EEPMOV	1
		Total 62 types

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. These instructions are not available on the H8/3039 Group.
3. Bcc is a generic branching instruction.

2.6.2 Instructions and Addressing Modes

Table 2.2 indicates the instructions available in the H8/300H CPU.

Table 2.2 Instructions and Addressing Modes

Function	Instruction	Addressing Modes												
		#xx	Rn	@ERn	@(d:16,ERn)	@(d:24, ERn)	@ERn+/@–ERn	@aa:8	@aa:16	@aa:24	@(d:8, PC)	@(d:16,PC)	@@aa:8	Implied
Data transfer	MOV	BWL	BWL	BWL	BWL	BWL	BWL	B	BWL	BWL	—	—	—	—
	POP, PUSH	—	—	—	—	—	—	—	—	—	—	—	—	WL
	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	—	—	—	—	—	—	—	—	—	—	—	—	○

Function	Instruction	Addressing Modes												
		#xx	Rn	@ERn	@(d:16,ERn)	@(d:24, ERn)	@ERn+/@–ERn	@aa:8	@aa:16	@aa:24	@(d:8, PC)	@(d:16,PC)	@@aa:8	Implied
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		—	—	—	—	—	—	—	—	—	—	—	—	BW

Legend:

B: Byte

W: Word

L: Longword

2.6.3 Tables of Instructions Classified by Function

Tables 2.3 to 2.10 summarize the instructions in each functional category. The operation notation used in these tables is defined as follows.

Operation Notation

Rd	General register (destination)*
Rs	General register (source)*
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
∧	Logical AND
∨	Logical OR
⊕	Exclusive logical OR
→	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 registers (R0 to R7, E0 to E7), and 32-bit data or address registers (ER0 to ER7).

Table 2.3 Data Transfer Instructions

Instruction	Size*	Function
MOV	B/W/L	(EAs) → Rd, Rs → (EAd) Moves data between two general registers or between a general register and memory, or moves immediate data to a general register.
MOVFPE	B	(EAs) → Rd Cannot be used in the H8/3039 Group.
MOVTPE	B	Rs → (EAs) Cannot be used in the H8/3039 Group.
POP	W/L	@SP+ → Rn Pops a general register from the stack. POP.W Rn is identical to MOV.W @SP+, Rn. Similarly, POP.L ERn is identical to MOV.L @SP+, ERn.
PUSH	W/L	Rn → @–SP Pushes a general register onto the stack. PUSH.W Rn is identical to MOV.W Rn, @–SP. Similarly, PUSH.L ERn is identical to MOV.L ERn, @–SP.

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

Table 2.4 Arithmetic Operation Instructions

Instruction	Size*	Function
ADD, SUB	B/W/L	Rd ± Rs → Rd, Rd ± #IMM → Rd Performs addition or subtraction on data in two general registers, or on immediate data and data in a general register. (Immediate byte data cannot be subtracted from data in a general register. Use the SUBX or ADD instruction.)
ADDX, SUBX	B	Rd ± Rs ± C → Rd, Rd ± #IMM ± C → Rd Performs addition or subtraction with carry on data in two general registers, or on immediate data and data in a general register.
INC, DEC	B/W/L	Rd ± 1 → Rd, Rd ± 2 → Rd Increments or decrements a general register by 1 or 2. (Byte 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 32-bit register.
DAA, DAS	B	Rd decimal adjust → Rd Decimal-adjusts an addition or subtraction result in a general register by referring to CCR to produce 4-bit BCD data.
MULXU	B/W	Rd × Rs → Rd Performs unsigned multiplication on data in two general registers: either 8 bits × 8 bits → 16 bits or 16 bits × 16 bits → 32 bits.
MULXS	B/W	Rd × Rs → Rd Performs signed multiplication on data in two general registers: either 8 bits × 8 bits → 16 bits or 16 bits × 16 bits → 32 bits.

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

Instruction	Size*	Function
DIVXU	B/W	Rd ÷ Rs → Rd Performs unsigned division on data in two general registers: either 16 bits ÷ 8 bits → 8-bit quotient and 8-bit remainder or 32 bits ÷ 16 bits → 16-bit quotient and 16-bit remainder.
DIVXS	B/W	Rd ÷ Rs → Rd Performs signed division on data in two general registers: either 16 bits ÷ 8 bits → 8-bit quotient and 8-bit remainder, or 32 bits ÷ 16 bits → 16-bit quotient and 16-bit remainder.
CMP	B/W/L	Rd – Rs, Rd – #IMM Compares data in a general register with data in another general register or with immediate data, and sets CCR according to the result.
NEG	B/W/L	0 – Rd → Rd Takes the two's complement (arithmetic complement) of data in a general register.
EXTS	W/L	Rd (sign extension) → Rd Extends byte data in the lower 8 bits of a 16-bit register to word data, or extends word data in the lower 16 bits of a 32-bit register to 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 to word data, or extends word data in the lower 16 bits of a 32-bit register to longword data, by padding with zeros.

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

Table 2.5 Logic Operation Instructions

Instruction	Size*	Function
AND	B/W/L	Rd ∧ Rs → Rd, Rd ∧ #IMM → Rd Performs a logical AND operation on a general register and another general register or immediate data.
OR	B/W/L	Rd ∨ Rs → Rd, Rd ∨ #IMM → Rd Performs a logical OR operation on a general register and another general register or immediate data.
XOR	B/W/L	Rd ⊕ Rs → Rd, Rd ⊕ #IMM → Rd Performs a logical exclusive OR operation on a general register and 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

Table 2.7 Bit Manipulation Instructions

Instruction	Size*	Function
BSET	B	1 → (<bit-No.> of <EAd>) Sets a specified bit in a general register or memory operand to 1. The bit number is specified by 3-bit immediate data or the lower 3 bits of a general register.
BCLR	B	0 → (<bit-No.> of <EAd>) Clears a specified bit in a general register or memory operand to 0. The bit number is specified by 3-bit immediate data or the lower 3 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 operand. The bit number is specified by 3-bit immediate data or the lower 3 bits of a general register.
BTST	B	¬ (<bit-No.> of <EAd>) → Z Tests a specified bit in a general register or memory operand and sets or clears the Z flag accordingly. The bit number is specified by 3-bit immediate data or the lower 3 bits of a general register.
BAND	B	C ∧ (<bit-No.> of <EAd>) → C ANDs the carry flag with a specified bit in a general register or 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 a general register or memory operand and stores the result in the carry flag. The bit number is specified by 3-bit immediate data.
BOR	B	C ∨ (<bit-No.> of <EAd>) → C ORs the carry flag with a specified bit in a general register or 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 general register or memory operand and stores the result in the carry flag. The bit number is specified by 3-bit immediate data.

Instruction	Size*	Function
BXOR	B	C ⊕ (<bit-No.> of <EAd>) → C Exclusive-ORs the carry flag with a specified bit in a general register or memory operand and stores the result in the carry flag.
BIXOR	B	C ⊕ [¬ (<bit-No.> of <EAd>)] → C Exclusive-ORs the carry flag with the inverse of a specified bit in a general register or memory operand and stores the result in the carry flag. 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 operand to the carry flag.
BILD	B	¬ (<bit-No.> of <EAd>) → C Transfers the inverse of a specified bit in a general register or 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 general register or memory operand.
BIST	B	C → ¬ (<bit-No.> of <EAd>) Transfers the inverse of the carry flag value to a specified bit in a general register or memory operand. The bit number is specified by 3-bit immediate data.

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

Table 2.8 Branching Instructions

Instruction	Size	Function
Bcc	—	Branches to a specified address if a specified condition is true. The branching conditions are listed below.

Mnemonic	Description	Condition
BRA (BT)	Always (true)	Always
BRN (BF)	Never (false)	Never
BHI	High	$C \vee Z = 0$
BLS	Low or same	$C \vee Z = 1$
Bcc (BHS)	Carry clear (high or same)	$C = 0$
BCS (BLO)	Carry set (low)	$C = 1$
BNE	Not equal	$Z = 0$
BEQ	Equal	$Z = 1$
BVC	Overflow clear	$V = 0$
BVS	Overflow set	$V = 1$
BPL	Plus	$N = 0$
BMI	Minus	$N = 1$
BGE	Greater or equal	$N \oplus V = 0$
BLT	Less than	$N \oplus V = 1$
BGT	Greater than	$Z \vee (N \oplus V) = 0$
BLE	Less or equal	$Z \vee (N \oplus V) = 1$

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

Table 2.9 System Control Instructions

Instruction	Size*	Function
TRAPA	—	Starts trap-instruction exception handling
RTE	—	Returns from an exception-handling routine
SLEEP	—	Causes a transition to the power-down state
LDC	B/W	(EAs) → CCR Moves the source operand contents to the condition code register. The condition code register size is one byte, but in transfer from memory, data is read by word access.
STC	B/W	CCR → (EAd) Transfers the CCR contents to a destination location. The condition code register size is one byte, but in transfer to memory, data is written by word access.
ANDC	B	CCR ∧ #IMM → CCR Logically ANDs the condition code register with immediate data.
ORC	B	CCR ∨ #IMM → CCR Logically ORs the condition code register with immediate data.
XORC	B	CCR ⊕ #IMM → CCR Logically exclusive-ORs the condition code register with immediate data.
NOP	—	PC + 2 → PC Only increments the program counter.

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

Table 2.10 Block Transfer Instruction

Instruction	Size	Function
EEPMOV.B	—	if R4L ≠ 0 then repeat @ER5+ → @ER6+, R4L – 1 → R4L until R4L = 0 else next;
EEPMOV.W		if R4 ≠ 0 then repeat @ER5+ → @ER6+, R4 – 1 → R4 until R4 = 0 else next;
		Transfers a data block according to parameters set in general registers R4L or R4, ER5, and ER6. R4L or R4: Size of block (bytes) ER5: Starting source address ER6: Starting destination address Execution of the next instruction begins as soon as the transfer is completed.

2.6.4 Basic Instruction Formats

The H8/300H instructions consist of 2-byte (1-word) units. An instruction consists of an operation field (OP field), a register field (r field), an effective address extension (EA field), and a condition field (cc).

Operation Field: Indicates the function of the instruction, the addressing mode, and the operation to be carried out on the operand. The operation field always includes the first 4 bits of the instruction. Some instructions have two operation fields.

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

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

Condition Field: Specifies the branching condition of Bcc instructions.

Figure 2.9 shows examples of instruction formats.

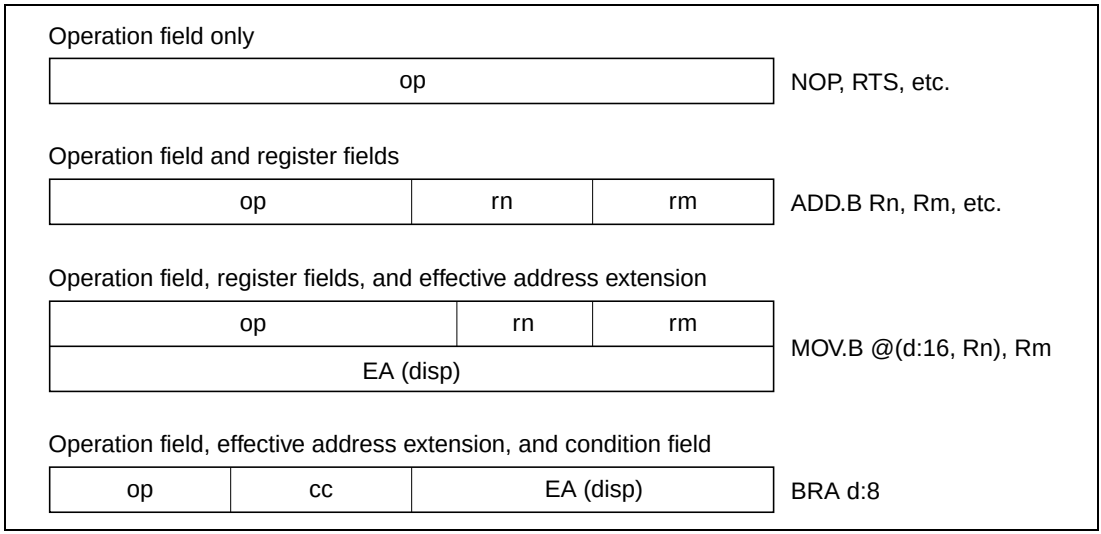


Figure 2.9 Instruction Formats

2.6.5 Notes on Use of Bit Manipulation Instructions

The BSET, BCLR, BNOT, BST, and BIST instructions read a byte of data, modify a bit in the byte, then write the byte back. Care is required when these instructions are used to access registers with write-only bits, or to access ports.

The BCLR instruction can be used to clear flags in the on-chip registers. In an interrupt-handling routine, for example, if it is known that the flag is set to 1, it is not necessary to read the 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 instruction uses a subset of these addressing modes. Arithmetic and logic instructions can use the register direct and immediate modes. Data transfer instructions can use all addressing modes except program-counter relative and memory indirect. Bit manipulation instructions use register direct, register indirect, or absolute (@aa:8) addressing mode to specify an operand, and register direct (BSET,

BCLR, BNOT, and BTST instructions) or immediate (3-bit) addressing mode to specify a bit number in the operand.

Table 2.11 Addressing Modes

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

1. Register Direct—Rn

The register field of the instruction code specifies an 8-, 16-, or 32-bit register containing the operand. R0H to R7H and R0L to R7L can be specified as 8-bit registers. R0 to R7 and E0 to E7 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 lower 24 bits of which contain the address of the operand.

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

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

4. Register Indirect with Post-Increment or Pre-Decrement—@ERn+ or @–ERn

- Register indirect with post-increment—@ERn+

  The register field of the instruction code specifies an address register (ERn) the lower 24 bits of which contain the address of a memory operand. After the operand is accessed, 1, 2, or 4 is added to the address register contents (32 bits) and the sum is stored in the address register. The value added is 1 for byte access, 2 for word access, or 4 for longword access. For word or longword access, the register value should be even.

- Register indirect with pre-decrement—@–ERn

  The value 1, 2, or 4 is subtracted from an address register (ERn) specified by the register field in the instruction code, and the lower 24 bits of the result become the address of a memory operand. The result is also stored in the address register. The value subtracted is 1 for byte access, 2 for word access, or 4 for longword access. For word or longword access, the resulting 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 address may be 8 bits long (@aa:8), 16 bits long (@aa:16), or 24 bits long (@aa:24). For an 8-bit absolute address, the upper 16 bits are all assumed to be 1 (H'FFFF). For a 16-bit absolute address the upper 8 bits are a sign extension. A 24-bit absolute address can access the entire address space. 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 (1,048,320 to 1,048,575)	H'FFFF00 to H'FFFFFF (16,776,960 to 16,777,215)
16 bits (@aa:16)	H'00000 to H'07FFF, H'F8000 to H'FFFFF (0 to 32,767, 1,015,808 to 1,048,575)	H'000000 to H'007FFF, H'FF8000 to H'FFFFFF (0 to 32,767, 16,744,448 to 16,777,215)
24 bits (@aa:24)	H'00000 to H'FFFFF (0 to 1,048,575)	H'000000 to H'FFFFFF (0 to 16,777,215)

6. Immediate—#xx:8, #xx:16, or #xx:32

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

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

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

This mode is used in the Bcc and BSR instructions. An 8-bit or 16-bit displacement contained in the instruction code is sign-extended to 24 bits and added to the 24-bit PC contents to generate a 24-bit branch address. The PC value to which the displacement is added is the address of the first byte of the next instruction, so the possible branching range is –126 to +128 bytes (–63 to +64 words) or –32766 to +32768 bytes (–16383 to +16384 words) from the branch instruction. The 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 contains an 8-bit absolute address specifying a memory operand. This memory operand contains a branch address. The memory operand is accessed by longword access. The first byte of the memory operand is ignored, generating a 24-bit branch address. See figure 2.10. The upper bits of the 8-bit absolute address are assumed to be 0 (H'0000), so the address range is 0 to 255 (H'000000 to H'0000FF). Note that the first part of this range is also the exception vector area. For further details see section 5, Interrupt Controller.

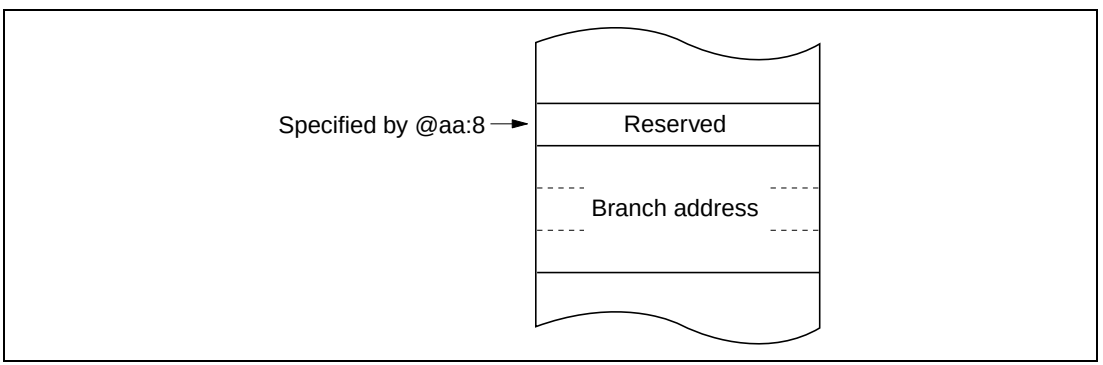


Figure 2.10 Memory-Indirect Branch Address Specification

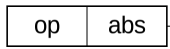
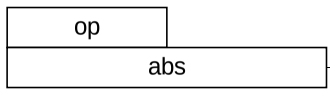
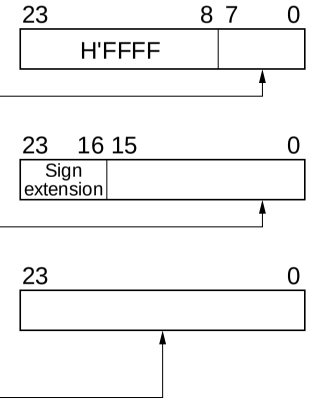
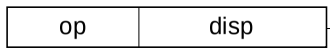
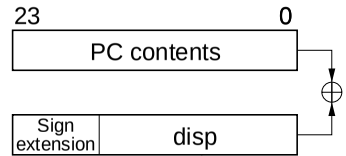
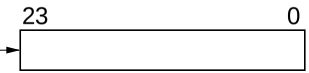
When a word-size or longword-size memory operand is specified, or when a branch address is specified, if the specified memory address is odd, the least significant bit is regarded as 0. The accessed data or instruction code therefore begins at the preceding address. See section 2.5.2, Memory Data Formats.

2.7.2 Effective Address Calculation

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

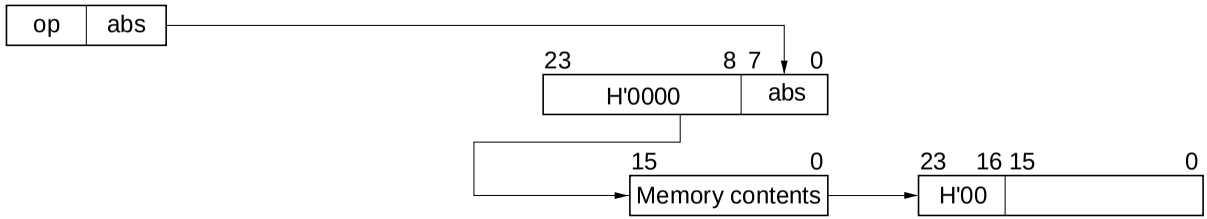
Table 2.13 Effective Address Calculation

No.	Addressing Mode and Instruction Format	Effective Address Calculation	Effective Address
1	Register direct (Rn) op | rm | rn		Operand is general register contents
2	Register indirect (@ERn) op | r	31 General register contents 0	23 0
3	Register indirect with displacement @(d:16, ERn)/@(d:24, ERn) op | r | disp	31 General register contents 0 Sign extension | disp	23 0
4	Register indirect with post-increment or pre-decrement Register indirect with post-increment @ERn+ op | r Register indirect with pre-decrement @−ERn op | r	31 General register contents 0 1, 2, or 4 31 General register contents 0 1, 2, or 4 1 for a byte operand, 2 for a word operand, 4 for a longword operand	23 0 23 0

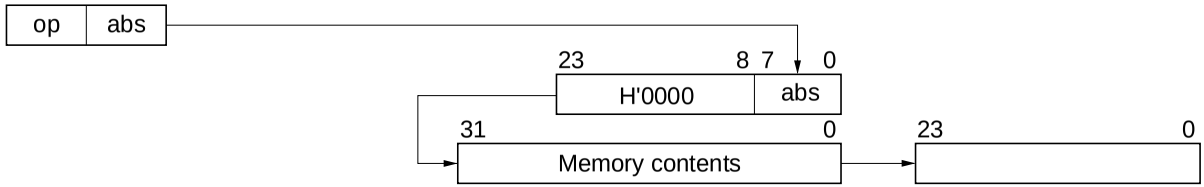
No.	Addressing Mode and Instruction Format	Effective Address Calculation	Effective Address
5	Absolute address @aa:8  @aa:16  @aa:24 		
6	Immediate #xx:8, #xx:16, or #xx:32 		Operand is immediate data
7	Program-counter relative @(d:8, PC) or @(d:16, PC) 		

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

Normal mode



Advanced mode



Legend:

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

2.8 Processing States

2.8.1 Overview

The H8/300H CPU has four processing states: the program execution state, exception-handling state, power-down state, and reset state. The power-down state includes sleep mode, software standby mode, and hardware standby mode. Figure 2.11 classifies the processing states. Figure 2.13 indicates the state transitions.

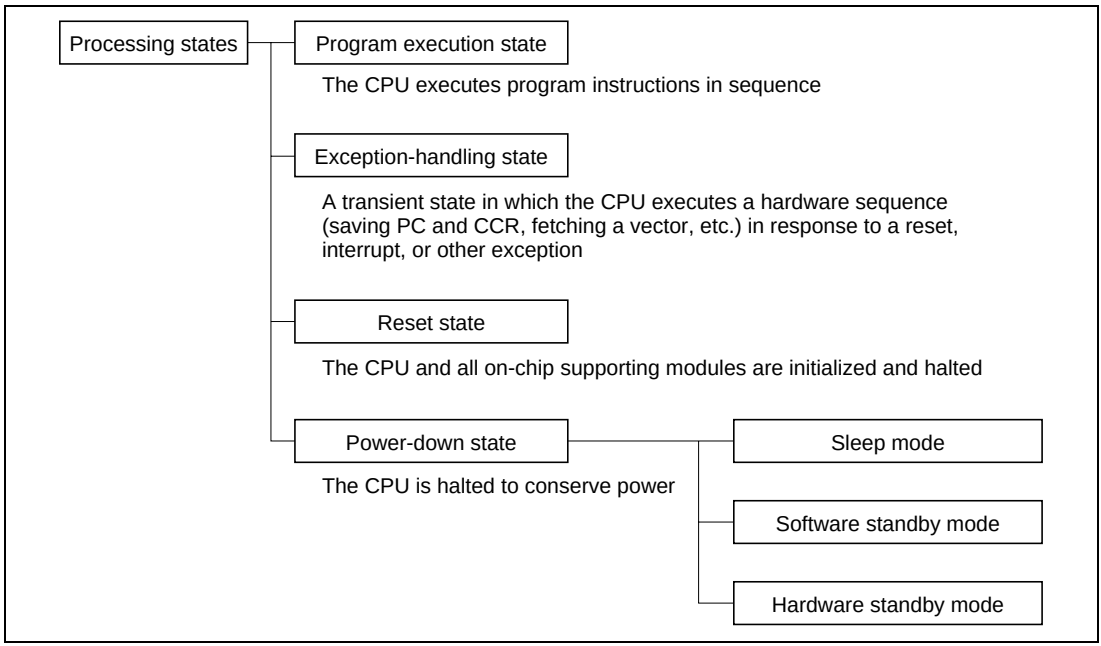


Figure 2.11 Processing States

2.8.2 Program Execution State

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

2.8.3 Exception-Handling State

The exception-handling state is a transient state that occurs when the CPU alters the normal program flow due to a reset, interrupt, or trap instruction. The CPU fetches a starting address from the exception vector table and branches to that address. In interrupt and trap exception handling the CPU references the stack pointer (ER7) and saves the program counter and condition code register.

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

Table 2.14 Exception Handling Types and Priority

Priority	Type of Exception	Detection Timing	Start of Exception Handling
High ▲	Reset	Synchronized with clock	Exception handling starts immediately when $\overline{\text{RES}}$ changes from low to high
	Interrupt	End of instruction execution or end of exception handling*	When an interrupt is requested, exception handling starts at the end of the current instruction or current exception-handling sequence
	Trap instruction	When TRAPA instruction is executed	Exception handling starts when a trap (TRAPA) instruction is executed
Low			

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

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

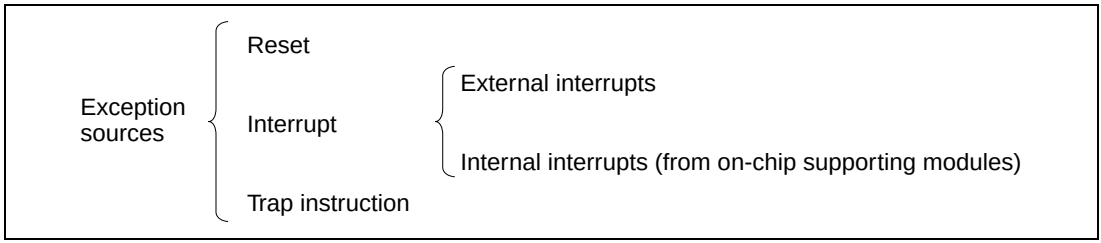
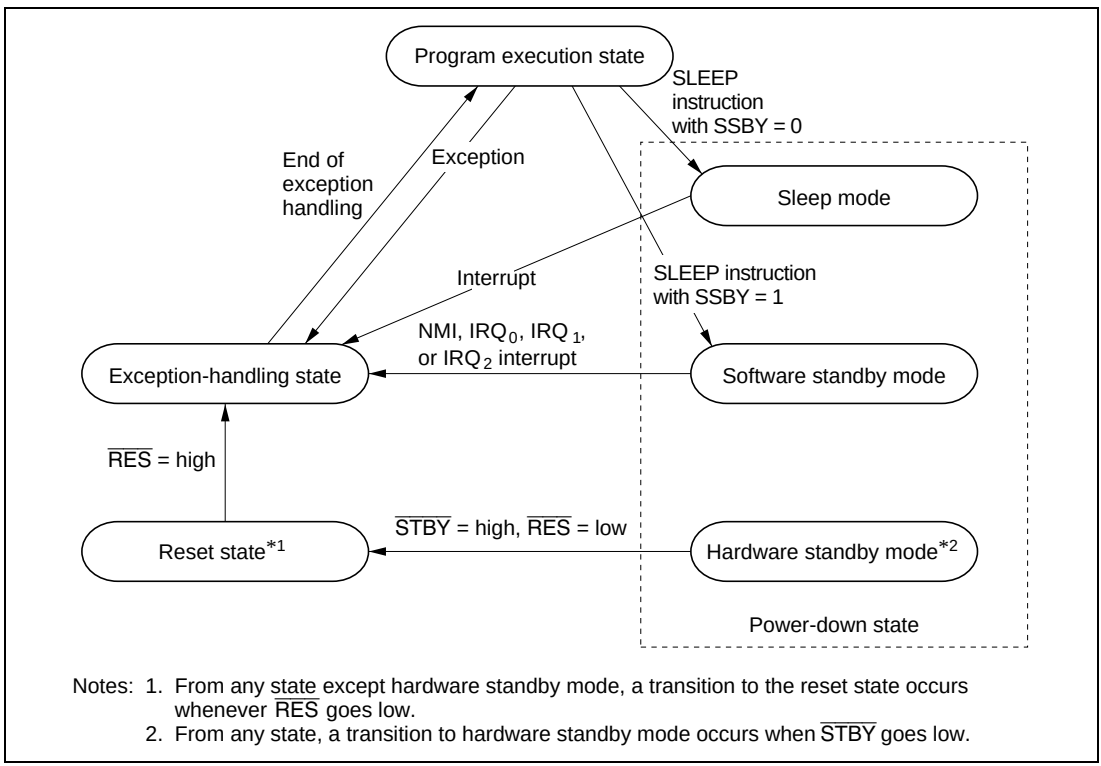


Figure 2.12 Classification of Exception Sources



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 reset state is entered when the $\overline{RES}$ signal goes low. Reset exception handling starts after that, when $\overline{RES}$ changes from low to high. When reset exception handling starts the CPU fetches a start address from the exception vector table and starts program execution from that address. All interrupts,

including NMI, are disabled during the reset exception-handling sequence and immediately after it ends.

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

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

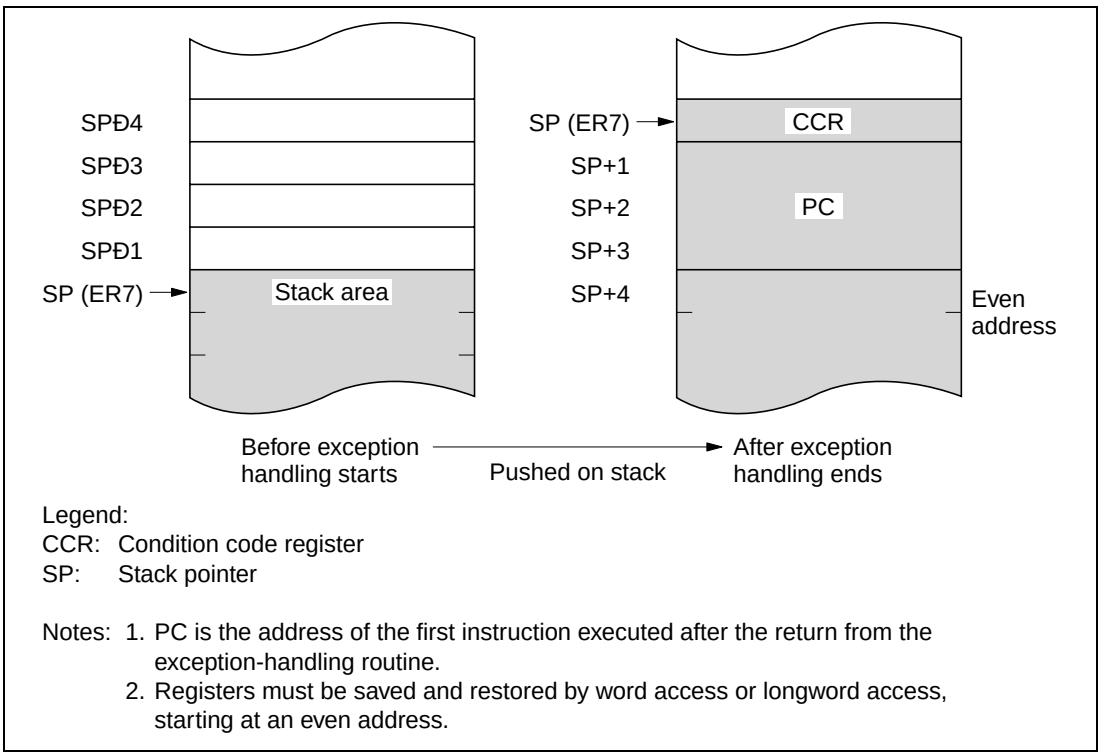


Figure 2.14 Stack Structure after Exception Handling

2.8.5 Reset State

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

The reset state can also be entered by a watchdog timer overflow. For details see section 10, Watchdog Timer.

2.8.6 Power-Down State

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

Sleep Mode: A transition to sleep mode is made if the SLEEP instruction is executed while the 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 registers are 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 halt and all on-chip supporting modules stop operating. The on-chip supporting modules are reset, but as long as a specified voltage is supplied the contents of CPU registers and on-chip RAM are retained. The I/O ports also remain in their existing states.

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

For further information see section 17, Power-Down State.

2.9 Basic Operational Timing

2.9.1 Overview

The H8/300H CPU operates according to the system clock (ϕ). The interval from one rise of the system clock to the next rise is referred to as a "state." A memory cycle or bus cycle consists of two or three states. The CPU uses different methods to access on-chip memory, the on-chip supporting modules, and the external address space. Access to the external address space can be 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 both byte and word access. Figure 2.15 shows the on-chip memory access cycle. Figure 2.16 indicates the pin states.

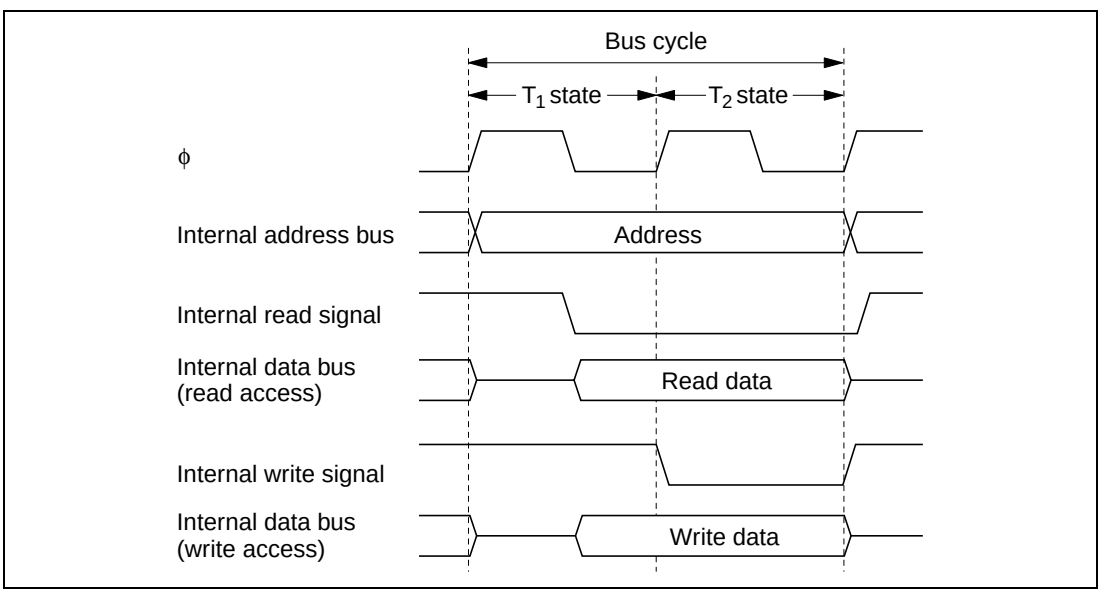


Figure 2.15 On-Chip Memory Access Cycle

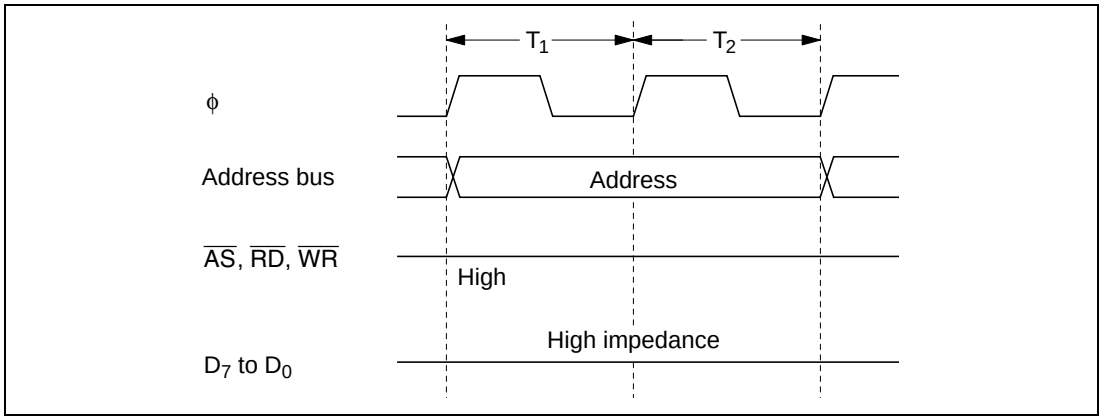


Figure 2.16 Pin States during On-Chip Memory Access

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 bits wide, depending on the register being accessed. Figure 2.17 shows the on-chip supporting module access timing. Figure 2.18 indicates the pin states.

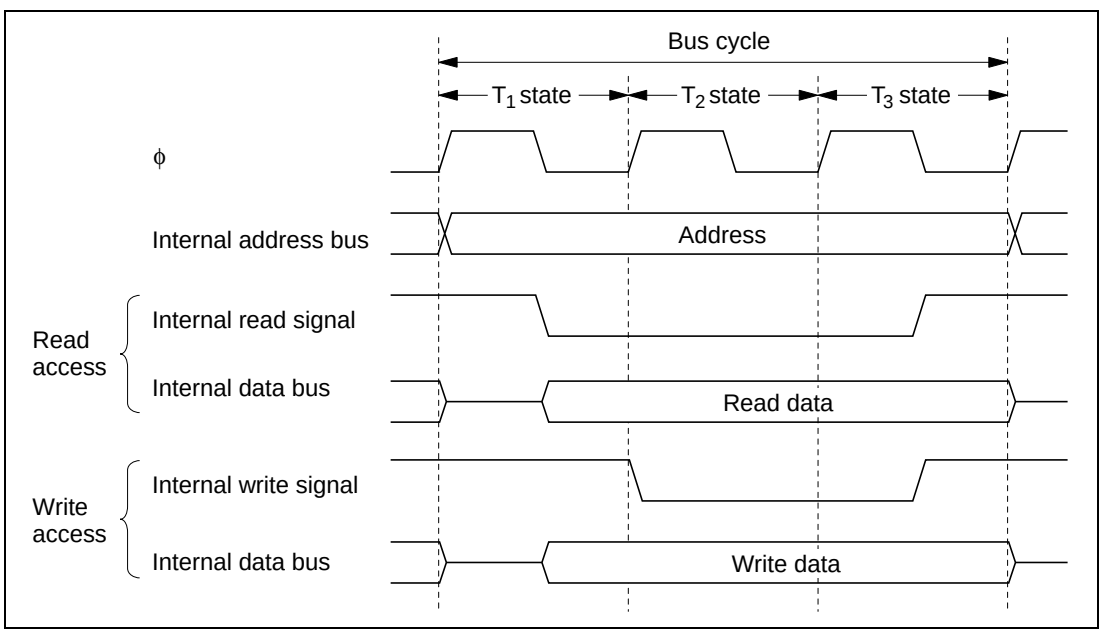


Figure 2.17 Access Cycle for On-Chip Supporting Modules

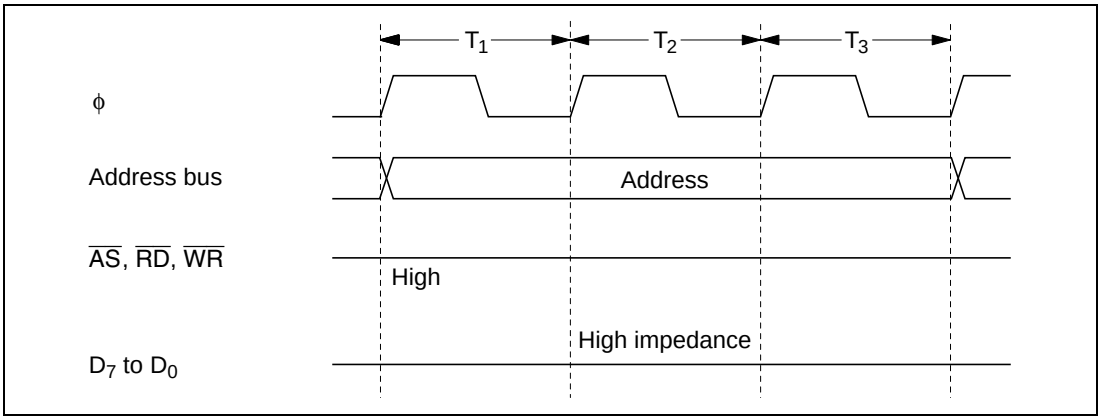


Figure 2.18 Pin States during Access to On-Chip Supporting Modules

2.9.4 Access to External Address Space

The external address space is divided into eight areas (areas 0 to 7). Bus-controller settings determine whether each area accessed in two or three states. For details see section 6, Bus Controller.

Section 3 MCU Operating Modes

3.1 Overview

3.1.1 Operating Mode Selection

The H8/3039 Group has five operating modes (modes 1, 3, 5 to7) that are selected by the mode pins (MD_2 to MD_0) as indicated in table 3.1. The input at these pins determines expanded mode or single-chip mode.

Table 3.1 Operating Mode Selection

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

Notes: 1. If the RAM enable bit (RAME) in the system control register (SYSCR) is cleared to 0, these addresses become external addresses.
2. In mode 6 and 7, clearing bit RAME in SYSCR to 0 and reading the on-chip RAM always return H'FF, and write access is ignored. For details, see section 14.3, Operation.

For the address space size there are three choices: 64 kbytes, 1 Mbyte, or 16 Mbytes.

Modes 1 and 3 are on-chip ROM disable expanded modes capable of accessing external memory and peripheral devices.

Mode 1 supports a maximum address space of 1 Mbyte.

Mode 3 supports a maximum address space of 16 Mbytes.

Mode 5 is externally expanded mode that enables access to external memory and peripheral devices and also enables access to the on-chip ROM. Mode 5 supports a maximum address space of 1 Mbyte.

Modes 6 and 7 are single-chip modes that operate using the on-chip ROM, RAM, and registers. All I/O ports are available. Mode 6 is a normal mode with 64-kbyte address space. Mode 7 is an advanced mode with a maximum address space of 1 Mbyte.

The H8/3039 Group can be used only in modes 1, 3, or 5 to 7. The inputs at the mode pins must select one of these seven modes. The inputs at the mode pins must not be changed during operation.

3.1.2 Register Configuration

The H8/3039 Group has a mode control register (MDCR) that indicates the inputs at the mode pins (MD_2 to MD_0), and a system control register (SYSCR). Table 3.2 summarizes these registers.

Table 3.2 Registers

Address*	Name	Abbreviation	R/W	Initial Value
H'FFF1	Mode control register	MDCR	R	Undetermined
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 H8/3039 Group.

Bit	7	6	5	4	3	2	1	0
	—	—	—	—	—	MDS2	MDS1	MDS0
Initial value	1	1	0	0	0	—*	—*	—*
Read/Write	—	—	—	—	—	R	R	R

Reserved bits

Mode select 2 to 0
Bits indicating the current operating mode

Note: Determined by pins MD_2 to MD_0.

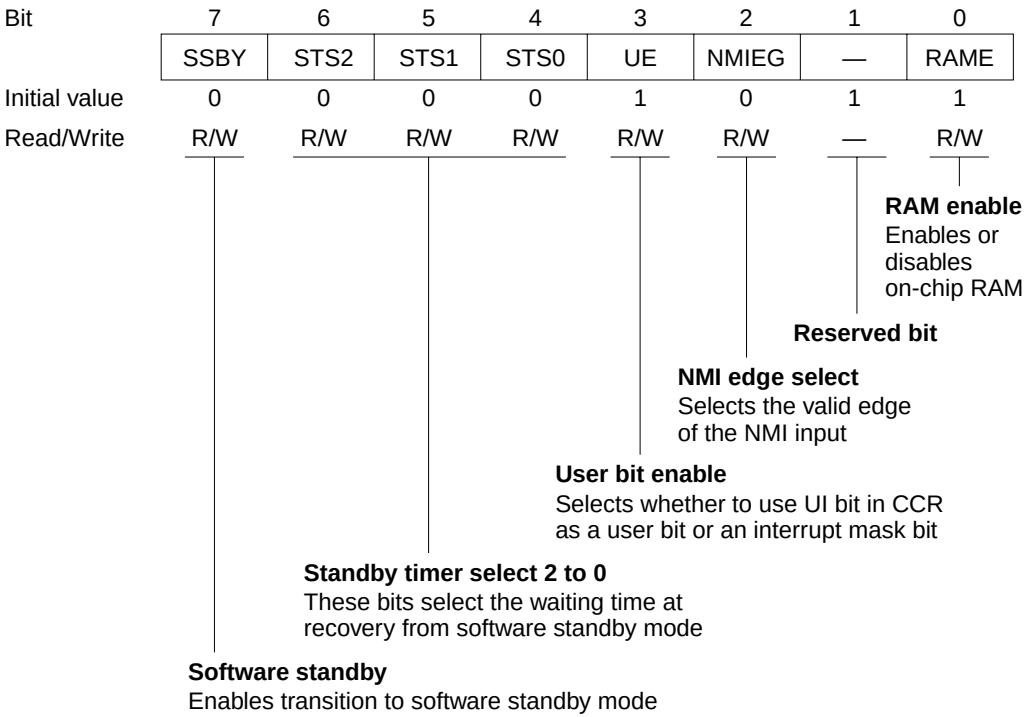
Bits 7 and 6—Reserved: These bits cannot be modified and are always read as 1.

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

Bits 2 to 0—Mode Select 2 to 0 (MDS_2 to MDS_0): These bits indicate the logic levels at pins MD_2 to MD_0 (the current operating mode). MDS_2 to MDS_0 correspond to MD_2 to MD_0. MDS_1 and MDS_0 are read-only bits. The mode pin (MD_2 to MD_0) levels are latched when MDCR is read.

3.3 System Control Register (SYSCR)

SYSCR is an 8-bit register that controls the operation of the H8/3039 Group.



Bit 7—Software Standby (SSBY): Enables transition to software standby mode. (For further information about software standby mode see section 17, Power-Down State.)

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

Bit 7 SSBY	Description	
0	SLEEP instruction causes transition to sleep mode	(Initial value)
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 time the CPU and on-chip supporting modules wait for the internal clock oscillator to settle when software standby mode is exited by an external interrupt. Set these bits so that the waiting time will be at

least 7 ms at the system clock rate. For further information about waiting time selection, see section 17.4.3, Selection of Oscillator Waiting Time after Exit from Software Standby Mode.

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

Bit 3—User Bit Enable (UE): Selects whether to use the UI bit in the condition code register as a 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	(Initial value)

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

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

Bit 1—Reserved: This bit cannot be modified and is always read as 1.

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

Bit 0 RAME	Description	
0	On-chip RAM is disabled	
1	On-chip RAM is enabled	(Initial value)

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

3.4.2 Mode 3

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

3.4.3 Mode 5

Ports 1, 2, and 5 can function as address pins A_{19} to A_0, permitting access to a maximum 1-Mbyte address space, but following a reset they are input ports. To use ports 1, 2, and 5 as an address bus, the corresponding bits in their data direction registers (P1DDR, P2DDR, and P5DDR) must be set to 1. The address bus width can be selected freely by setting DDR of ports 1, 2, and 5. The initial bus mode after a reset is 8 bits, with 8-bit access to all areas.

3.4.4 Mode 6

This mode operates using the on-chip ROM, RAM, and registers. All I/O ports are available. Mode 6 is a normal mode with 64-kbyte address space.

3.4.5 Mode 7

This mode is an advanced mode with a 1-Mbyte address space which operates using the on-chip ROM, RAM, and registers. All I/O ports are available.

Note: The H8/3039 Group cannot be used in mode 2 and 4.

3.5 Pin Functions in Each Operating Mode

The pin functions of ports 1 to 3, port 5 and port A vary depending on the operating mode. Table 3.3 indicates their functions in each operating mode.

Table 3.3 Pin Functions in Each Mode

Port	Mode 1	Mode 2*[1]	Mode 3	Mode 4*[1]	Mode 5	Mode 6	Mode 7
Port 1	A_7 to A_0	—	A_7 to A_0	—	$P1_7$ to $P1_0$*[2]	$P1_7$ to $P1_0$	$P1_7$ to $P1_0$
Port 2	A_{15} to A_8	—	A_{15} to A_8	—	$P2_7$ to $P2_0$*[2]	$P2_7$ to $P2_0$	$P2_7$ to $P2_0$
Port 3	D_7 to D_0	—	D_7 to D_0	—	D_7 to D_0	$P3_7$ to $P3_0$	$P3_7$ to $P3_0$
Port 5	A_{19} to A_{16}	—	A_{19} to A_{16}	—	$P5_3$ to $P5_0$*[2]	$P5_3$ to $P5_0$	$P5_3$ to $P5_0$
Port A	PA_7 to PA_4	—	PA_6 to PA_4*[3], A_{20}	—	PA_7 to PA_4	PA_7 to PA_4	PA_7 to PA_4

Notes: 1. H8/3039 Group cannot be used in these modes.

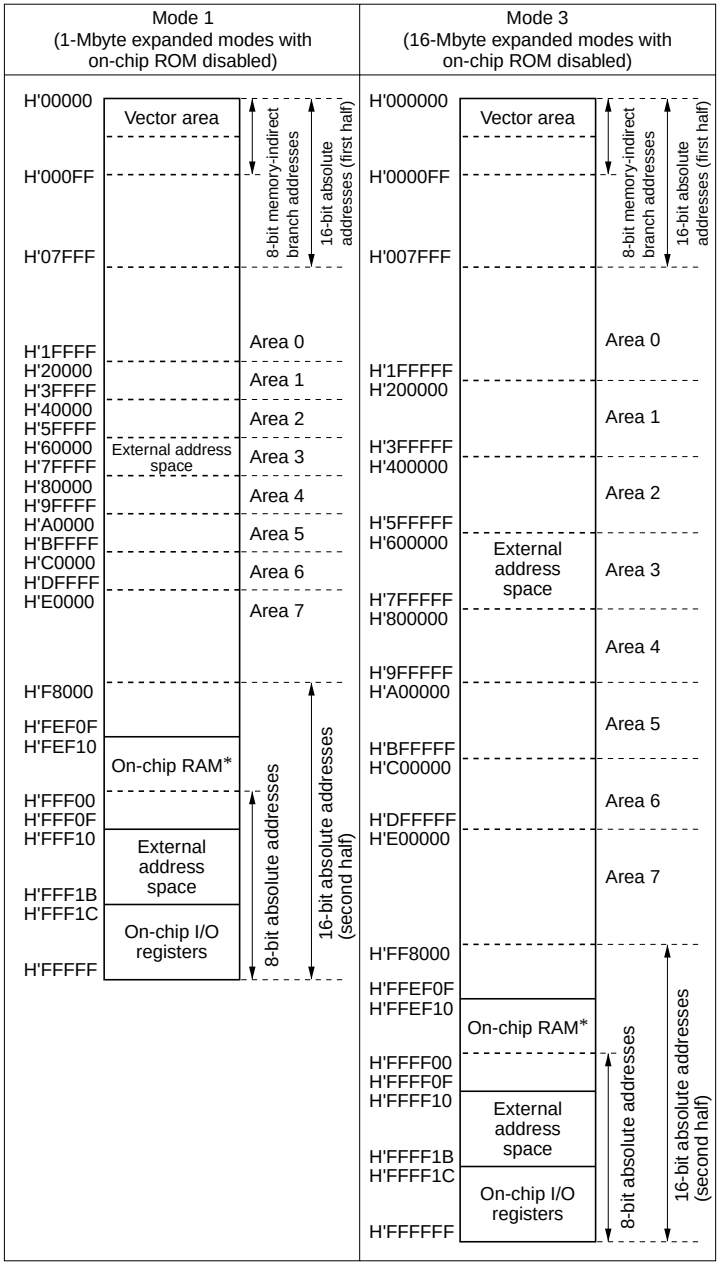
2. Initial state. These pins become address output pins when the corresponding bits in the data direction registers (P1DDR, P2DDR, P5DDR) are set to 1.
3. Initial state A_{20} is always an address output pin. PA_6 to PA_4 are switched over to A_{23} to A_{21} output by writing 0 in bits 7 to 5 of ADRCR.

3.6 Memory Map in Each Operating Mode

Figure 3.1 shows a memory map of the H8/3039. Figure 3.2 shows a memory map of the H8/3038. Figure 3.3 shows a memory map of the H8/3037. Figure 3.4 shows a memory map of the H8/3036. The address space is divided into eight areas.

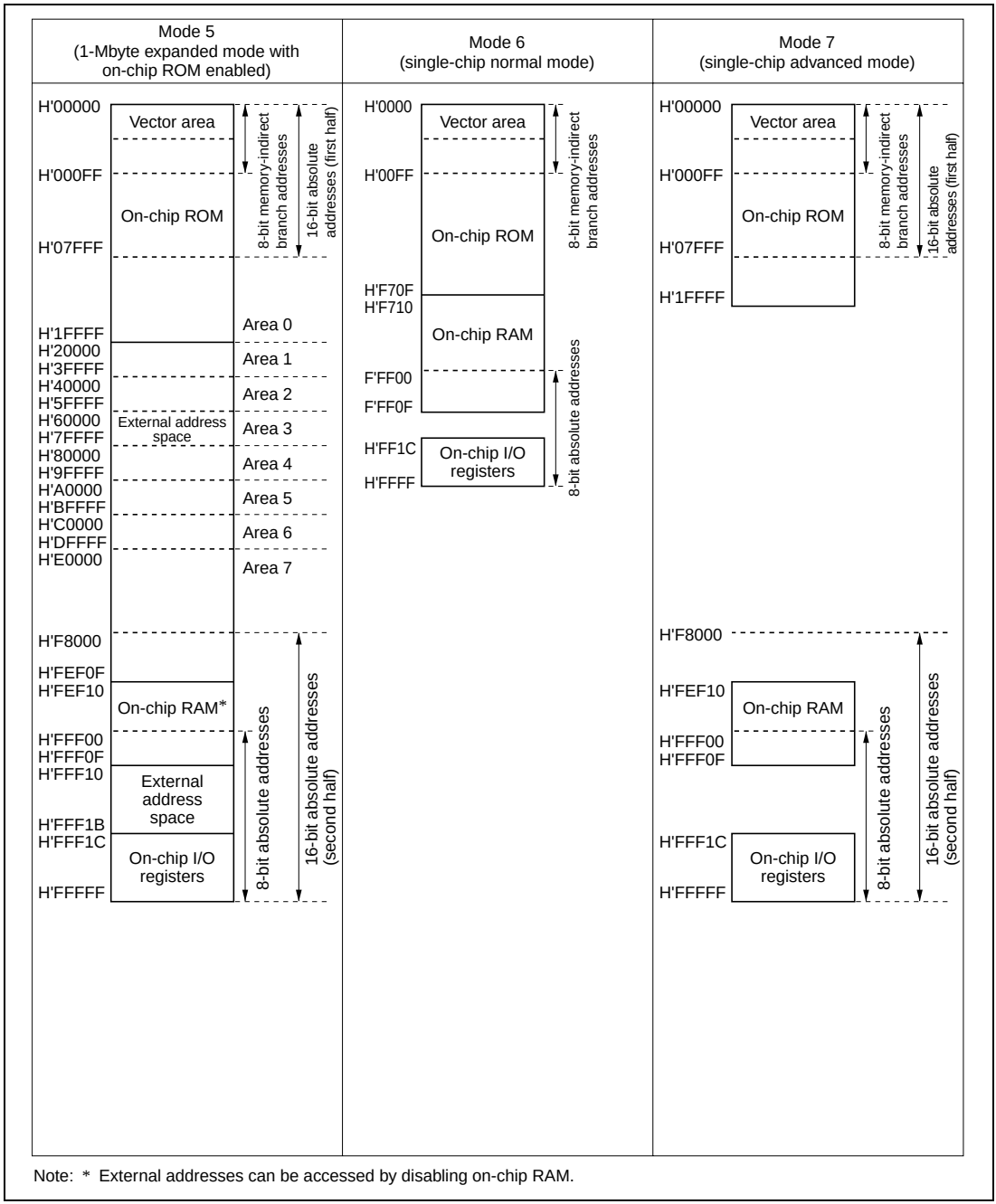
Modes 1, 3 and 5 are the 8-bit bus mode.

The address locations of the on-chip RAM and on-chip registers differ between the 1-Mbyte modes (modes 1, 5, and 7) and 16-Mbyte mode (mode 3), and 64-kbyte mode (mode 6). The address range specifiable by the CPU in the 8- and 16-bit absolute addressing modes (@aa:8 and @aa:16) also differs.



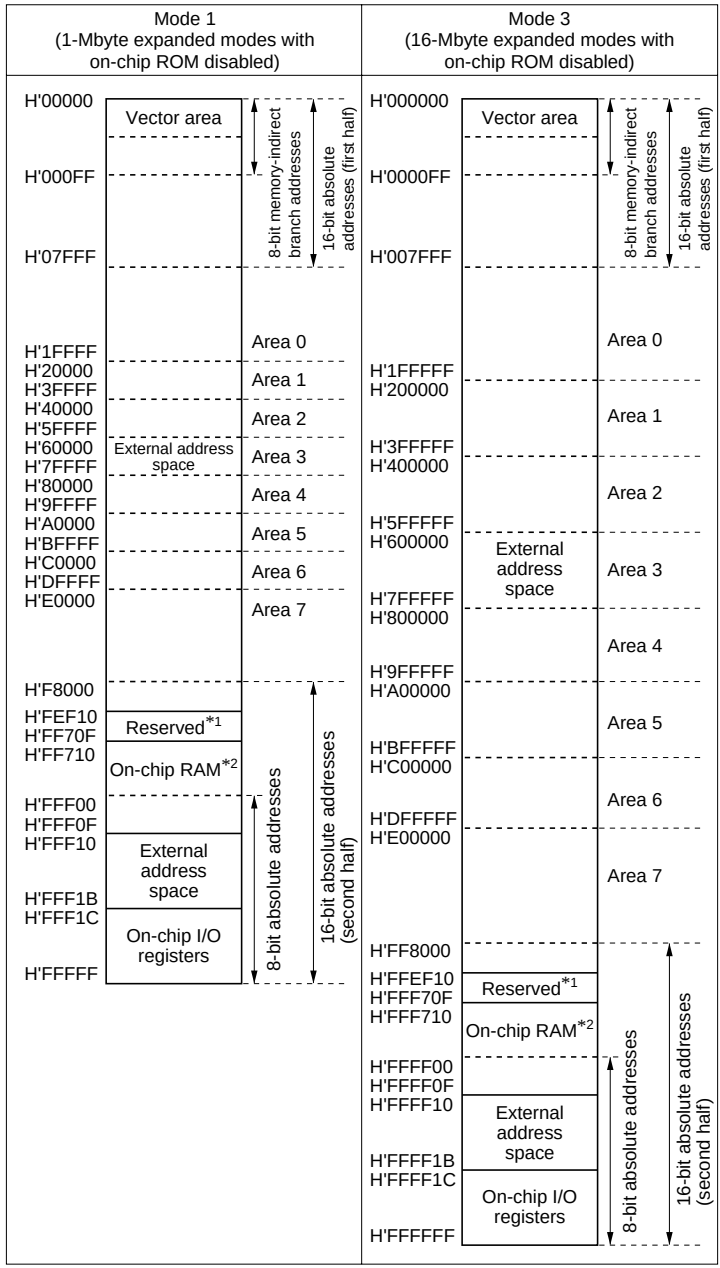
Note: * External addresses can be accessed by disabling on-chip RAM.

Figure 3.1 H8/3039 Memory Map in Each Operating Mode (1)



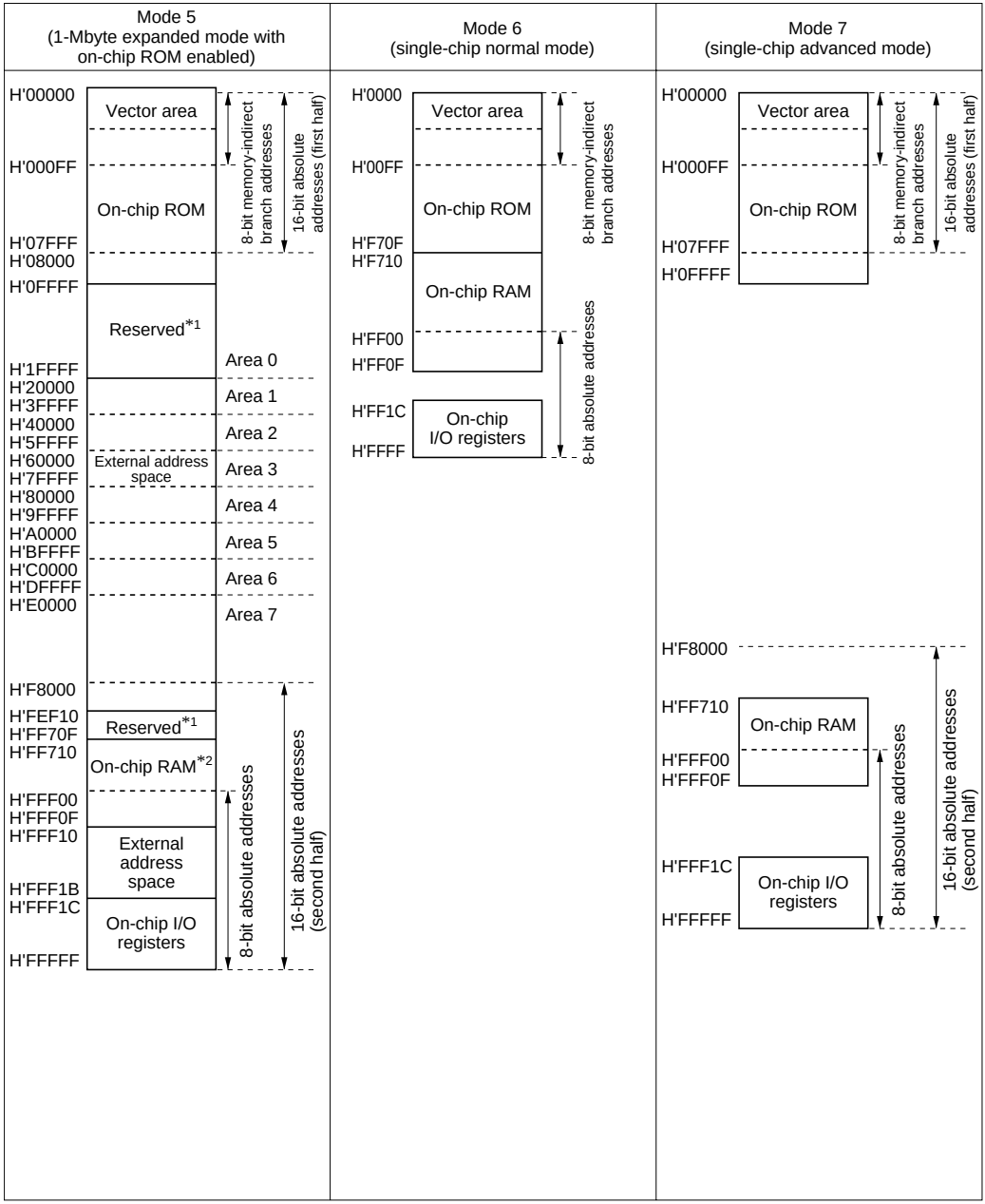
Note: * External addresses can be accessed by disabling on-chip RAM.

Figure 3.1 H8/3039 Memory Map in Each Operating Mode (2)



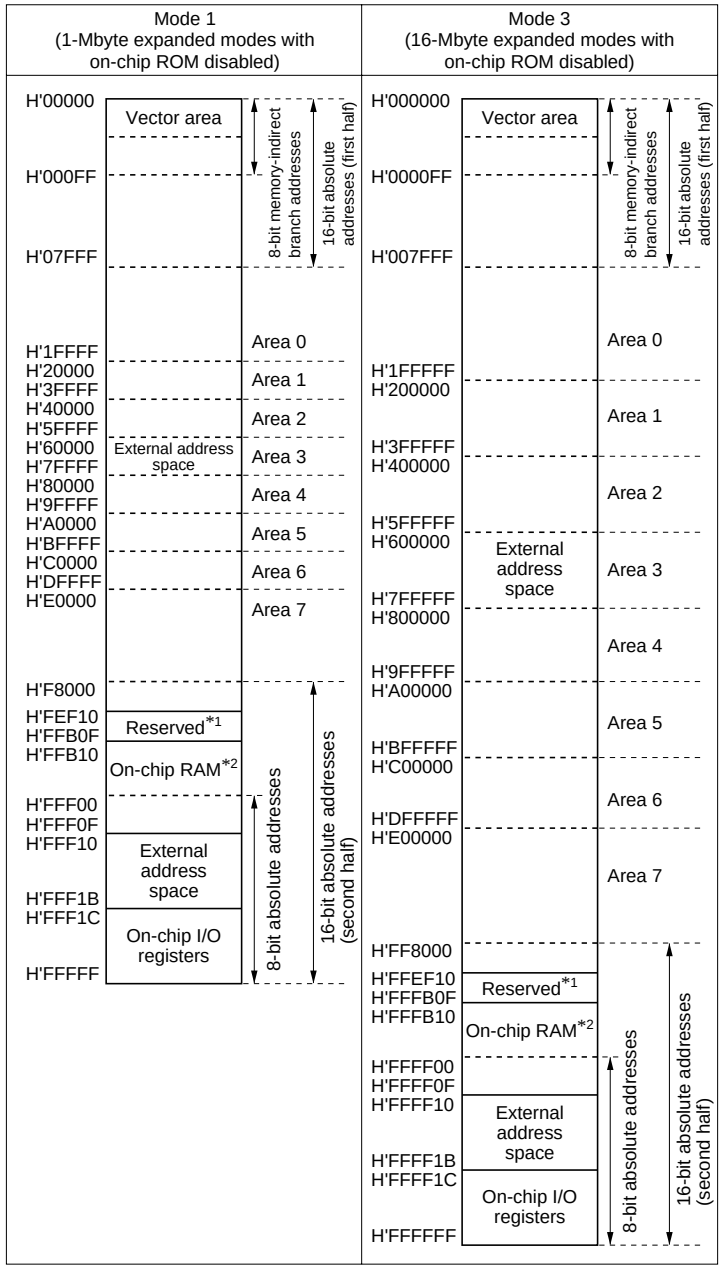
Notes: 1. Do not access the reserved area.
2. External addresses can be accessed by disabling on-chip RAM.

Figure 3.2 H8/3038 Memory Map in Each Operating Mode (1)



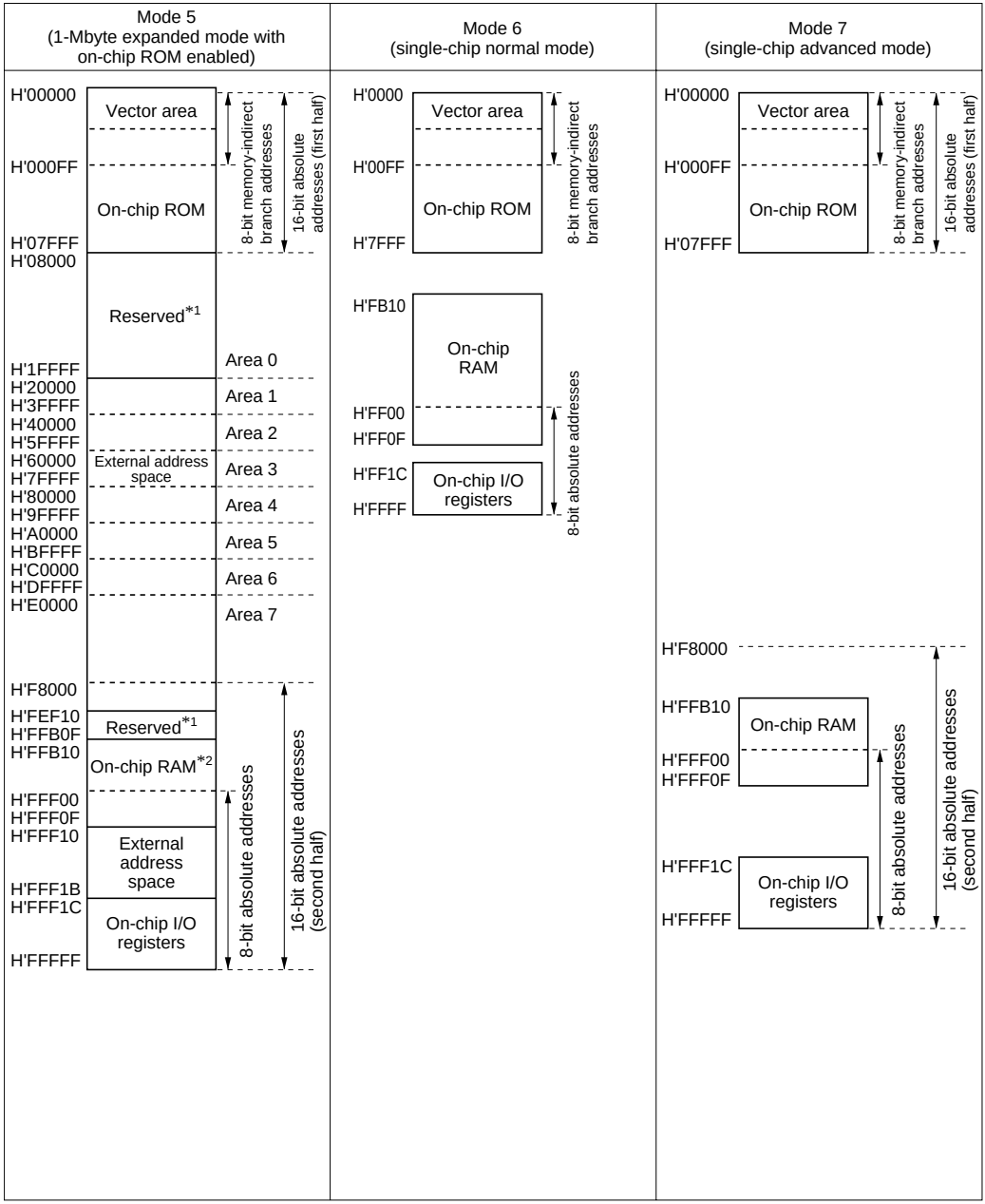
Notes: 1. Do not access the reserved area.
2. External addresses can be accessed by disabling on-chip RAM.

Figure 3.2 H8/3038 Memory Map in Each Operating Mode (2)



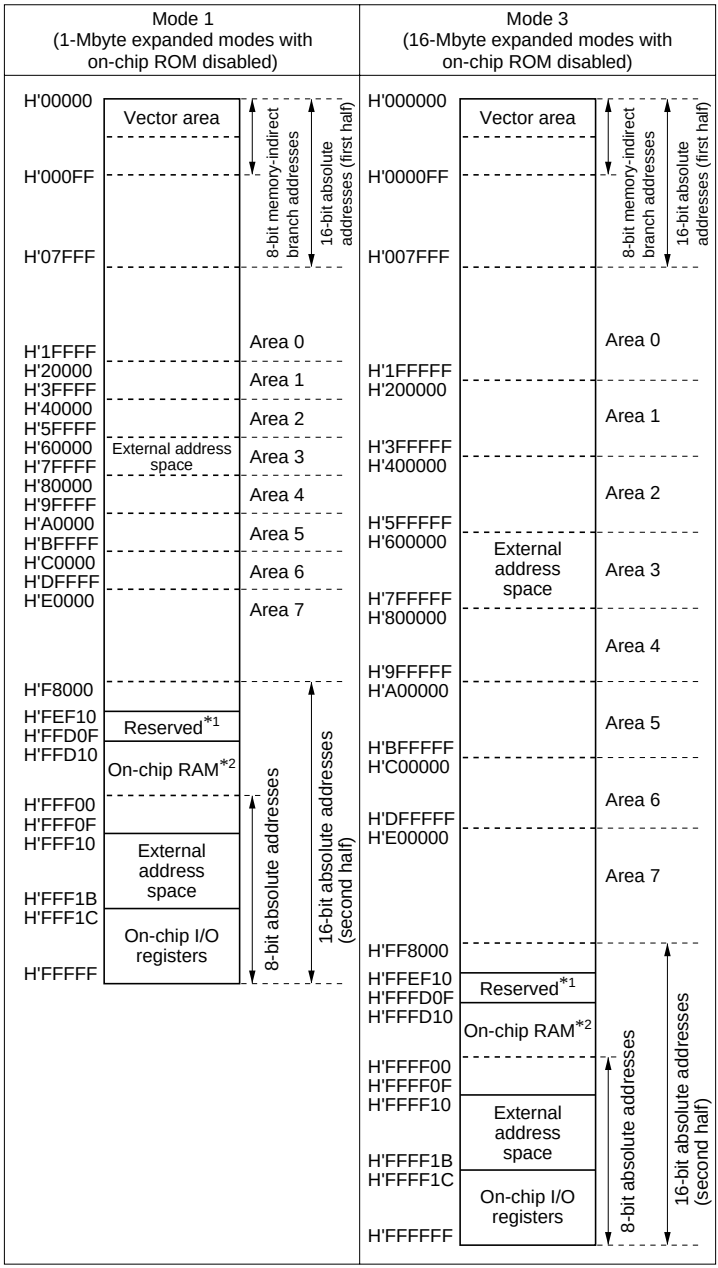
Notes: 1. Do not access the reserved area.
2. External addresses can be accessed by disabling on-chip RAM.

Figure 3.3 H8/3037 Memory Map in Each Operating Mode (1)



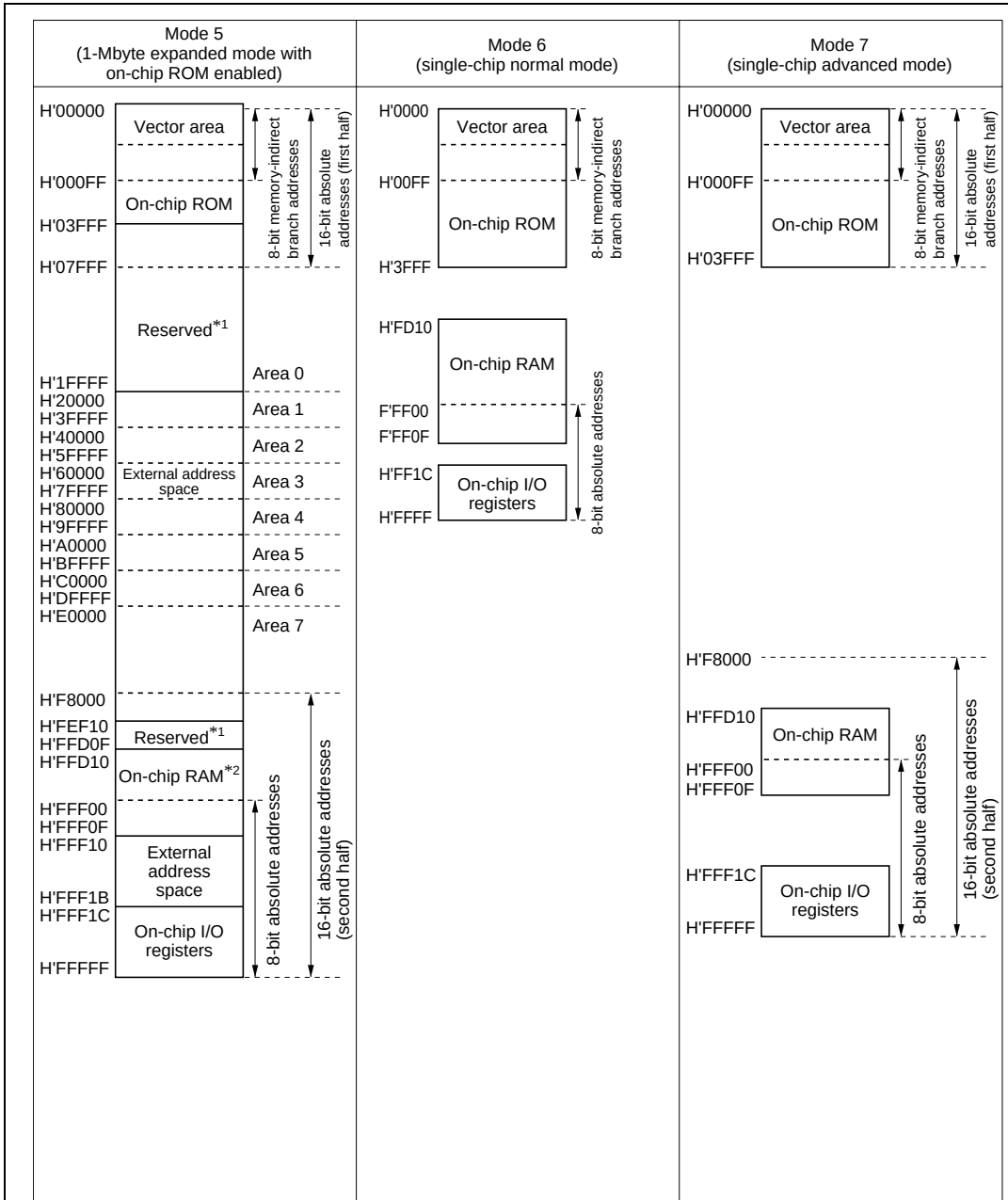
Notes: 1. Do not access the reserved area.
2. External addresses can be accessed by disabling on-chip RAM.

Figure 3.3 H8/3037 Memory Map in Each Operating Mode (2)



Notes: 1. Do not access the reserved area.
2. External addresses can be accessed by disabling on-chip RAM.

Figure 3.4 H8/3036 Memory Map in Each Operating Mode (1)



Notes: 1. Do not access the reserved area.
2. External addresses can be accessed by disabling on-chip RAM.

Figure 3.4 H8/3036 Memory Map in Each Operating Mode (2)

3.7 Restrictions on Use of Mode 6

In mode 6 (single-chip normal mode), on-chip ROM area data is undefined if address H'10000 or above (64 kbytes or above) is accessed, and therefore instruction code fetch and data read operations may not always be performed normally.

However, there is no problem with address H'10000 and above if the lower 16-bit address is an on-chip RAM (H'F710 to H'FF0F) or internal I/O register (H'FF1C to H'FFFF) address.

Table 3.4 shows the restrictions concerning each addressing mode.

Table 3.4 Access Restrictions in Mode 6 (Single-Chip Normal Mode)

	Conditions			
Addressing Mode	**Restricted Item**	**Address Range**	**Operation**	**Restriction**
Register direct (Rn)	—	—	No problem	—
Register indirect (@ERn)	Contents of ERn	H'00010000 or above, with lower 16 bits in range H'0000 to H'F710	Read data is undefined. Writes are invalid.	Set upper 16 bits of ERn to H'0000; or, write same data as in H'00000–H'0FFFF to H'10000–H'1FFFF in on-chip ROM.
Register indirect with displacement (@(d:16,ERn), @(d:16,ERn))	Value of ERn contents plus displacement			
Register indirect with post-increment (@ERn+) Register indirect with pre-decrement (@ERn-)	Value of ERn contents incremented (or decremented) by 1, 2, or 4			
Absolute address (@aa:8)	—	—	No problem	—
Absolute address (@aa:16)	Value of @aa sign-extended to 24 bits	H'010000 or above, with lower 16 bits in range H'0000 to H'F710	Read data is undefined. Writes are invalid.	Do not specify H'8000 or above as absolute address; or, write same data as in H'00000–H'0FFFF to H'10000–H'1FFFF in on-chip ROM.

	Conditions			
Addressing Mode	**Restricted Item**	**Address Range**	**Operation**	**Restriction**
Absolute address (@aa:24)	Value of @aa	H'010000 or above, with lower 16 bits in range H'0000 to H'F710	Read data is undefined. Writes are invalid.	Do not access addresses in range shown under conditions; or, write same data as in H'00000–H'0FFFF to H'10000–H'1FFFF in on-chip ROM.
Immediate	—	—	No problem	—
Program-counter relative (@(d:8,PC), @(d:16,PC))	Value of PC plus displacement	H'010000 or above, with lower 16 bits in range H'0000 to H'F710	Does not operate normally since instruction code is undefined.	Do not access addresses in range shown under conditions; or, write same data as in H'00000–H'0FFFF to H'10000–H'1FFFF in on-chip ROM.
Memory indirect (@@aa:8)	—	—	No problem	—

Section 4 Exception Handling

4.1 Overview

4.1.1 Exception Handling Types and Priority

As table 4.1 indicates, exception handling may be caused by a reset, trap instruction, or interrupt. Exception handling is prioritized as shown in table 4.1. If two or more exceptions occur simultaneously, they are accepted and processed in priority order. Trap instruction exceptions are accepted at all times in the program execution state.

Table 4.1 Exception Types and Priority

Priority	Exception Type	Start of Exception Handling
High ▲	Reset	Starts immediately after a low-to-high transition at the $\overline{\text{RES}}$ pin
	Interrupt	Interrupt requests are handled when execution of the current instruction or handling of the current exception is completed
Low	Trap instruction (TRAPA)	Started by execution of a trap instruction (TRAPA)

4.1.2 Exception Handling Operation

Exceptions originate from various sources. Trap instructions and interrupts are handled as follows.

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

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

4.1.3 Exception Vector Table

The exception sources are classified as shown in figure 4.1. Different vectors are assigned to different exception sources. Table 4.2 lists the exception sources and their vector addresses.

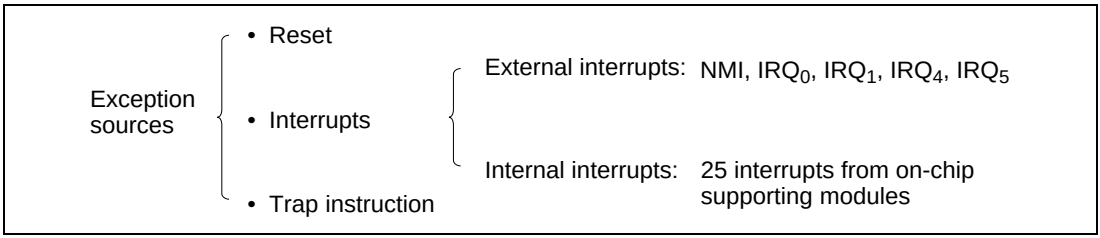


Figure 4.1 Exception Sources

Table 4.2 Exception Vector Table

Exception Source		Vector Number	Vector Address*1	
			Normal Mode	Advanced Mode
Reset		0	H'0000 to H'0001	H'0000 to H'0003
Reserved for system use		1	H'0002 to H'0003	H'0004 to H'0007
		2	H'0004 to H'0005	H'0008 to H'000B
		3	H'0006 to H'0007	H'000C to H'000F
		4	H'0008 to H'0009	H'0010 to H'0013
		5	H'000A to H'000B	H'0014 to H'0017
		6	H'000C to H'000D	H'0018 to H'001B
External interrupt (NMI)		7	H'000E to H'000F	H'001C to H'001F
Trap instruction (4 sources)		8	H'0010 to H'0011	H'0020 to H'0023
		9	H'0012 to H'0013	H'0024 to H'0027
		10	H'0014 to H'0015	H'0028 to H'002B
		11	H'0016 to H'0017	H'002C to H'002F
External interrupt	IRQ_0	12	H'0018 to H'0019	H'0030 to H'0033
	IRQ_1	13	H'001A to H'001B	H'0034 to H'0037
Reserved for system use		14	H'001C to H'001D	H'0038 to H'003B
		15	H'001E to H'001F	H'003C to H'003F
External interrupt	IRQ_4	16	H'0020 to H'0021	H'0040 to H'0043
	IRQ_5	17	H'0022 to H'0023	H'0044 to H'0047
Reserved for system use		18	H'0024 to H'0025	H'0048 to H'004B
		19	H'0026 to H'0027	H'004C to H'004F
Internal interrupts*2		20 to 60	H'0028 to H'0029 to H'0078 to H'0079	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.

4.2 Reset

4.2.1 Overview

A reset is the highest-priority exception. When the $\overline{\text{RES}}$ pin goes low, all processing halts and the H8/3039 Group enters the reset state. A reset initializes the internal state of the CPU and the registers of the on-chip supporting modules. Reset exception handling begins when the $\overline{\text{RES}}$ pin changes from low to high.

The chip can also be reset by overflow of the watchdog timer. For details see section 10, Watchdog Timer.

4.2.2 Reset Sequence

The H8/3039 Group enters the reset state when the $\overline{\text{RES}}$ pin goes low.

To ensure that the chip is reset, hold the $\overline{\text{RES}}$ pin low for at least 20 ms at power-up. To reset the chip during operation, hold the $\overline{\text{RES}}$ pin low for at least 10 system clock (ϕ) cycles. When using the flash memory version, hold at "Low" level for a least 1usec. See 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 H8/3039 Group chip starts reset exception handling as follows.

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

Figure 4.2 shows the reset sequence in modes 5 and 7.

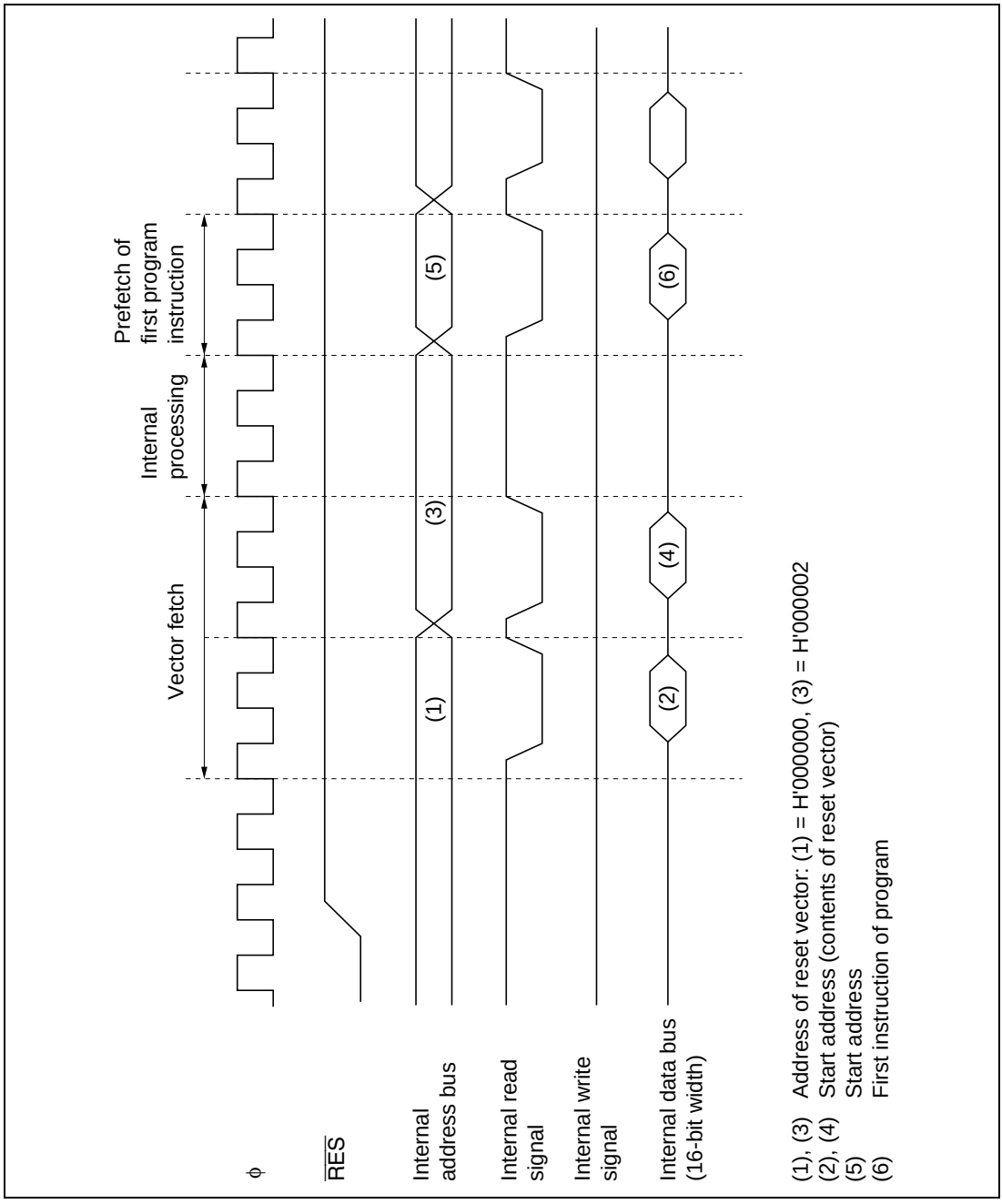


Figure 4.2 Reset Sequence (Modes 5 and 7)

4.2.3 Interrupts after Reset

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

4.3 Interrupts

Interrupt exception handling can be requested by five external sources (NMI, IRQ_0, IRQ_1, IRQ_4, IRQ_5) and 25 internal sources in the on-chip supporting modules. Figure 4.3 classifies the interrupt sources and indicates the number of interrupts of each type.

The on-chip supporting modules that can request interrupts are the watchdog timer (WDT), 16-bit integrated timer unit (ITU), serial communication interface (SCI), and A/D converter. Each interrupt source has a separate vector address.

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

For details on interrupts see section 5, Interrupt Controller.

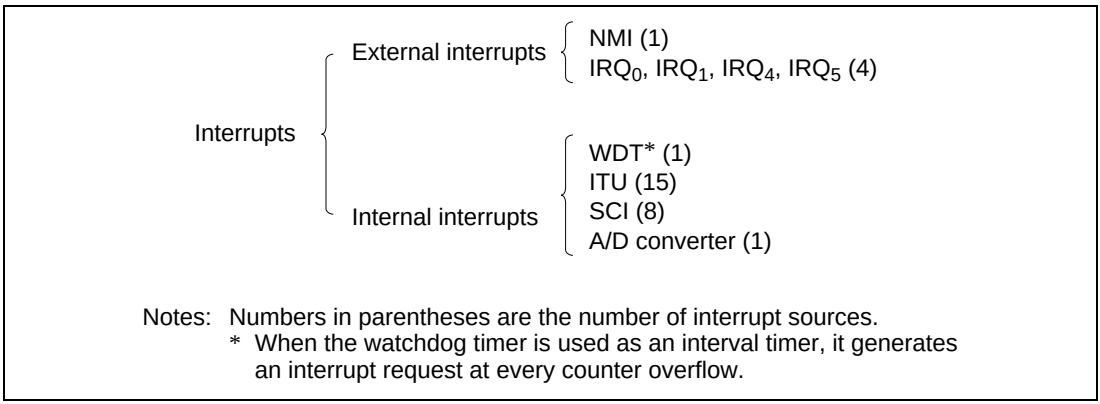


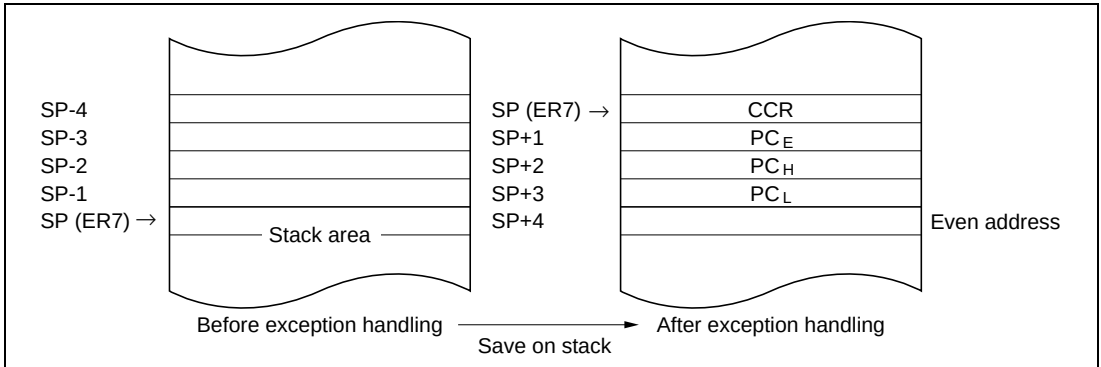
Figure 4.3 Interrupt Sources and Number of Interrupts

4.4 Trap Instruction

Trap instruction exception handling starts when a TRAPA instruction is executed. If the UE bit is set to 1 in the system control register (SYSCR), the exception handling sequence sets the I bit to 1 in CCR. If the UE bit is 0, the I and UI bits are both set to 1. The TRAPA instruction fetches a start address from a vector table entry corresponding to a vector number from 0 to 3, which is specified in the instruction code.

4.5 Stack Status after Exception Handling

Figure 4.4 shows the stack after completion of trap instruction exception handling and interrupt 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. Saving and restoring of registers must be conducted at even addresses in word-size or longword-size units.

Figure 4.4 Stack after Completion of Exception Handling (Advanced Mode)

4.6 Notes on Stack Usage

When accessing word data or longword data, the H8/3039 Group regards the lowest address bit as 0. The stack should always be accessed by word access or longword access, and the value of the stack pointer (SP, ER7) should always be kept even. Use the following instructions to save registers:

```
PUSH.W Rn  (or MOV.W Rn, @–SP)
PUSH.L ERn (or MOV.L ERn, @–SP)
```

Use the following instructions to restore registers:

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

Setting SP to an odd value may lead to a malfunction. Figure 4.5 shows an example of what happens when the SP value is odd.

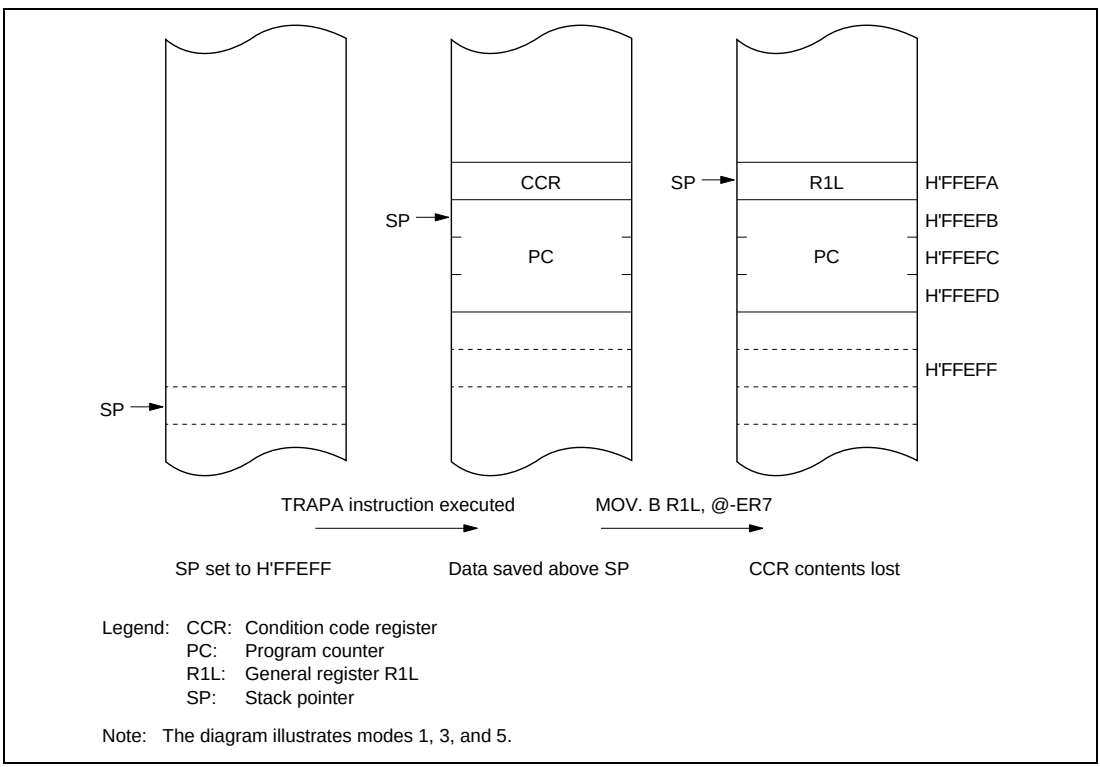


Figure 4.5 Operation when SP Value is Odd

Section 5 Interrupt Controller

5.1 Overview

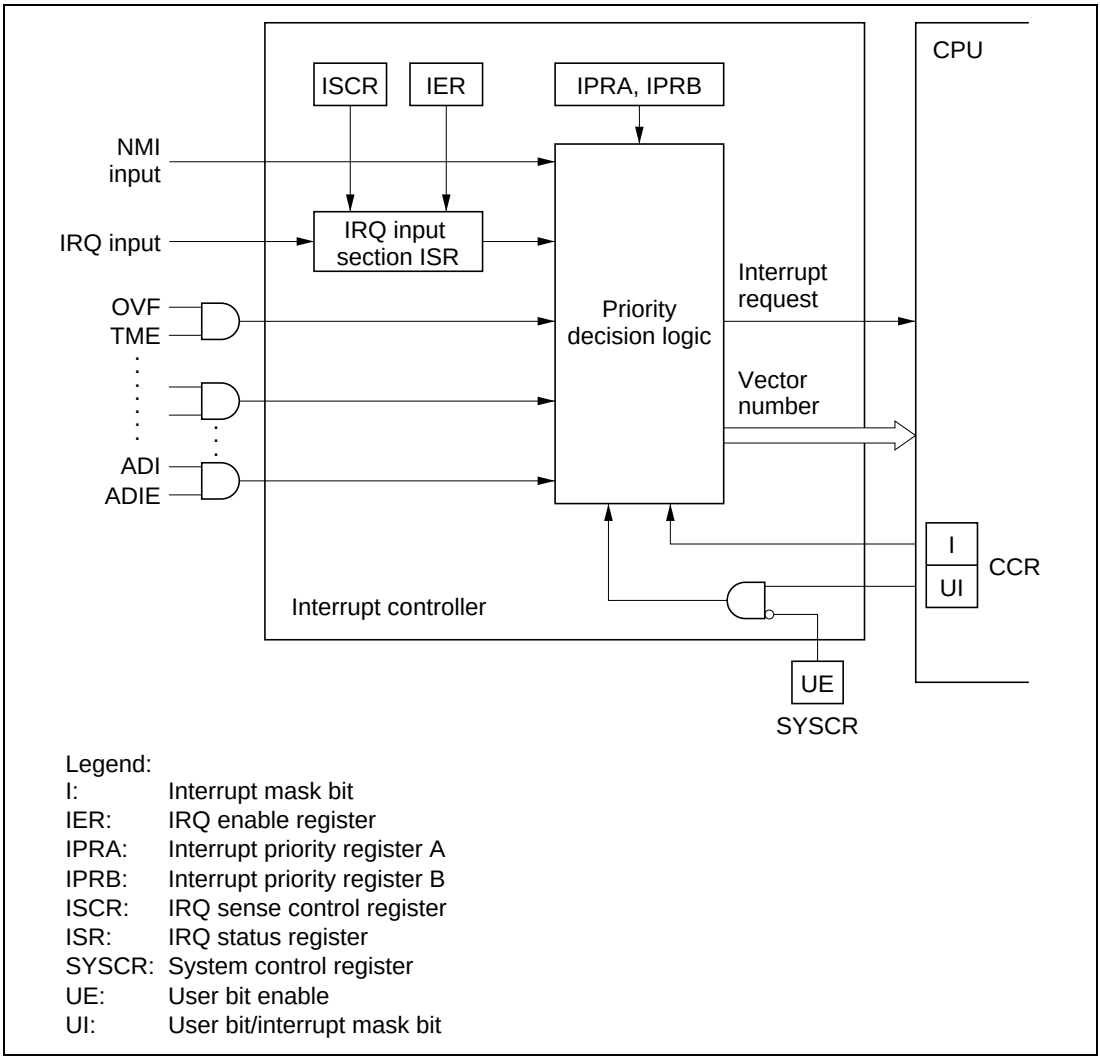
5.1.1 Features

The interrupt controller has the following features:

- Interrupt priority registers (IPRs) for setting interrupt priorities
  Interrupts other than NMI can be assigned to two priority levels on a module-by-module basis 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 have to identify the interrupt source.
- Five external interrupt pins
  NMI has the highest priority and is always accepted; either the rising or falling edge can be selected. For each of IRQ_0, IRQ_1, IRQ_4, and IRQ_5, sensing of the falling edge or level sensing can be selected independently.

5.1.2 Block Diagram

Figure 5.1 shows a block diagram of the interrupt controller.



Legend:

I:	Interrupt mask bit
IER:	IRQ enable register
IPRA:	Interrupt priority register A
IPRB:	Interrupt priority register B
ISCR:	IRQ sense control register
ISR:	IRQ status register
SYSCR:	System control register
UE:	User bit enable
UI:	User bit/interrupt mask bit

Figure 5.1 Interrupt Controller Block Diagram

5.1.3 Pin Configuration

Table 5.1 lists the interrupt pins.

Table 5.1 Interrupt Pins

Name	Abbreviation	I/O	Function
Nonmaskable interrupt	NMI	Input	Nonmaskable interrupt, rising edge or falling edge selectable
External interrupt request 5, 4, 1, and 0	$\overline{IRQ}_5$, $\overline{IRQ}_4$, and $\overline{IRQ}_1$, $\overline{IRQ}_0$	Input	Maskable interrupts, falling edge or level sensing selectable

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	Initial Value
H'FFF2	System control register	SYSCR	R/W	H'0B
H'FFF4	IRQ sense control register	ISCR	R/W	H'00
H'FFF5	IRQ enable register	IER	R/W	H'00
H'FFF6	IRQ status register	ISR	R/(W)*[2]	H'00
H'FFF8	Interrupt priority register A	IPRA	R/W	H'00
H'FFF9	Interrupt priority register B	IPRB	R/W	H'00

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

5.2 Register Descriptions

5.2.1 System Control Register (SYSCR)

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

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

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

Bit	7	6	5	4	3	2	1	0
	SSBY	STS2	STS1	STS0	UE	NMIEG	—	RAME
Initial value	0	0	0	0	1	0	1	1
Read/Write	R/W	R/W	R/W	R/W	R/W	R/W	—	R/W

RAM enable

Reserved bit

NMI edge select
Selects the NMI input edge

Standby timer select 2 to 0

Software standby

User bit enable
Selects whether to use the UI bit in CCR as a user bit or interrupt mask bit

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

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

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

Bit 2 NMIEG	Description	
0	Interrupt is requested at falling edge of NMI input	(Initial value)
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.

Interrupt Priority Register A (IPRA)

IPRA is an 8-bit readable/writable register in which interrupt priority levels can be set.

Bit	7	6	5	4	3	2	1	0
	IPRA7	IPRA6	—	IPRA4	IPRA3	IPRA2	IPRA1	IPRA0
Initial value	0	0	0	0	0	0	0	0
Read/Write	R/W	R/W	R/W	R/W	R/W	R/W	R/W	R/W

Priority level A0
Selects the priority level of ITU channel 2 interrupt requests

Priority level A1
Selects the priority level of ITU channel 1 interrupt requests

Priority level A2
Selects the priority level of ITU channel 0 interrupt requests

Priority level A3
Selects the priority level of WDT interrupt requests

Priority level A4
Selects the priority level of IRQ_4 and IRQ_5 interrupt requests

Reserved bit

Priority level A6
Selects the priority level of IRQ_1 interrupt requests

Priority level A7
Selects the priority level of IRQ_0 interrupt requests

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

Bit 7—Priority Level A7 (IPRA7): Selects the priority level of IRQ0 interrupt requests.

Bit7 IPRA7	Description	
0	IRQ0 interrupt requests have priority level 0 (low priority)	(Initial value)
1	IRQ0 interrupt requests have priority level 1 (high priority)	

Bit 6—Priority Level A6 (IPRA6): Selects the priority level of IRQ1 interrupt requests.

Bit6 IPRA6	Description	
0	IRQ1 interrupt requests have priority level 0 (low priority)	(Initial value)
1	IRQ1 interrupt requests have priority level 1 (high priority)	

Bit 5—Reserved bit: This bit can be written and read, but it does not affect interrupt priority.

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

Bit4 IPRA4	Description	
0	IRQ_4, IRQ_5 interrupt requests have priority level 0 (low priority)	(Initial value)
1	IRQ_4, IRQ_5 interrupt requests have priority level 1 (high priority)	

Bit 3—Priority Level A3 (IPRA3): Selects the priority level of WTD interrupt requests.

Bit3 IPRA3	Description	
0	WDT interrupt requests have priority level 0 (low priority)	(Initial value)
1	WDT interrupt requests have priority level 1 (high priority)	

Bit 2—Priority Level A2 (IPRA2): Selects the priority level of ITU channel 0 interrupt requests.

Bit2 IPRA2	Description	
0	ITU channel 0 interrupt requests have priority level 0 (low priority)	(Initial value)
1	ITU channel 0 interrupt requests have priority level 1 (high priority)	

Bit 1—Priority Level A1 (IPRA1): Selects the priority level of ITU channel 1 interrupt requests.

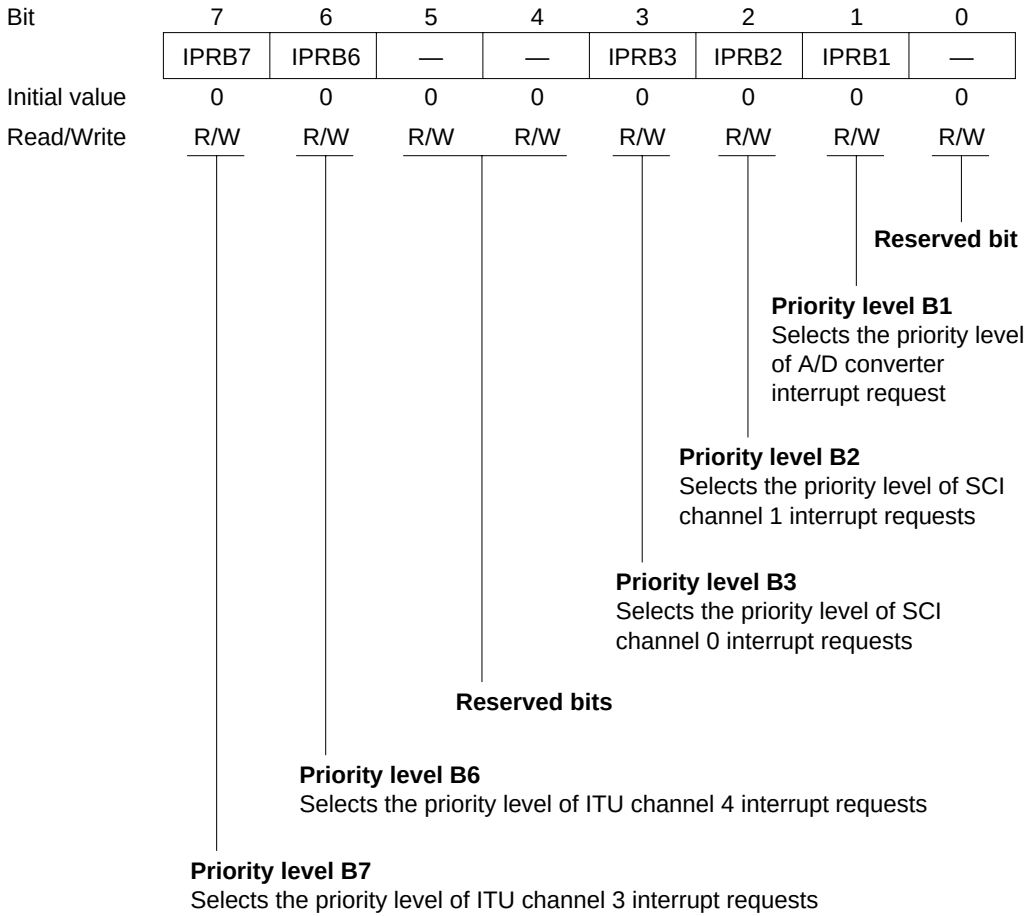
Bit1 IPRA1	Description	
0	ITU channel 1 interrupt requests have priority level 0 (low priority)	(Initial value)
1	ITU channel 1 interrupt requests have priority level 1 (high priority)	

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

Bit0 IPRA0	Description	
0	ITU channel 2 interrupt requests have priority level 0 (low priority)	(Initial value)
1	ITU channel 2 interrupt requests have priority level 1 (high priority)	

Interrupt Priority Register B (IPRB)

IPRB is an 8-bit readable/writable register in which interrupt priority levels can be set.



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

Bit 7—Priority Level B7 (IPRB7): Selects the priority level of ITU channel 3 interrupt requests.

Bit7 IPRB7	Description	
0	ITU channel 3 interrupt requests have priority level 0 (low priority)	(Initial value)
1	ITU channel 3 interrupt requests have priority level 1 (high priority)	

Bit 6—Priority Level B6 (IPRB6): Selects the priority level of ITU channel 4 interrupt requests.

Bit6 IPRB6	Description	
0	ITU channel 4 interrupt requests have priority level 0 (low priority)	(Initial value)
1	ITU channel 4 interrupt requests have priority level 1 (high priority)	

Bits 5 and 4—Reserved: These bits cannot be modified and are always read as 0.

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

Bit3 IPRB3	Description	
0	SCI channel 0 interrupt requests have priority level 0 (low priority)	(Initial value)
1	SCI channel 0 interrupt requests have priority level 1 (high priority)	

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

Bit2 IPRB2	Description	
0	SCI channel 1 interrupt requests have priority level 0 (low priority)	(Initial value)
1	SCI channel 1 interrupt requests have priority level 1 (high priority)	

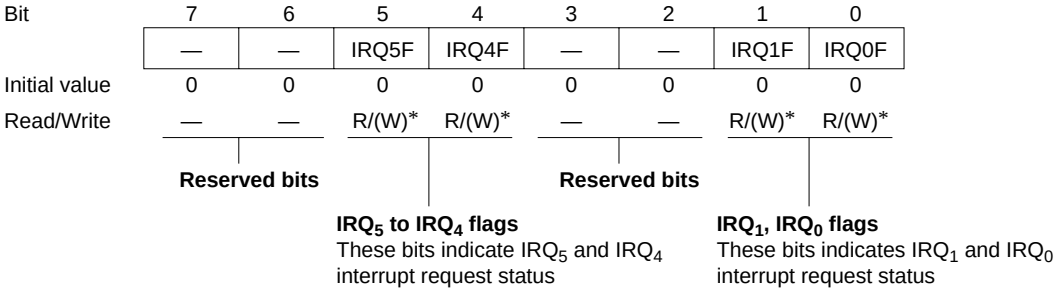
Bit 1—Priority Level B1 (IPRB1): Selects the priority level of A/D converter interrupt requests.

Bit1 IPRB1	Description	
0	A/D converter interrupt requests have priority level 0 (low priority)	(Initial value)
1	A/D converter interrupt requests have priority level 1 (high priority)	

Bit 0—Reserved: This bit cannot be modified and is always read as 0.

5.2.3 IRQ Status Register (ISR)

ISR is an 8-bit readable/writable register that indicates the status of IRQ_0, IRQ_1, IRQ_4, and IRQ_5 interrupt requests.



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, 6, 3 and 2—Reserved: These bits cannot be modified and are always read as 0.

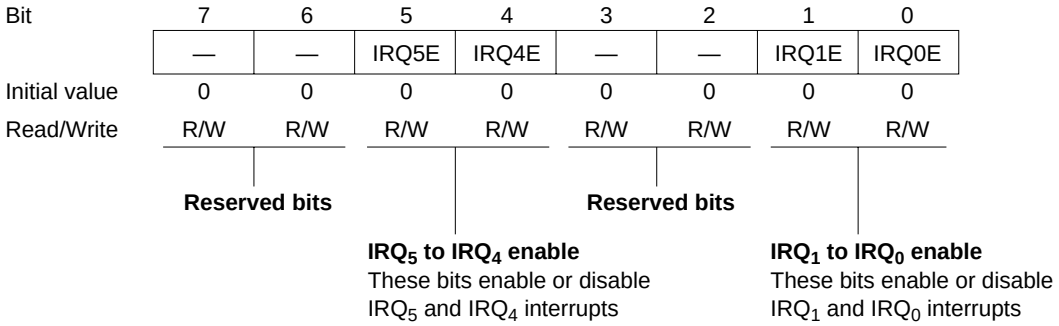
Bits 5, 4, 1 and 0—IRQ_5, IRQ_4, IRQ_1 and IRQ_0 Flags (IRQ5F, IRQ4F, IRQ1F, and IRQ0F): These bits indicate the status of IRQ_5, IRQ_4, IRQ_1, and IRQ_0 interrupt requests.

Bits 5, 4, 1, and 0 IRQ5F, IRQ4F, IRQ1F, and IRQ0F	Description
0	[Clearing conditions] (Initial value) • 0 is written in IRQnF after reading the IRQnF flag when IRQnF = 1. • IRQnSC = 0, $\overline{IRQ}$n input is high, and interrupt exception handling is carried out. • IRQnSC = 1 and IRQn interrupt exception handling is carried out.
1	[Setting conditions] • IRQnSC = 0 and $\overline{IRQ}$n input is low. • IRQnSC = 1 and $\overline{IRQ}$n input changes from high to low.

Note: n = 5, 4, 1 and 0

5.2.4 IRQ Enable Register (IER)

IER is an 8-bit readable/writable register that enables or disables IRQ_0, IRQ_1, IRQ_4, and IRQ_5 interrupt requests.



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

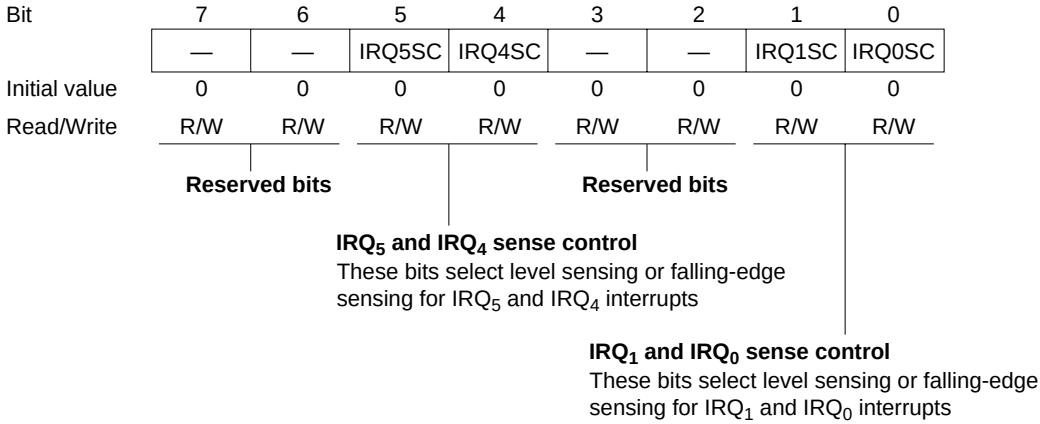
Bits 7, 6, 3, and 2—Reserved: These bits cannot be modified and are always read as 0.

Bits 5, 4, 1, and 0—IRQ_5, IRQ_4, IRQ_1, and IRQ_0 Enable (IRQ5E, IRQ4E, IRQ1E, IRQ0E): These bits enable or disable IRQ5, IRQ4, IRQ1, IRQ0 interrupts.

Bits 5, 4, 1, and 0 IRQ5E, IRQ4E, IRQ1E, and IRQ0E	Description	
0	IRQ_5, IRQ_4, IRQ_1, IRQ_0 interrupts are disabled	(Initial value)
1	IRQ_5, IRQ_4, IRQ_1, IRQ_0 interrupts are enabled	

5.2.5 IRQ Sense Control Register (ISCR)

ISCR is an 8-bit readable/writable register that selects level sensing or falling-edge sensing of the inputs at pins $\overline{IRQ_5}$, $\overline{IRQ_4}$, $\overline{IRQ_1}$, and $\overline{IRQ_0}$



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

Bits 7, 6, 3, and 2—Reserved: These bits are readable/writable and do not affect selection of level sensing or falling-edge sensing.

Bits 5, 4, 1, and 0—IRQ_5, IRQ_4, IRQ_1, and IRQ_0 Sense Control (IRQ5SC, IRQ4SC, IRQ1SC, IRQ0SC): These bits selects whether interrupts IRQ_5, IRQ_4, IRQ_1, IRQ_0 are requested by level sensing of pins $\overline{IRQ_5}$, $\overline{IRQ_4}$, $\overline{IRQ_1}$, $\overline{IRQ_0}$ or by falling-edge sensing.

Bits 5, 4, 1, and 0 IRQ5SC, IRQ4SC, IRQ1SC, IRQ0SC	Description
0	Interrupts are requested when $\overline{IRQ_5}$, $\overline{IRQ_4}$, $\overline{IRQ_1}$, $\overline{IRQ_0}$ inputs are low (Initial value)
1	Interrupts are requested by falling-edge input at $\overline{IRQ_5}$, $\overline{IRQ_4}$, $\overline{IRQ_1}$, $\overline{IRQ_0}$

5.3 Interrupt Sources

The interrupt sources include external interrupts (NMI, IRQ_5, IRQ_4, IRQ_1 and IRQ_0) and 25 internal interrupts.

5.3.1 External Interrupts

There are five external interrupts: NMI, and IRQ_5, IRQ_4, IRQ_1, and IRQ_0. Of these, NMI, IRQ0, IRQ1, can be used to exit software standby mode.

NMI: NMI is the highest-priority interrupt and is always accepted, regardless of the states of the I and UI bits in CCR. The NMIEG bit in SYSCR selects whether an interrupt is requested by the rising or falling edge of the input at the NMI pin. NMI interrupt exception handling has vector number 7.

IRQ_5, IRQ_4, IRQ_1, IRQ_0 Interrupts: These interrupts are requested by input signals at pins $\overline{IRQ_5}$, $\overline{IRQ_4}$, $\overline{IRQ_1}$, $\overline{IRQ_0}$. The IRQ_5, IRQ_4, IRQ_1, IRQ_0 interrupts have the following features.

- ISCR settings can select whether an interrupt is requested by the low level of the input at pins $\overline{IRQ_5}$, $\overline{IRQ_4}$, $\overline{IRQ_1}$, $\overline{IRQ_0}$, or by the falling edge.
- IER settings can enable or disable the IRQ_5, IRQ_4, IRQ_1, IRQ_0 interrupts.
  Interrupt priority levels can be assigned by four bits in IPRA (IPRA7, IPRA6, and IPRA4).
- The status of IRQ_5, IRQ_4, IRQ_1, IRQ_0 interrupt requests is indicated in ISR. The ISR flags can be cleared to 0 by software.

Figure 5.2 shows a block diagram of interrupts IRQ_5, IRQ_4, IRQ_1, IRQ_0.

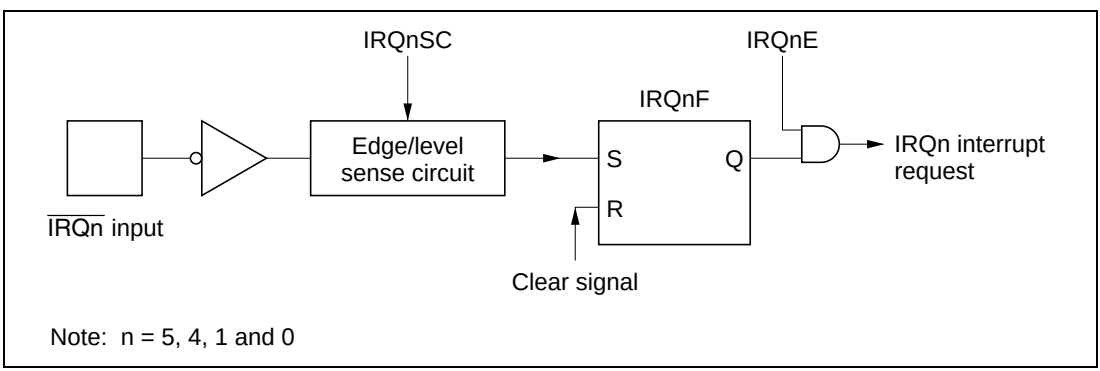


Figure 5.2 Block Diagram of Interrupts IRQ_5, IRQ_4, IRQ_1, and IRQ_0

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

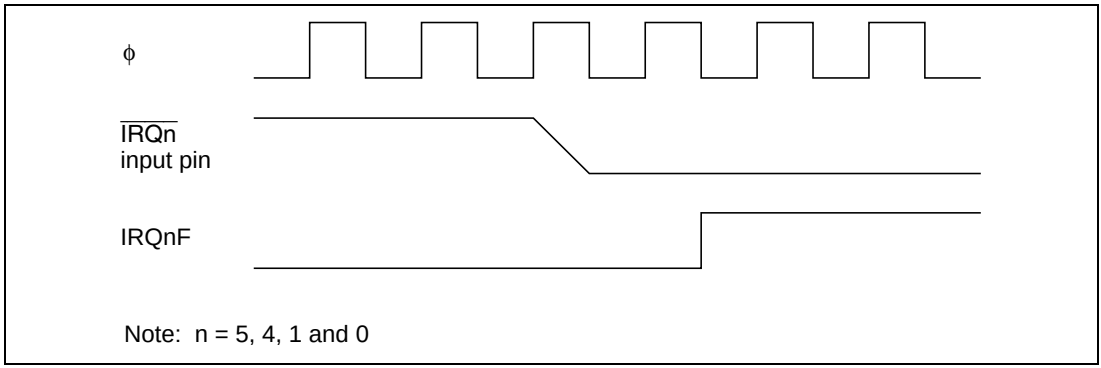


Figure 5.3 Timing of Setting of IRQnF

Interrupts IRQ_5, IRQ_4, IRQ_1, IRQ_0 have vector numbers 17, 16, 13, 12. These interrupts are detected regardless of whether the corresponding pin is set for input or output. When using a pin for external interrupt input, clear its DDR bit to 0 and do not use the pin for SCI input or output.

5.3.2 Internal Interrupts

Twenty-five internal interrupts are requested from the on-chip supporting modules.

- Each on-chip supporting module has status flags for indicating interrupt status, and enable bits for enabling or disabling interrupts.
- Interrupt priority levels can be assigned in IPRA and IPRB.

5.3.3 Interrupt Vector Table

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

Table 5.3 Interrupt Sources, Vector Addresses, and Priority

Interrupt Source	Origin	Vector Number	Vector Address*		IPR	Priority
			Normal Mode	Advanced Mode		
NMI	External pins	7	H'000E to H'000F	H'001C to H'001F	—	High
IRQ_0		12	H'0018 to H'0019	H'0030 to H'0033	IPRA7	
IRQ_1		13	H'001A to H'001B	H'0034 to H0037	IPRA6	
Reserved	—	14	H'001C to H'001D	H'0038 to H'003B	—	
		15	H'001E to H'001F	H'003C to H'003F		
IRQ_4	External pins	16	H'0020 to H'0021	H'0040 to H'0043	IPRA4	
IRQ_5		17	H'0022 to H'0023	H'0044 to H'0047		
Reserved	—	18	H'0024 to H'0025	H'0048 to H'004B		
		19	H'0026 to H'0027	H'004C to H'004F		
WOVI (interval timer)	Watchdog timer	20	H'0028 to H'0029	H'0050 to H'0053	IPRA3	
Reserved	—	21	H'002A to H'002B	H'0054 to H'0057		
		22	H'002C to H'002D	H'0058 to H'005B		
		23	H'002E to H'002F	H'005C to H'005F		
IMIA0 (compare match/ input capture A0)	ITU channel 0	24	H'0030 to H'0031	H'0060 to H'0063	IPRA2	
IMIB0 (compare match/ input capture B0)		25	H'0032 to H'0033	H'0064 to H'0067		
OVI0 (overflow 0)		26	H'0034 to H'0035	H'0068 to H'006B		
Reserved	—	27	H'0036 to H'0037	H'006C to H'006F		
IMIA1 (compare match/ input capture A1)	ITU channel 1	28	H'0038 to H'0039	H'0070 to H'0073	IPRA1	
IMIB1 (compare match/ input capture B1)		29	H'003A to H'003B	H'0074 to H'0077		
OVI1 (overflow 1)		30	H'003C to H'003D	H'0078 to H'007B		
Reserved	—	31	H'003E to H'003F	H'007C to H'007F		
IMIA2 (compare match/ input capture A2)	ITU channel 2	32	H'0040 to H'0041	H'0080 to H'0083	IPRA0	
IMIB2 (compare match/ input capture B2)		33	H'0042 to H'0043	H'0084 to H'0087		
OVI2 (overflow 2)		34	H'0044 to H'0045	H'0088 to H'008B		
Reserved	—	35	H'0046 to H'0047	H'008C to H'008F		

Interrupt Source	Origin	Vector Number	Vector Address*		IPR	Priority
			Normal Mode	Advanced Mode		
IMIA3 (compare match/ input capture A3)	ITU channel 3	36	H'0048 to H'0049	H'0090 to H'0093	IPRB7	↑
IMIB3 (compare match/ input capture B3)		37	H'004A to H'004B	H'0094 to H'0097		
OVI3 (overflow 3)		38	H'004C to H'004D	H'0098 to H'009B		
Reserved	—	39	H'004E to H'004F	H'009C to H'009F		
IMIA4 (compare match/ input capture A4)	ITU channel 4	40	H'0050 to H'0051	H'00A0 to H'00A3	IPRB6	
IMIB4 (compare match/ input capture B4)		41	H'0052 to H'0053	H'00A4 to H'00A7		
OVI4 (overflow 4)		42	H'0054 to H'0055	H'00A8 to H'00AB		
Reserved	—	43	H'0056 to H'0057	H'00AC to H'00AF	—	
		44	H'0058 to H'0059	H'00B0 to H'00B3		
		45	H'005A to H'005B	H'00B4 to H'00B7		
		46	H'005C to H'005D	H'00B8 to H'00BB		
		47	H'005E to H'005F	H'00BC to H'00BF		
		48	H'0060 to H'0061	H'00C0 to H'00C3		
		49	H'0062 to H'0063	H'00C4 to H'00C7		
		50	H'0064 to H'0065	H'00C8 to H'00CB		
		51	H'0066 to H'0067	H'00CC to H'00CF		
ERI0 (receive error 0)	SCI channel 0	52	H'0068 to H'0069	H'00D0 to H'00D3	IPRB3	
RXI0 (receive data full 0)		53	H'006A to H'006B	H'00D4 to H'00D7		
TXI0 (transmit data empty 0)		54	H'006C to H'006D	H'00D8 to H'00DB		
TEI0 (transmit end 0)		55	H'006E to H'006F	H'00DC to H'00DF		
ERI1 (receive error 1)	SCI channel 1	56	H'0070 to H'0071	H'00E0 to H'00E3	IPRB2	
RXI1 (receive data full 1)		57	H'0072 to H'0073	H'00E4 to H'00E7		
TXI1 (transmit data empty 1)		58	H'0074 to H'0075	H'00E8 to H'00EB		
TEI1 (transmit end 1)		59	H'0076 to H'0077	H'00EC to H'00EF		
ADI (A/D end)	A/D	60	H'0078 to H'0079	H'00F0 to H'00F3	IPRB1	↓ Low

Note: * Lower 16 bits of the address.

5.4 Interrupt Operation

5.4.1 Interrupt Handling Process

The H8/3039 Group handles interrupts differently depending on the setting of the UE bit. When UE = 1, interrupts are controlled by the I bit. When UE = 0, interrupts are controlled by the I and UI bits. Table 5.4 indicates how interrupts are handled for all setting combinations of the UE, I, and UI bits.

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

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 have higher priority.
	1	—	No interrupts are accepted except NMI.
0	0	—	All interrupts are accepted. Interrupts with priority level 1 have higher priority.
	1	0	NMI and interrupts with priority level 1 are accepted.
		1	No interrupts are accepted except NMI.

UE = 1

Interrupts IRQ_0, IRQ_1, IRQ_4, and IRQ_5 and interrupts from the on-chip supporting modules can all be masked by the I bit in the CPU's CCR. Interrupts are masked when the I bit is set to 1, and unmasked when the I bit is cleared to 0. Interrupts with priority level 1 have higher priority. Figure 5.4 is a flowchart showing how interrupts are accepted when UE = 1.

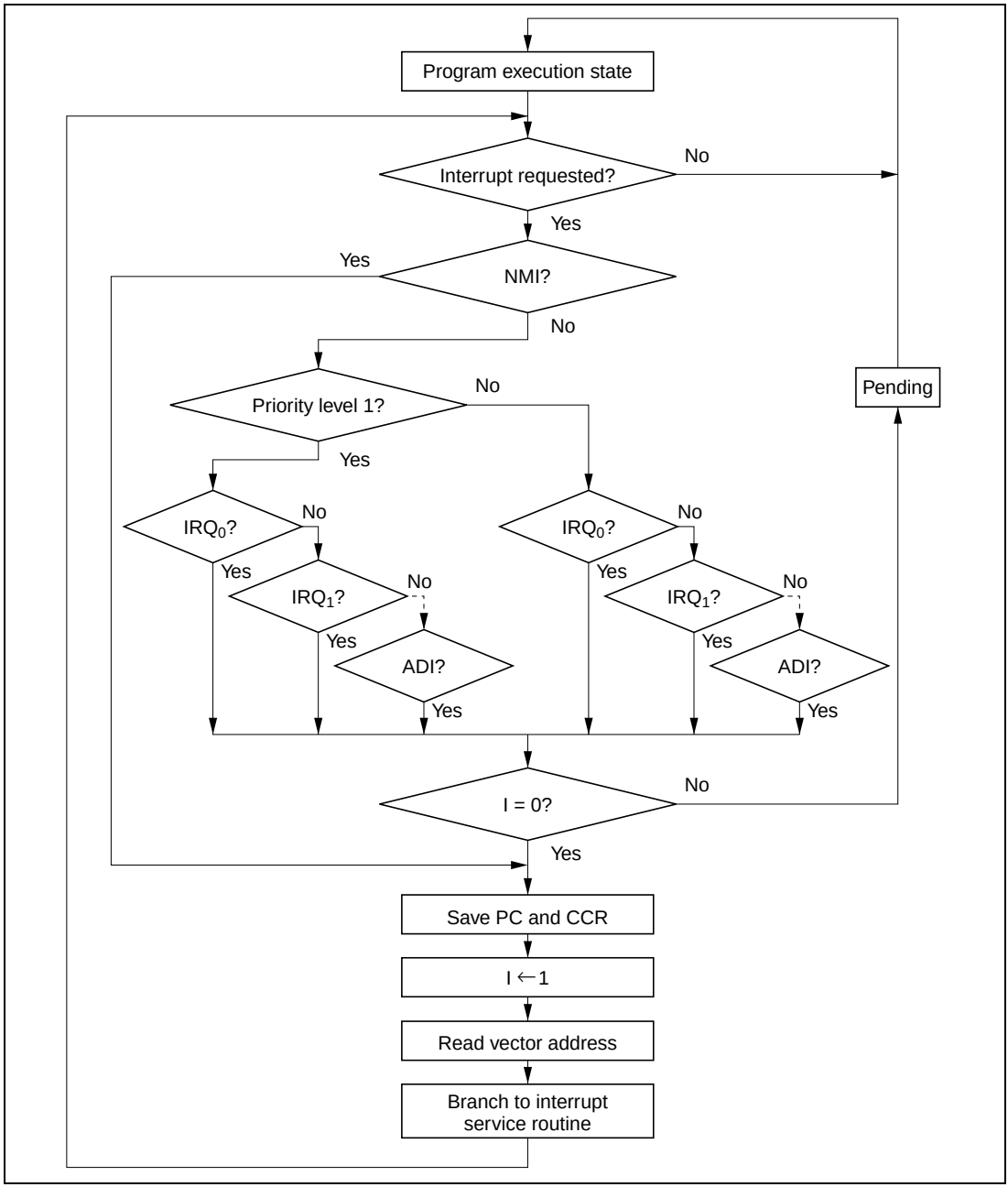


Figure 5.4 Process Up to Interrupt Acceptance when UE = 1

- If an interrupt condition occurs and the corresponding interrupt enable bit is set to 1, an interrupt request is sent to the interrupt controller.
- When the interrupt controller receives one or more interrupt requests, it selects the highest-priority request, following the IPR interrupt priority settings, and holds other requests pending. If two or more interrupts with the same IPR setting are requested simultaneously, the interrupt 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 interrupt request is accepted. If the I bit is set to 1, only NMI is accepted; other interrupt requests are held pending.
- When an interrupt request is accepted, interrupt exception handling starts after execution of the current instruction has been completed.
- In interrupt exception handling, PC and CCR are saved to the stack area. The PC value that is saved indicates the address of the first instruction that will be executed after the return from the 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 routine 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 IRQ_0, IRQ_1, IRQ_4, and IRQ_5 interrupts and interrupts from the on-chip supporting modules.

- Interrupt requests with priority level 0 are masked when the I bit is set to 1, and are unmasked when the I bit is cleared to 0.
- Interrupt requests with priority level 1 are masked when the I and UI bits are both set to 1, and 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 is set to H'10, and IPRB is set to H'00 (giving IRQ_4 and IRQ_5 interrupt requests priority over other interrupts), interrupts are masked as follows:

  a. If I = 0, all interrupts are unmasked (priority order: NMI > IRQ_4 > IRQ_5 > IRQ0 …).
  b. If I = 1 and UI = 0, only NMI, IRQ_4, and IRQ_5 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.

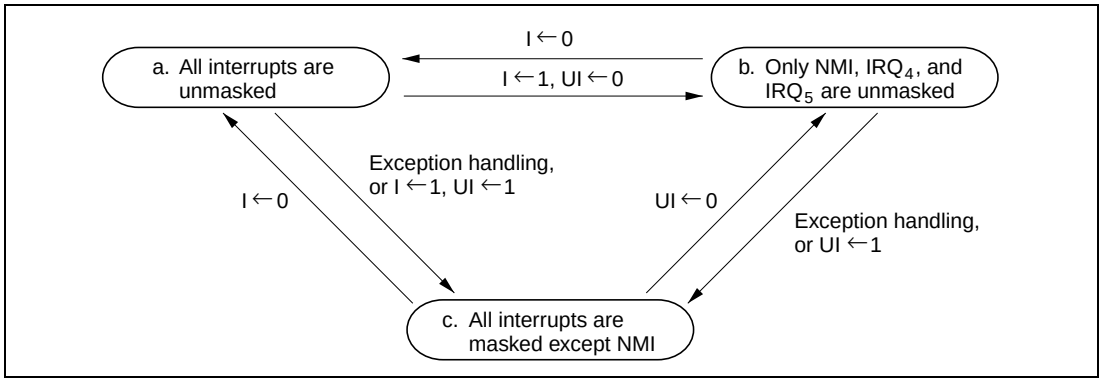


Figure 5.5 Interrupt Masking State Transitions (Example)

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, an interrupt request is sent to the interrupt controller.
- When the interrupt controller receives one or more interrupt requests, it selects the highest-priority request, following the IPR interrupt priority settings, and holds other requests pending. If two or more interrupts with the same IPR setting are requested simultaneously, the interrupt 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 interrupt request is accepted regardless of its IPR setting, and regardless of the UI bit. If the I bit is set to 1 and the UI bit is cleared to 0, only NMI and interrupts with priority level 1 are accepted; interrupt requests with priority level 0 are held pending. If the I bit and UI bit are both set to 1, only NMI is accepted; all other interrupt requests are held pending.
- When an interrupt request is accepted, interrupt exception handling starts after execution of the current instruction has been completed.
- In interrupt exception handling, PC and CCR are saved to the stack area. The PC value that is saved indicates the address of the first instruction that will be executed after the return from the 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 routine starts executing from the address indicated by the contents of the vector address.

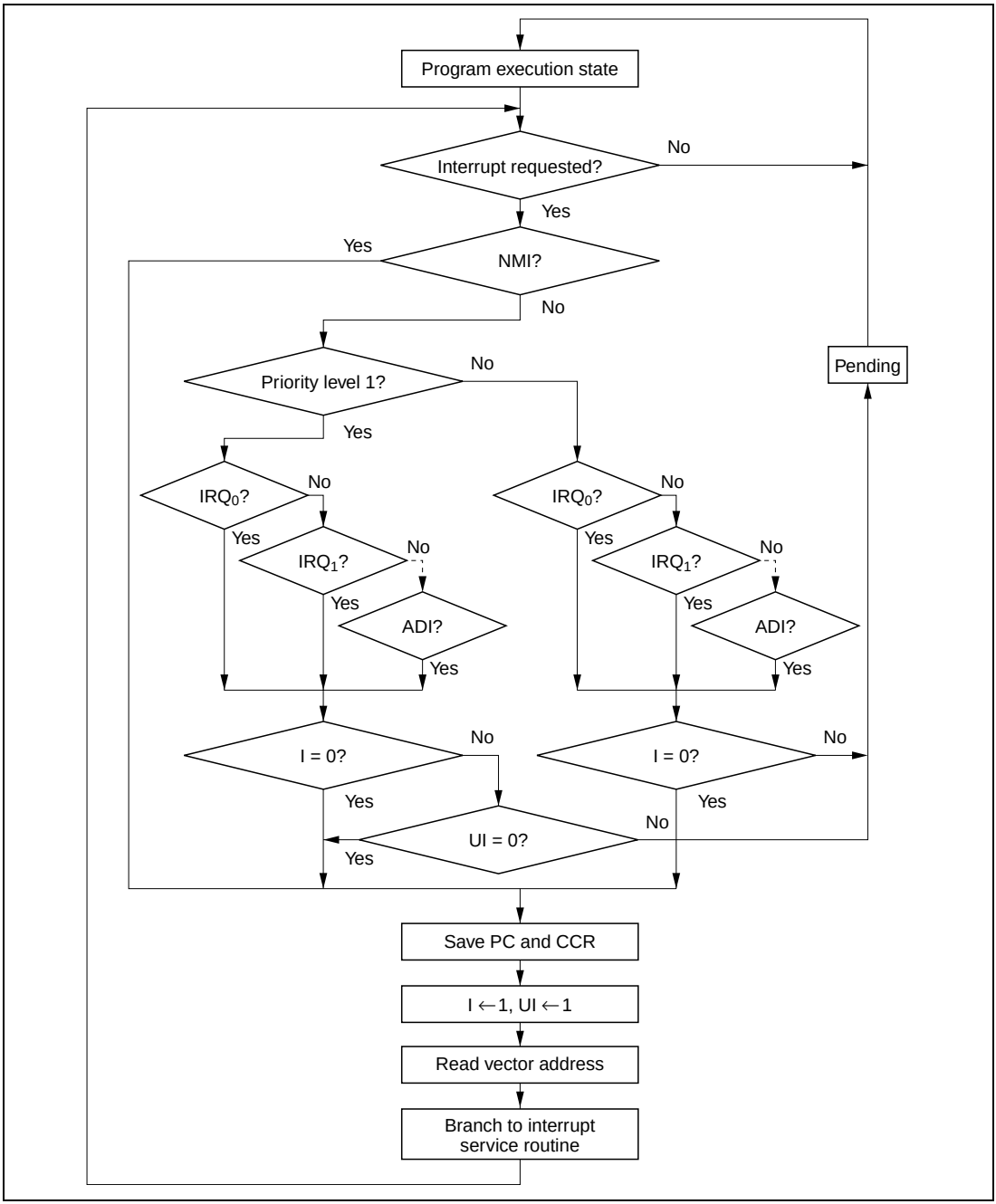


Figure 5.6 Process Up to Interrupt Acceptance when UE = 0

5.4.2 Interrupt Sequence

Figure 5.7 shows the interrupt sequence in mode 5 when the program code and stack are in an on-chip memory area.

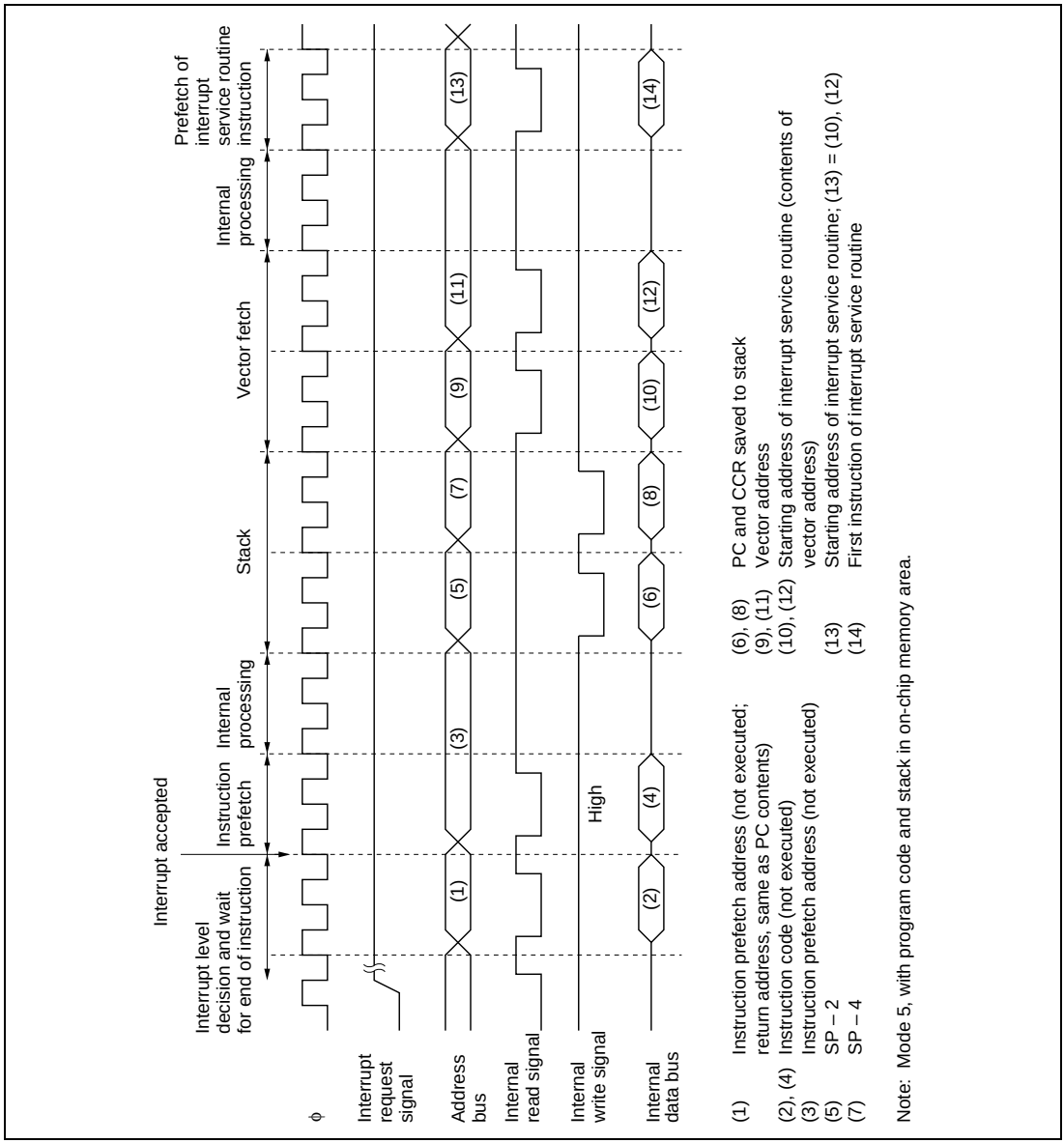


Figure 5.7 Interrupt Sequence (Mode 5, Stack in On-Chip Memory)

5.4.3 Interrupt Response Time

Table 5.5 indicates the interrupt response time from the occurrence of an interrupt request until the first instruction of the interrupt service routine is executed.

Table 5.5 Interrupt Response Time

			External Memory	
			8-Bit Bus	
No.	Item	On-Chip Memory	2 States	3 States
1	Interrupt priority decision	2*[1]	2*[1]	2*[1]
2	Maximum number of states	1 to 23	1 to 27	1 to 31*[4]
	until end of current instruction			
3	Saving PC and CCR to stack	4	8	12*[4]
4	Vector fetch	4	8	12*[4]
5	Instruction prefetch*[2]	4	8	12*[4]
6	Internal processing*[3]	4	4	4
Total		19 to 41	31 to 57	43 to 73

Notes: 1. 1 state for internal interrupts.
2. Prefetch after the interrupt is accepted and prefetch of the first instruction in the interrupt service routine.
3. Internal processing after the interrupt is accepted and internal processing after prefetch.
4. The number of states increases if wait states are inserted in external memory access.

5.5 Usage Notes

5.5.1 Contention between Interrupt and Interrupt-Disabling Instruction

When an instruction clears an interrupt enable bit to 0 to disable the interrupt, the interrupt is not disabled until after execution of the instruction is completed. If an interrupt occurs while a BCLR, MOV, or other instruction is being executed to clear its interrupt enable bit to 0, at the instant when execution of the instruction ends the interrupt is still enabled, so its interrupt exception handling is carried out. If a higher-priority interrupt is also requested, however, interrupt exception handling for the higher-priority interrupt is carried out, and the lower-priority interrupt is ignored. 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 the ITU's TIER.

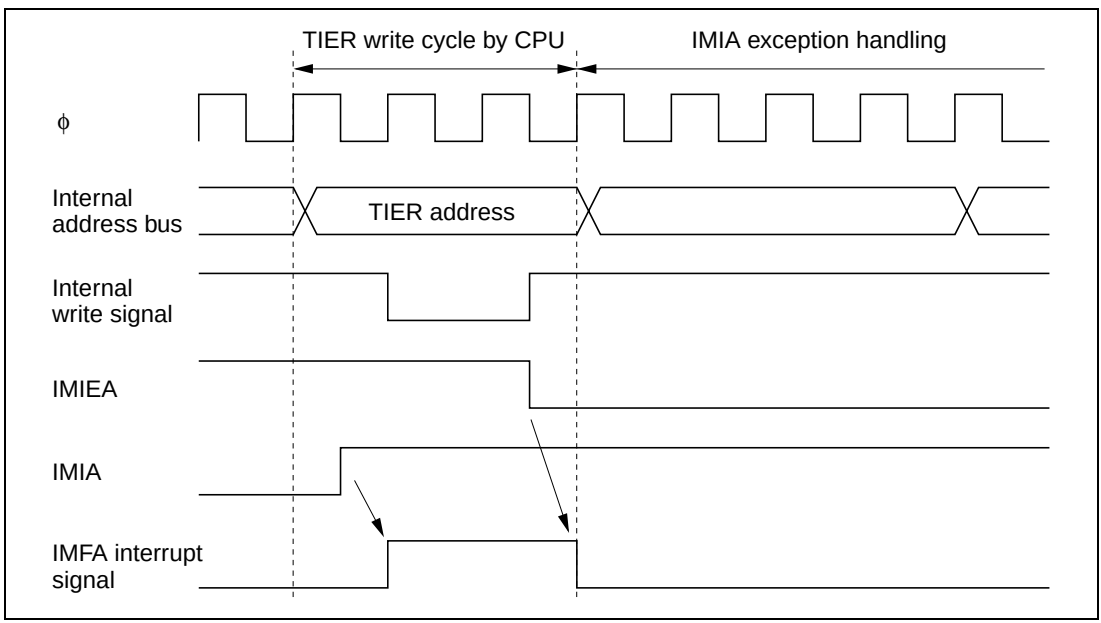


Figure 5.8 Contention between Interrupt and Interrupt-Disabling Instruction

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

5.5.2 Instructions that Inhibit Interrupts

The LDC, ANDC, ORC, and XORC instructions inhibit interrupts. When an interrupt occurs, after determining the interrupt priority, the interrupt controller requests a CPU interrupt. If the CPU is currently executing one of these interrupt-inhibiting instructions, however, when the instruction is completed the CPU always continues by executing the next instruction.

5.5.3 Interrupts during EEPMOV Instruction Execution

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

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

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

```
L1: EEPMO V.W
    MOV.W R4, R4
    BNE  L1
```

5.5.4 Usage Notes

The IRQnF flag specification calls for the flag to be cleared by writing 0 to it after it has been read while set to 1. However, it is possible for the IRQnF flag to be cleared by mistake simply by writing 0 to it, irrespective of whether it has been read while set to 1, with the result that interrupt exception handling is not executed. This will occur when the following conditions are met.

1. 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 flag, and clearing by hardware for the IRQbF flag.

(3) A bit-manipulation instruction is used on the IRQ status register for clearing the IRQaF flag, or else ISR is read as a byte unit, the IRQaF flag bit is cleared, and the values read in the other bits are written as a byte unit.

2. Generation Conditions

(1) A read of the ISR register is executed to clear the IRQaF flag while it is set to 1, then the IRQbF flag is cleared by the execution of interrupt exception handling.

(2) When the IRQaF flag is cleared, there is contention with IRQb generation (IRQaF flag setting). (IRQbF was 0 when ISR was read to clear the IRQaF flag, but IRQbF is set to 1 before ISR is written to.)

If the above setting conditions (1) to (3) and generation conditions (1) and (2) are all fulfilled, when the ISR write in generation condition (2) is performed the IRQbF flag will be cleared inadvertently, and interrupt exception handling will not be executed.

However, this inadvertent clearing of the IRQbF flag will not occur if 0 is written to this flag even once between generation conditions (1) and (2).

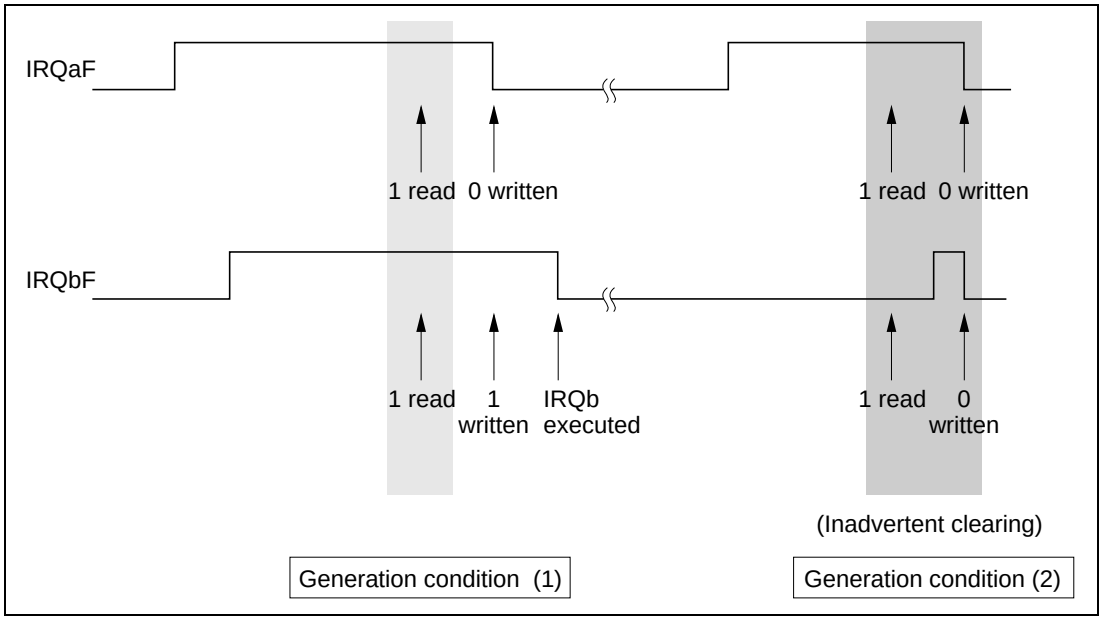


Figure 5.9 IRQnF Flag when Interrupt Exception Handling is not Executed

Either of the methods shown below should be used to prevent this problem.

Method 1: When clearing the IRQaF flag, read ISR as a byte unit instead of using a bit-manipulation instruction, and write a byte value that clears the IRQaF flag to 0 and sets the other bits to 1.

Example: When a = 0

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

Method 2: Perform dummy processing within the IRQb interrupt exception handling routine to clear the IRQbF flag.

Example: When b = 1

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

Section 6 Bus Controller

6.1 Overview

The H8/3039 Group has an on-chip bus controller that divides the external address space into eight areas and can assign different bus specifications to each. This enables different types of memory to be connected easily.

6.1.1 Features

Features of the bus controller are listed below.

- Independent settings for address areas 0 to 7
  - 128-kbyte areas in 1-Mbyte mode.
  - 2-Mbyte areas in 16-Mbyte mode.
  - 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 be selected.
  - Zero to three wait states can be inserted automatically.

6.1.2 Block Diagram

Figure 6.1 shows a block diagram of the bus controller.

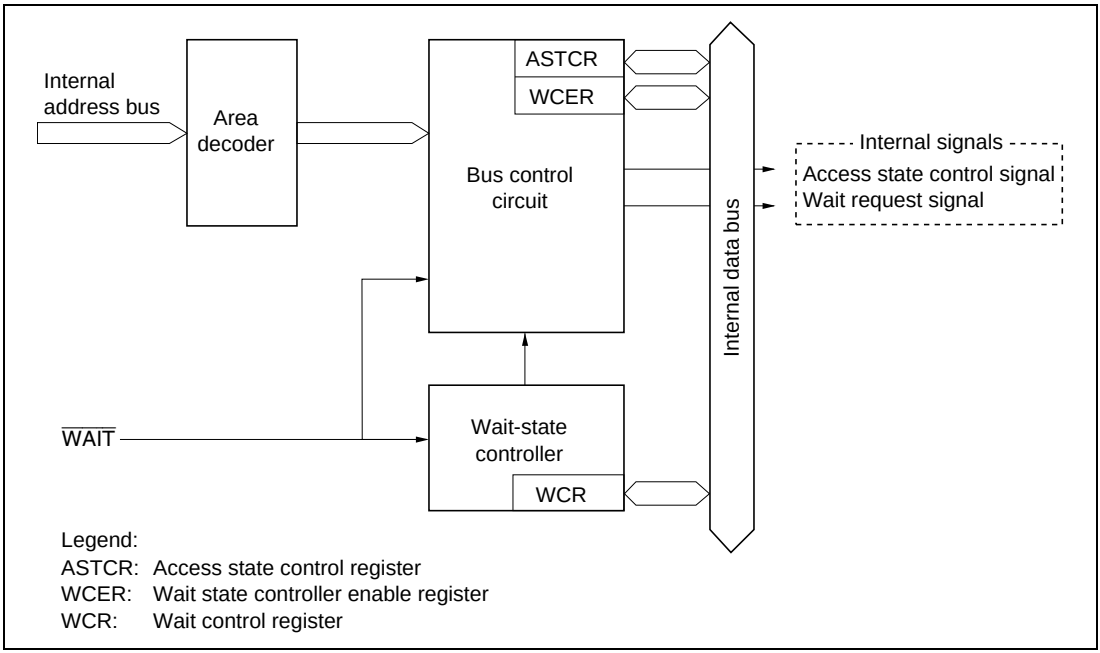


Figure 6.1 Block Diagram of Bus Controller

6.1.3 Input/Output Pins

Table 6.1 summarizes the bus controller's input/output pins.

Table 6.1 Bus Controller Pins

Name	Abbreviation	I/O	Function
Address strobe	$\overline{AS}$	Output	Strobe signal indicating valid address output on the address bus
Read	$\overline{RD}$	Output	Strobe signal indicating reading from the external address space
Write	$\overline{WR}$	Output	Strobe signal indicating writing to the external address space, with valid data on the data bus(D7 to D0)
Wait	$\overline{WAIT}$	Input	Wait request signal for access to external three-state-access areas

6.1.4 Register Configuration

Table 6.2 summarizes the bus controller's registers.

Table 6.2 Bus Controller Registers

Address*	Name	Abbreviation	R/W	Initial Value
H'FFED	Access state control register	ASTCR	R/W	H'FF
H'FFEE	Wait control register	WCR	R/W	H'F3
H'FFEF	Wait state controller enable register	WCER	R/W	H'FF
H'FFF3	Address control register	ADRCR	R/W	H'FE

Note: * Lower 16 bits of the address.

6.2 Register Descriptions

6.2.1 Access State Control Register (ASTCR)

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

Bit	7	6	5	4	3	2	1	0
	AST7	AST6	AST5	AST4	AST3	AST2	AST1	AST0
Initial value	1	1	1	1	1	1	1	1
Read/Write	R/W	R/W	R/W	R/W	R/W	R/W	R/W	R/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 initialized in software standby mode.

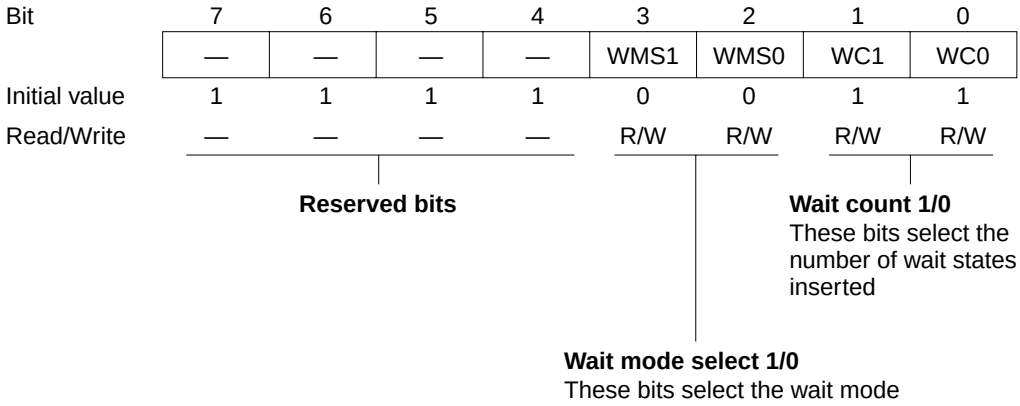
Bits 7 to 0—Area 7 to 0 Access State Control (AST7 to AST0): These bits select whether the corresponding area is accessed in two or three states.

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	(Initial value)

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

6.2.2 Wait Control Register (WCR)

WCR is an 8-bit readable/writable register that selects the wait mode for the wait-state controller (WSC) and specifies the number of wait states.



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

Bits 7 to 4—Reserved: These bits cannot be modified and are always read as 1.

Bits 3 and 2—Wait Mode Select 1 and 0 (WMS1/0): These bits select the wait mode.

Bit3 WMS1	Bit2 WMS0	Description	
0	0	Programmable wait mode	(Initial value)
	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 (WC1/0): These bits select the number of wait states inserted in access to external three-state-access areas.

Bit1 WC1	Bit0 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 (Initial value)

6.2.3 Wait State Controller Enable Register (WCER)

WCER is an 8-bit readable/writable register that enables or disables wait-state control of external three-state-access areas by the wait-state controller.

Bit	7	6	5	4	3	2	1	0
	WCE7	WCE6	WCE5	WCE4	WCE3	WCE2	WCE1	WCE0
Initial value	1	1	1	1	1	1	1	1
Read/Write	R/W	R/W	R/W	R/W	R/W	R/W	R/W	R/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 initialized in software standby mode.

Bits 7 to 0—Wait-State Controller Enable 7 to 0 (WCE7 to WCE0): These bits enable or 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 (Initial value)

WCER enables or disables wait-state control of external three-state-access areas. Therefore, in the single-chip modes (modes 6 and 7), the set value is meaningless.

6.2.4 Address Control Register (ADRCR)

ADRCR is an 8-bit readable/writable register that enables address output on bus lines A23 to A21.

	Bit	7	6	5	4	3	2	1	0
		$A_{23}E$	$A_{22}E$	$A_{21}E$	—	—	—	—	—
Modes 1 and 5 to 7	Initial value	1	1	1	1	1	1	1	0
	Read/Write	—	—	—	—	—	—	—	R/W
Mode 3	Initial value	1	1	1	1	1	1	1	0
	Read/Write	R/W	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

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

Bit 7—Address 23 Enable ($A_{23}E$): Enables PA4 to be used as the A_{23} address output pin. Writing 0 in this bit enables A_{23} address output from PA4. In modes other than 3 this bit cannot be modified and PA4 has its ordinary input/output functions

Bit 7 $A_{23}E$	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	(Initial value)

Bit 6—Address 22 Enable ($A_{22}E$): Enables PA_5 to be used as the A_{22} address output pin. Writing 0 in this bit enables A_{22} address output from PA_5. In modes other than 3 this bit cannot be modified and PA_5 has its ordinary input/output functions.

Bit 6 $A_{22}E$	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	(Initial value)

Bit 5—Address 21 Enable ($A_{21}E$): Enables PA_6 to be used as the A_{21} address output pin. Writing 0 in this bit enables A_{21} address output from PA_6. In modes other than 3 this bit cannot be modified and PA_6 has its ordinary input/output functions.

Bit 5 $A_{21}E$	Description	
0	PA_6 is the A_{21} address output pin	
1	PA_6 is the $PA_6/TP_6/TIOCA_2$ input/output pin	(Initial value)

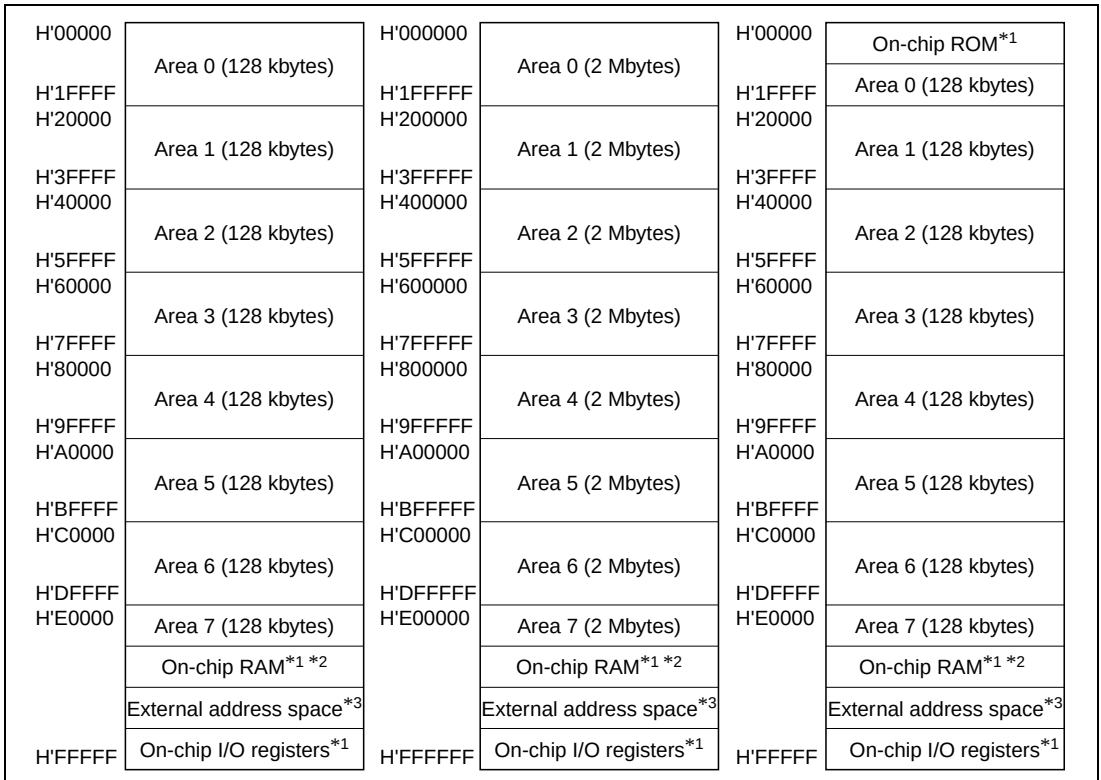
Bits 4 to 0—Reserved

6.3 Operation

6.3.1 Area Division

The external address space is divided into areas 0 to 7. Each area has a size of 128 kbytes in the

1-Mbyte mode and 2 Mbytes in the 16-Mbyte mode. Figure 6.2 shows a general view of the memory map.



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

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

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

Notes: There is no area division in modes 6 and 7.

1. The number of access states to on-chip ROM, on-chip RAM, and on-chip I/O registers is fixed.
2. This area follows area 7 specifications when the RAME bit in SYSCR is 0.
3. This area follows area 7 specifications.

Figure 6.2 Access Area Map (Mode 1, 3, and 5)

The bus specifications for each area can be selected in ASTCR, WCER, and WCR as shown in table 6.3.

Table 6.3 Bus Specifications

ASTCR	WCER	WCR		Bus Specifications		
ASTn	WCEn	WMS1	WMS0	Bus Width	Access States	Wait Mode
0	—	—	—	8	2	Disabled
1	0	—	—	8	3	Pin wait mode 0
	1	0	0	8	3	Programmable wait mode
			1	8	3	Disabled
		1	0	8	3	Pin wait mode 1
			1	8	3	Pin auto-wait mode

Note: n = 0 to 7

6.3.2 Bus Control Signal Timing

8-Bit, Three-State-Access Areas

Figure 6.3 shows the timing of bus control signals for an 8-bit, three-state-access area. Wait states can be inserted.

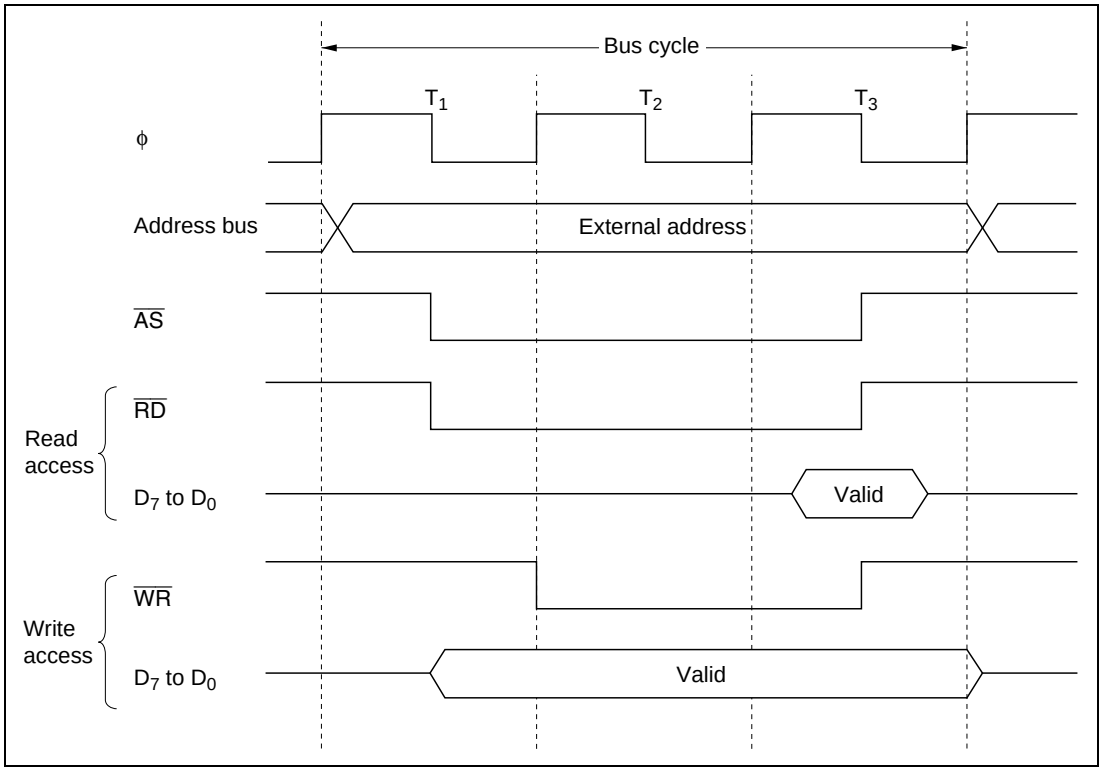


Figure 6.3 Bus Control Signal Timing for 8-Bit, Three-State-Access Area

8-Bit, Two-State-Access Areas

Figure 6.4 shows the timing of bus control signals for an 8-bit, two-state-access area. Wait states cannot be inserted.

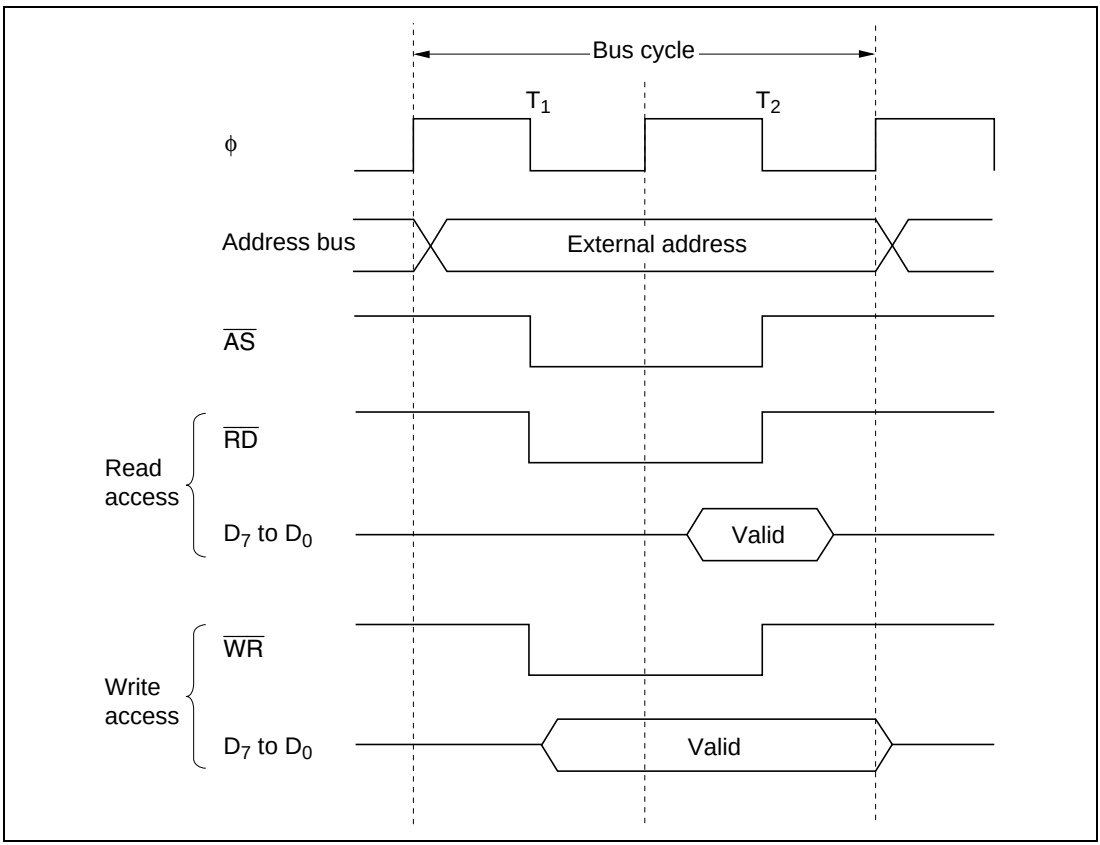


Figure 6.4 Bus Control Signal Timing for 8-Bit, Two-State-Access Area

6.3.3 Wait Modes

Four wait modes can be selected for each area as shown in table 6.4.

Table 6.4 Wait Mode Selection

ASTCR	WCER	WCR			
ASTn Bit	WCEn Bit	WMS1 Bit	WMS0 Bit	WSC Control	Wait Mode
0	—	—	—	Disabled	No wait states
1	0	—	—	Disabled	Pin wait mode 0
	1	0	0	Enabled	Programmable wait mode
			1	Enabled	No wait states
		1	0	Enabled	Pin wait mode 1
			1	Enabled	Pin auto-wait mode

Note: n = 0 to 7

The ASTn and WCEn bits can be set independently for each area. Bits WMS1 and WMS0 apply to all areas. All areas for which WSC control is enabled operate in the same wait mode.

Pin Wait Mode 0

The wait state controller is disabled. Wait states can only be inserted by $\overline{\text{WAIT}}$ pin control. During access to an external three-state-access area, if the $\overline{\text{WAIT}}$ pin is low at the fall of the system clock (ϕ) in the T_2 state, a wait state (T_W) is inserted. If the $\overline{\text{WAIT}}$ pin remains low, wait states continue to be inserted until the $\overline{\text{WAIT}}$ signal goes high. Figure 6.5 shows the timing.

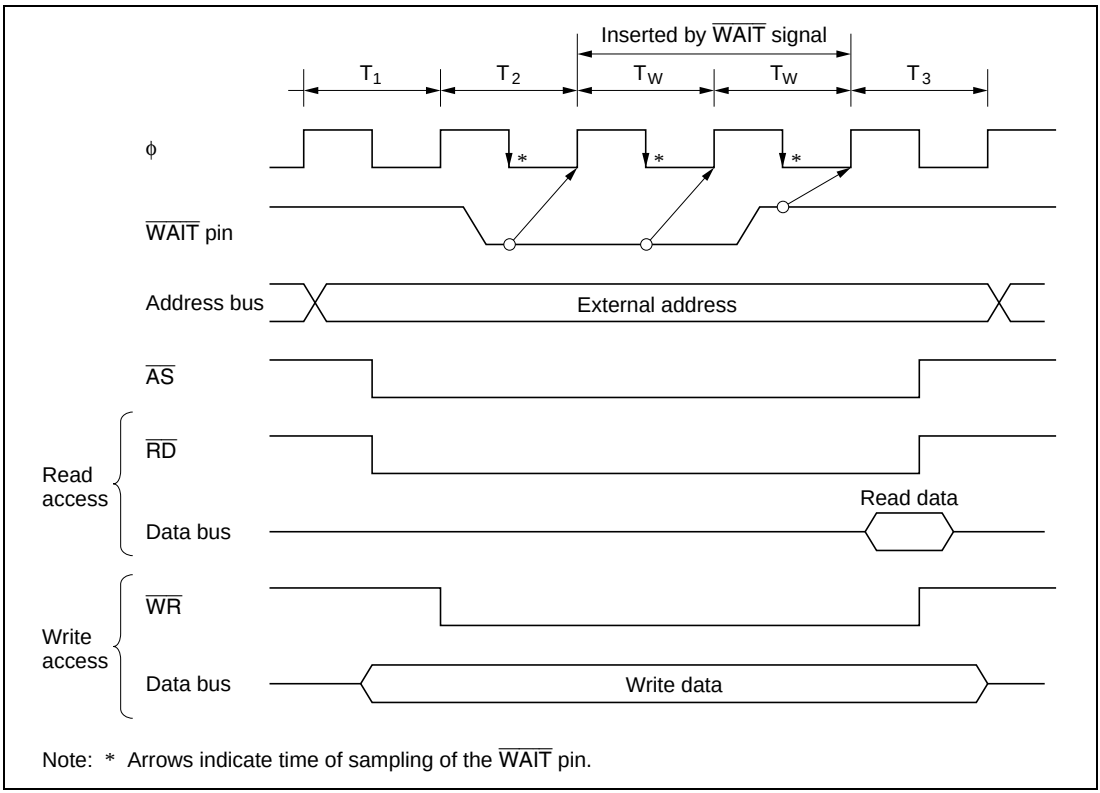


Figure 6.5 Pin Wait Mode 0

Pin Wait Mode 1

In all accesses to external three-state-access areas, the number of wait states (T_w) selected by bits WC1 and WC0 are inserted. If the $\overline{\text{WAIT}}$ pin is low at the fall of the system clock (ϕ) in the last of these wait states, an additional wait state is inserted. If the $\overline{\text{WAIT}}$ pin 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 different numbers 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.

Figure 6.6 shows the timing when the wait count is 1 (WC1 = 0, WC0 = 1) and one additional wait state is inserted by $\overline{\text{WAIT}}$ input.

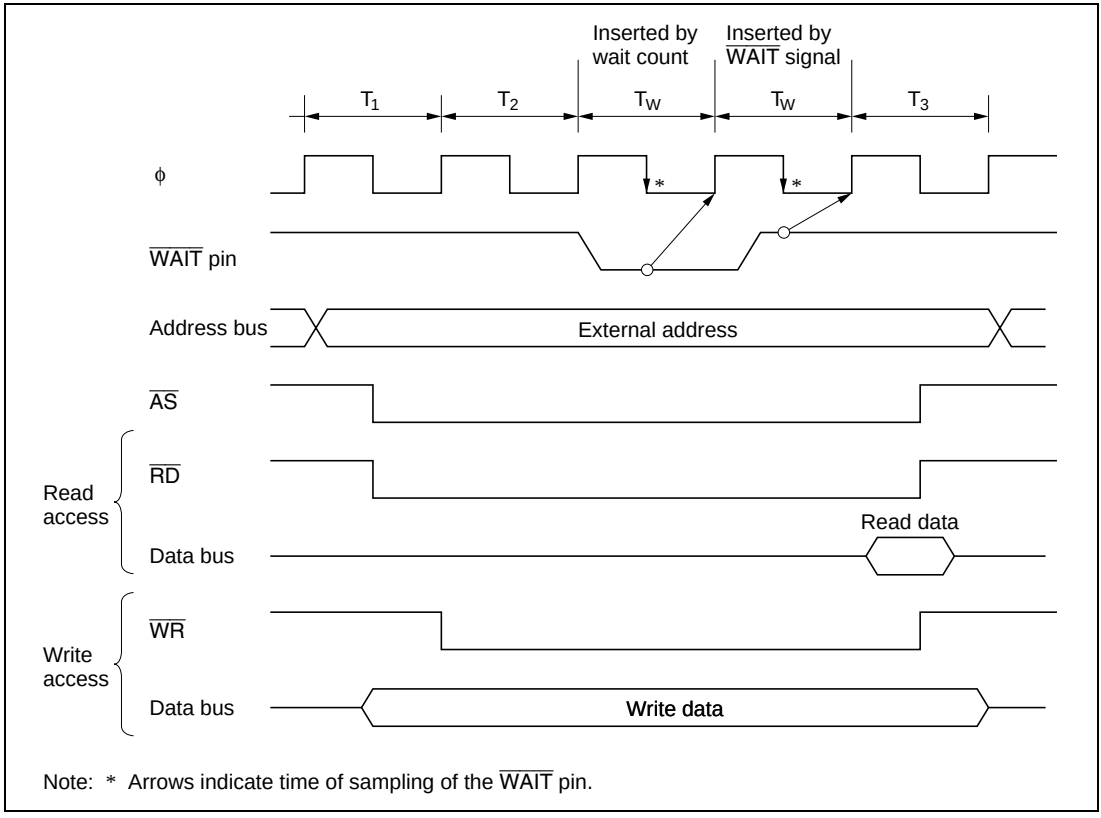


Figure 6.6 Pin Wait Mode 1

Pin Auto-Wait Mode

If the $\overline{\text{WAIT}}$ pin is low, the number of wait states (T_W) selected by bits WC1 and WC0 are inserted.

In pin auto-wait mode, if the $\overline{\text{WAIT}}$ pin is low at the fall of the system clock (ϕ) in the T_2 state, the number of wait states (T_W) selected by bits WC1 and WC0 are inserted. No additional wait states are inserted even if the $\overline{\text{WAIT}}$ pin remains low.

Figure 6.7 shows the timing when the wait count is 1.

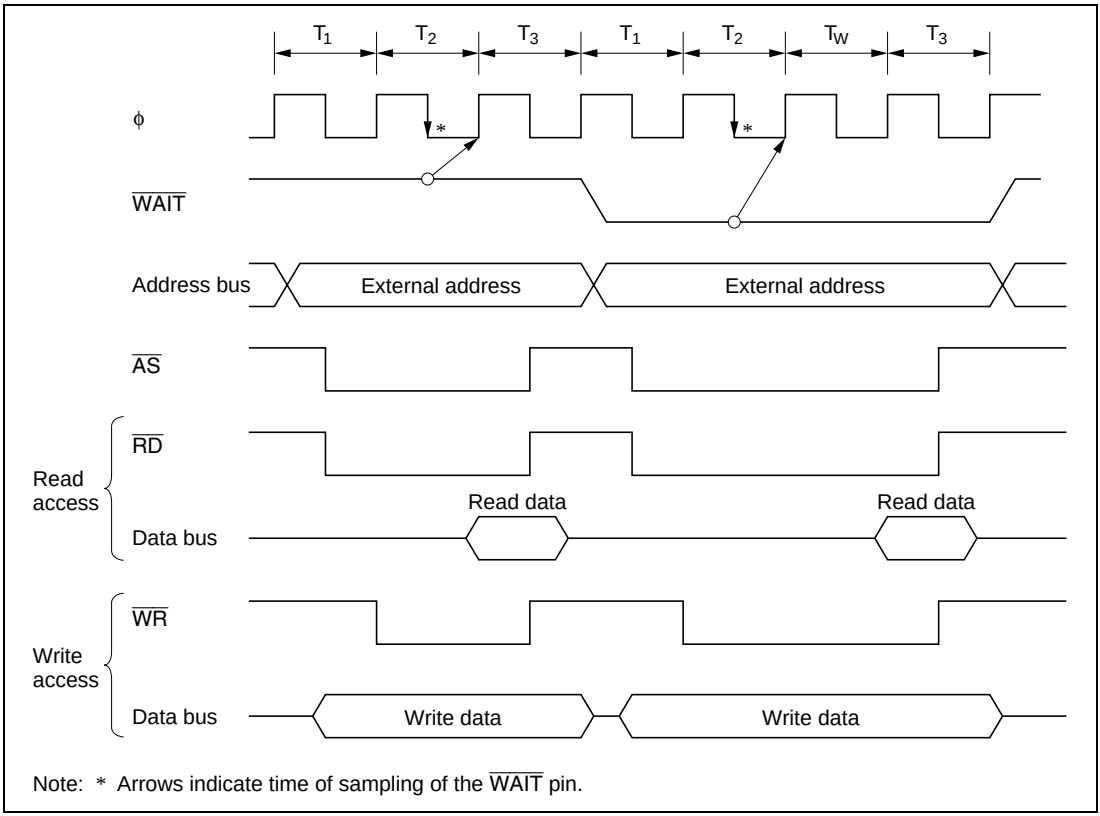


Figure 6.7 Pin Auto-Wait Mode

Programmable Wait Mode

The number of wait states (T_w) selected by bits WC1 and WC0 are inserted in all accesses to external three-state-access areas. Figure 6.8 shows the timing when the wait count is 1 (WC1 = 0, WC0 = 1).

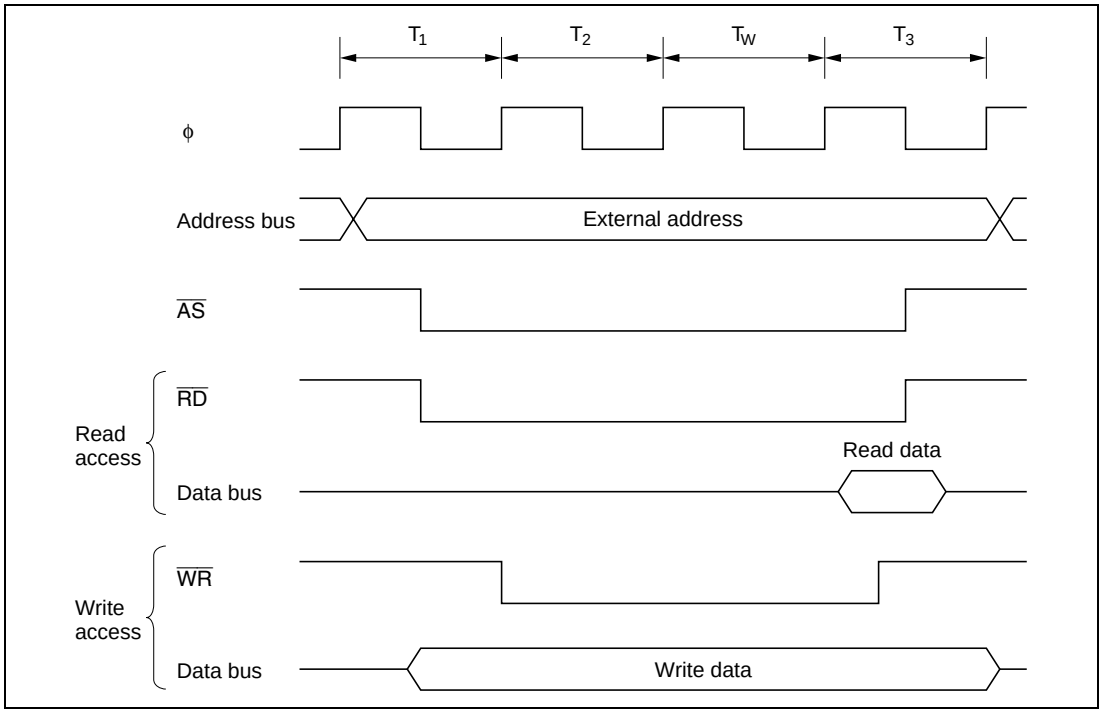


Figure 6.8 Programmable Wait Mode

Example of Wait State Control Settings

A reset initializes ASTCR and WCER to H'FF and WCR to H'F3, selecting programmable wait mode and three wait states for all areas. Software can select other wait modes for individual areas by modifying the ASTCR, WCER, and WCR settings. Figure 6.9 shows an example of wait mode settings.

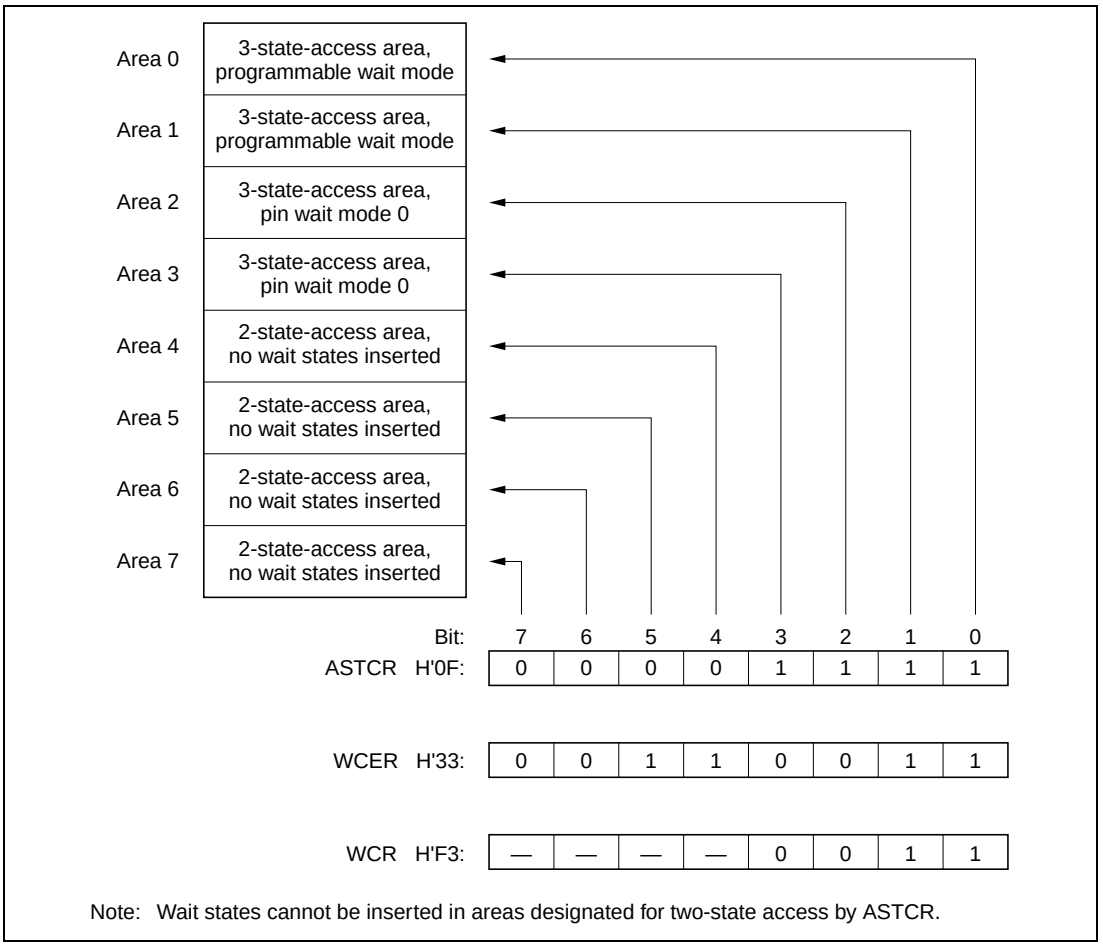


Figure 6.9 Wait Mode Settings (Example)

6.3.4 Interconnections with Memory (Example)

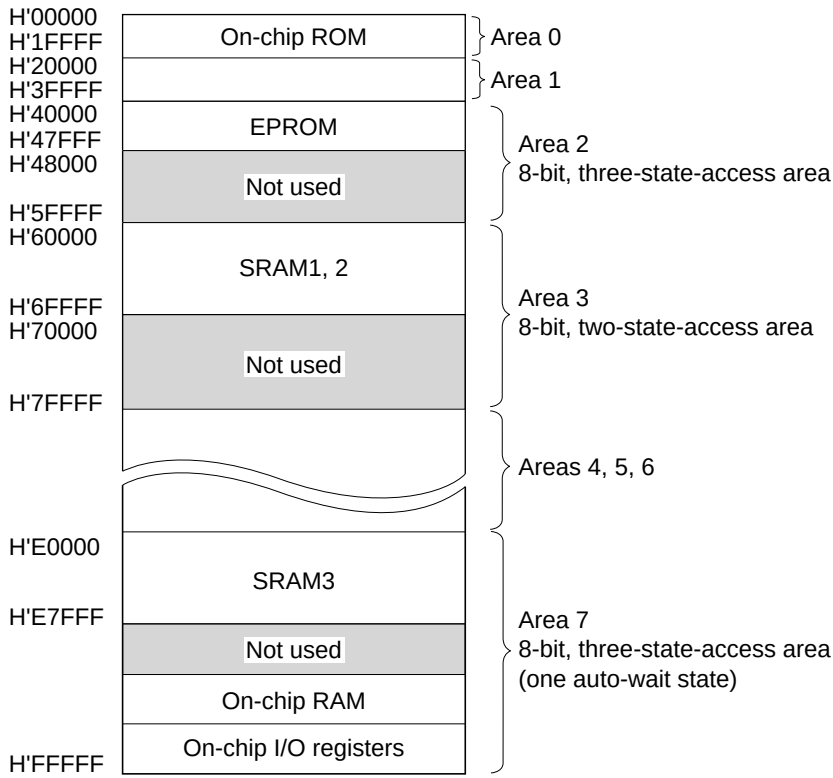
For each area, the bus controller can select two- or three-state access. In three-state-access areas, wait states can be inserted in a variety of modes, simplifying the connection of both high-speed and low-speed devices.

Figure 6.10 shows a memory map for this example.

A 32-kword × 8-bit EPROM is connected to area 2. This device is accessed in three states via an 8-bit bus.

Two 32-kword × 8-bit SRAM devices (SRAM1 and SRAM2) are connected to area 3. These devices are accessed in two states via an 8-bit bus.

One 32-kword × 8-bit SRAM (SRAM3) is connected to area 7. This device is accessed via an 8-bit bus, using three-state access with an additional wait state inserted in pin auto-wait mode.



Note: The bus width and the number of access states of the on-chip memories and I/O registers are fixed; they cannot be changed by register setting.

Figure 6.10 Memory Map (H8/3039 Mode 5)

6.4 Usage Notes

6.4.1 Register Write Timing

ASTCR and WCER Write Timing

Data written to ASTCR or WCER takes effect starting from the next bus cycle. Figure 6.11 shows the timing when an instruction fetched from area 2 changes area 2 from three-state access to two-state access.

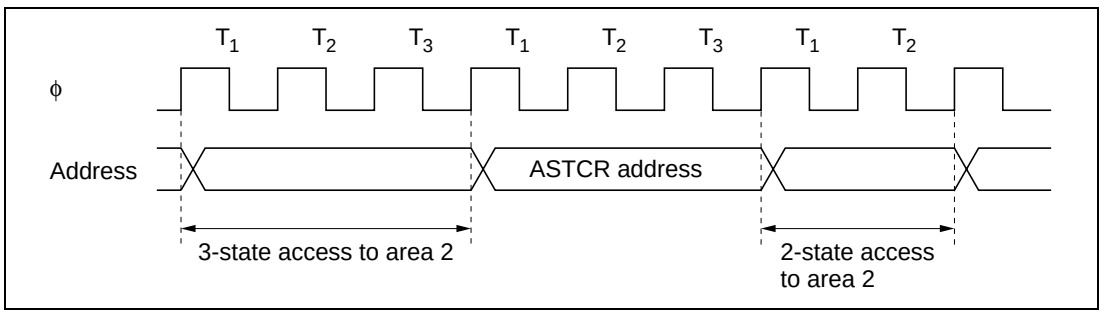


Figure 6.11 ASTCR Write Timing

6.4.2 Precautions on Setting ASTCR and ABWCR*

Use the H8/3039 Group on-chip program to set ASTCR and ABWCR as shown below, so that the on-chip ROM access cycle for H8/3039 Group can be emulated using the evaluation chip for support tools.

Modes 5 and 7

ASTCR0 = 0
ABWCR = H'FC

Note: * The ABWCR (bus width control register; lower 16-bit address: H'FFEC) is not built onto this LSI. For detailed features of the ABWCR, see the H8/3048 Group, H8/3048F-ZTAT™ Hardware Manual.

Section 7 I/O Ports

7.1 Overview

The H8/3039 Group has nine input/output ports (ports 1, 2, 3, 5, 6, 8, 9, A, and B) and one input port (port 7). Table 7.1 summarizes the port functions. The pins in each port are multiplexed as shown in table 7.1.

Each port has a data direction register (DDR) for selecting input or output, and a data register (DR) for storing output data. In addition to these registers, ports 2, and 5 have an input pull-up control register (PCR) for switching input pull-up transistors on and off.

Ports 1 to 3 and ports 5, 6, and 8 can drive one TTL load and a 90-pF capacitive load. Ports 9, A, and B can drive one TTL load and a 30-pF capacitive load. Ports 1 to 3 and ports 5, 6, 8, 9, A, and B can drive a Darlington pair. Ports 1, 2, 5, and B can drive LEDs (with 10-mA current sink). Pins $P8_1$, $P8_0$, PA_7 to PA_0, and PB_3 to PB_0 have Schmitt-trigger input circuits.

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

Table 7.1 Port Functions

Port	Description	Pins	Mode 1	Mode 3	Mode 5	Mode 6	Mode 7
Port 1	• 8-bit I/O port • Can drive LEDs	$P1_7$ to $P1_0$/ A_7 to A_0	Address output pins (A_7 to A_0)		Address output (A_7 to A_0) and generic input DDR = 0: generic input DDR = 1: address output	Generic input/ output	
Port 2	• 8-bit I/O port • Input pull-up • 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	Generic input/ output	
Port 3	• 8-bit I/O port	$P3_7$ to $P3_0$/ D_7 to D_0	Data input/output (D_7 to D_0)			Generic input/ output	
Port 5	• 4-bit I/O port • Input pull-up • Can drive LEDs	$P5_3$ to $P5_0$/ A_{19} to A_{16}	Address output (A_{19} to A_{16})		Address output (A_{19} to A_{16}) and 4-bit generic input DDR = 0: generic input DDR = 1: address output	Generic input/ output	
Port 6	• 4-bit I/O port	$P6_5$/$\overline{WR}$, $P6_4$/$\overline{RD}$, $P6_3$/$\overline{AS}$	Bus control signal output ($\overline{WR}$, $\overline{RD}$, $\overline{AS}$)			Generic input/ output	
		$P6_0$/$\overline{WAIT}$	Bus control signal input/output ($\overline{WAIT}$) and 1-bit generic input/output				
Port 7	• 8-bit Input port	$P7_7$ to $P7_0$/ AN_7 to AN_0	Analog input (AN_7 to AN_0) to A/D converter, and generic input				
Port 8	• 2-bit I/O port • $P8_1$ and $P8_0$ have Schmitt inputs	$P8_1$/ $\overline{IRQ_1}$	$\overline{IRQ_1}$ input and 1-bit generic input/output			$\overline{IRQ_1}$ and $\overline{IRQ_0}$ input and generic input/ output	
		$P8_0$/$\overline{IRQ_0}$	$\overline{IRQ_0}$ input and 1-bit generic input/output				

Port	Description	Pins	Mode 1	Mode 3	Mode 5	Mode 6	Mode 7
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 serial communication interfaces 1 and 0 (SCI0, 1), $\overline{IRQ_5}$ and $\overline{IRQ_4}$ input, and 6-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		
		$PA_6/TP_6/TIOCA_2/A_{21}$, $PA_5/TP_5/TIOCB_1/A_{22}$, $PA_4/TP_4/TIOCA_1/A_{23}$	TPC output (TP_6 to TP_4), ITU input and output ($TIOCA_2$, $TIOCB_1$, $TIOCA_1$), 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}), and generic input/output	TPC output (TP_6 to TP_4), ITU input and output ($TIOCA_2$, $TIOCB_1$, $TIOCA_1$), and generic input/output		
		$PA_3/TP_3/TIOCB_0/TCLKD$, $PA_2/TP_2/TIOCA_0/TCLKC$, $PA_1/TP_1/TCLKB$, $PA_0/TP_0/TCLKA$	TPC output (TP_3 to TP_0), ITU input and output (TCLKD, TCLKC, TCLKB, TCLKA, $TIOCB_0$, $TIOCA_0$), and generic input/output				

Port	Description	Pins	Mode 1	Mode 3	Mode 5	Mode 6	Mode 7
Port B	• 7-bit I/Oport • Can drive LEDs • PB_3 to PB_0 have Schmitt inputs	$PB_7/TP_{15}/\overline{ADTRG}$	TPC output (TP_{15}), trigger input ($\overline{ADTRG}$) to A/D converter, 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 ($TOCXB_4$, $TOCXA_4$, $TIOCB_4$, $TIOCA_4$, $TIOCB_3$, $TIOCA_3$), and generic input/output				

7.2 Port 1

7.2.1 Overview

Port 1 is an 8-bit input/output port with the pin configuration shown in figure 7.1. The pin functions differ between the expanded modes with on-chip ROM disabled, expanded modes with on-chip ROM enabled, and single-chip mode. In modes 1, 3 (expanded modes with on-chip ROM disabled), they are address bus output pins (A_7 to A_0).

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

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

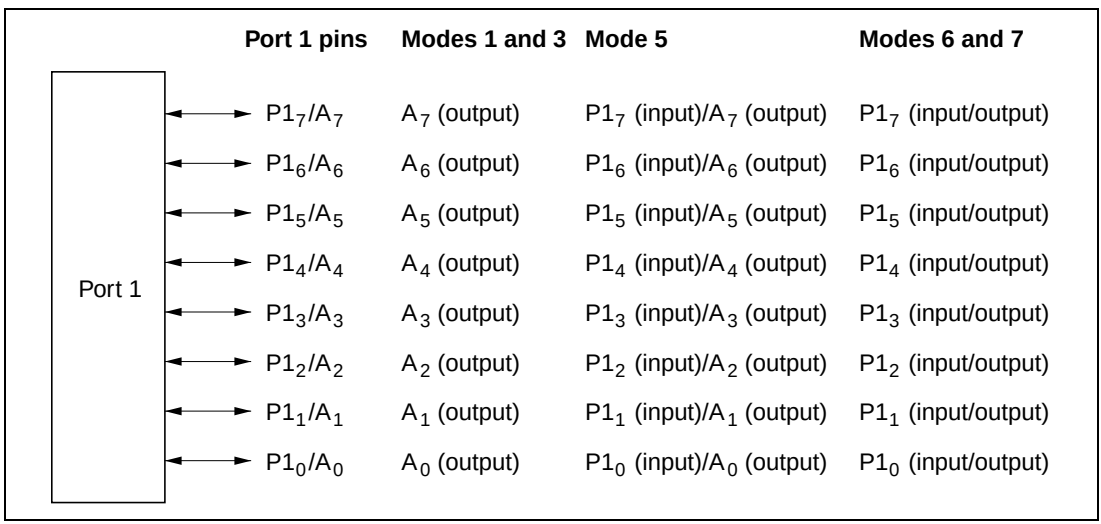


Figure 7.1 Port 1 Pin Configuration

7.2.2 Register Descriptions

Table 7.2 summarizes the registers of port 1.

Table 7.2 Port 1 Registers

Address*	Name	Abbreviation	R/W	Initial Value	
				Modes 1, 3	Modes 5 to 7
H'FFC0	Port 1 data direction register	P1DDR	W	H'FF	H'00
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 port 1.

Bit		7	6	5	4	3	2	1	0
		$P1_7DDR$	$P1_6DDR$	$P1_5DDR$	$P1_4DDR$	$P1_3DDR$	$P1_2DDR$	$P1_1DDR$	$P1_0DDR$
Modes 1, 3	Initial value	1	1	1	1	1	1	1	1
	Read/Write	—	—	—	—	—	—	—	—
Modes 5 to 7	Initial value	0	0	0	0	0	0	0	0
	Read/Write	W	W	W	W	W	W	W	W

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

P1DDR is initialized to H'00 by a reset and in hardware standby mode. In software standby mode it retains its previous setting. If a P1DDR bit is set to 1, the corresponding pin maintains its output state in software standby mode.

Port 1 Data Register (P1DR)

P1DR is an 8-bit readable/writable register that stores data for pins $P1_7$ to $P1_0$.

Bit	7	6	5	4	3	2	1	0
	$P1_7$	$P1_6$	$P1_5$	$P1_4$	$P1_3$	$P1_2$	$P1_1$	$P1_0$
Initial value	0	0	0	0	0	0	0	0
Read/Write	R/W	R/W	R/W	R/W	R/W	R/W	R/W	R/W

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

When a bit in P1DDR is set to 1, if port 1 is read the value of the corresponding P1DR bit is returned directly, regardless of the actual state of the pin. When a bit in P1DDR is cleared to 0, if port 1 is read the corresponding pin level is read.

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

7.2.3 Pin Functions in Each Mode

The pin functions of port 1 differ between mode 1, 3 (expanded mode with on-chip ROM disabled), mode 5 (expanded mode with on-chip ROM enabled), mode 6, and 7 (single-chip mode). The pin functions in each mode are described as follows.

Modes 1 and 3

Address output can be selected for each pin in port 1. Figure 7.2 shows the pin functions in modes 1 and 3.

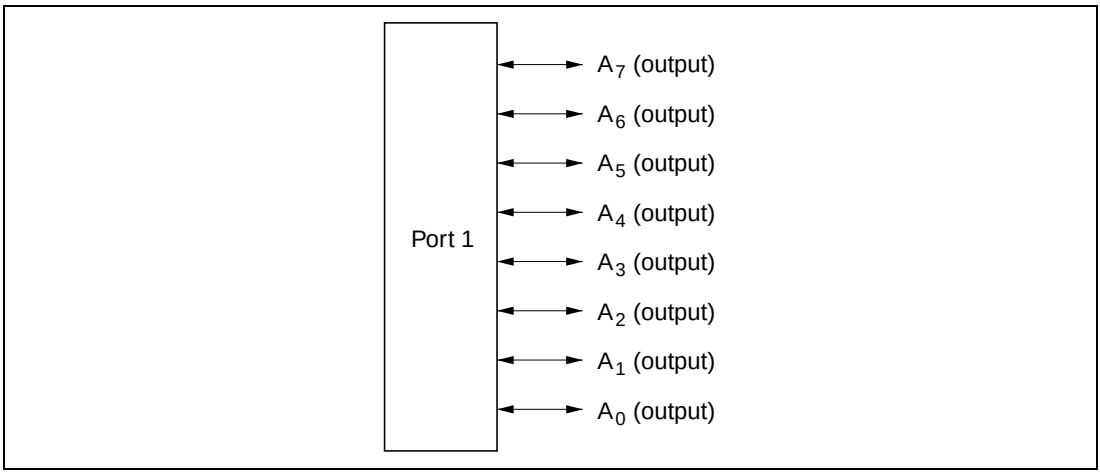


Figure 7.2 Pin Functions in Modes 1 and 3 (Port 1)

Mode 5

Address output or generic input can be selected for each pin in port 1. A pin becomes an address output pin if the corresponding P1DDR bit is set to 1, and a generic input pin if this bit is cleared to 0. Following a reset, all pins are input pins. To use a pin for address output, its P1DDR bit must be set to 1. Figure 7.3 shows the pin functions in mode 5.

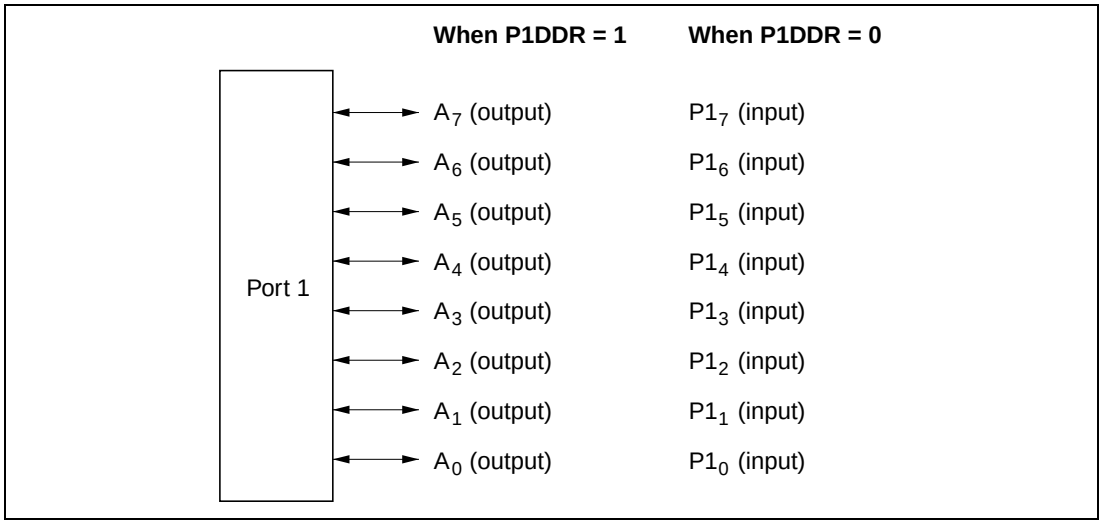


Figure 7.3 Pin Functions in Mode 5 (Port 1)

Modes 6 and 7 (Single-Chip Mode)

Input or output can be selected separately for each pin in port 1. A pin becomes an output pin if the corresponding P1DDR bit is set to 1, and an input pin if this bit is cleared to 0. Figure 7.4 shows the pin functions in modes 6 and 7.

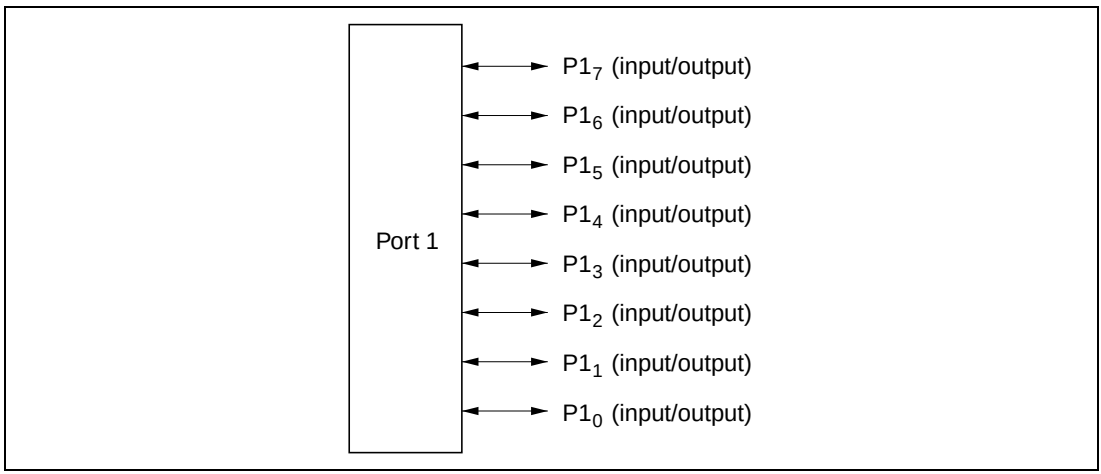


Figure 7.4 Pin Functions in Modes 6 and 7 (Port 1)

7.3 Port 2

7.3.1 Overview

Port 2 is an 8-bit input/output port with the pin configuration shown in figure 7.5. Pin functions differ according to operation mode.

In modes 1 and 3 (expanded mode with on-chip ROM disabled), port 2 consists of address bus output pins (A_{15} to A_8). In mode 5 (expanded mode with on-chip ROM enabled), settings in the port 2 data direction register (P2DDR) can designate pins for address bus output (A_{15} to A_8) or generic input. In modes 6 and 7 (single-chip mode), port 2 is a generic input/output port.

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

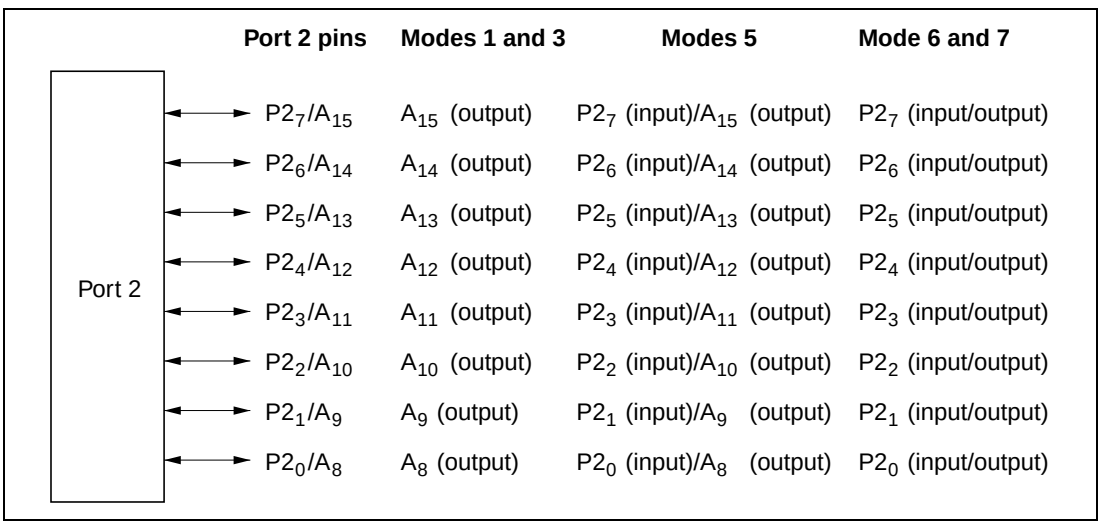


Figure 7.5 Port 2 Pin Configuration

7.3.2 Register Descriptions

Table 7.3 summarizes the registers of port 2.

Table 7.3 Port 2 Registers

Address*	Name	Abbreviation	R/W	Initial Value	
				Modes 1 and 3	Modes 5 to 7
H'FFC1	Port 2 data direction register	P2DDR	W	H'FF	H'00
H'FFC3	Port 2 data register	P2DR	R/W	H'00	H'00
H'FFD8	Port 2 input pull-up control register	P2PCR	R/W	H'00	H'00

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 port 2.

Bit		7	6	5	4	3	2	1	0
		$P2_7DDR$	$P2_6DDR$	$P2_5DDR$	$P2_4DDR$	$P2_3DDR$	$P2_2DDR$	$P2_1DDR$	$P2_0DDR$
Modes 1 and 3	Initial value	1	1	1	1	1	1	1	1
	Read/Write	—	—	—	—	—	—	—	—
Modes 5 to 7	Initial value	0	0	0	0	0	0	0	0
	Read/Write	W	W	W	W	W	W	W	W

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

P2DDR is initialized to H'00 by a reset and in hardware standby mode. In software standby mode it retains its previous setting. If a P2DDR bit is set to 1, the corresponding pin maintains its output state in software standby mode.

Port 2 Data Register (P2DR)

P2DR is an 8-bit readable/writable register that stores data for pins $P2_7$ to $P2_0$.

Bit	7	6	5	4	3	2	1	0
	$P2_7$	$P2_6$	$P2_5$	$P2_4$	$P2_3$	$P2_2$	$P2_1$	$P2_0$
Initial value	0	0	0	0	0	0	0	0
Read/Write	R/W	R/W	R/W	R/W	R/W	R/W	R/W	R/W

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

When a bit in P2DDR is set to 1, if port 2 is read the value of the corresponding P2DR bit is returned directly, regardless of the actual state of the pin. When a bit in P2DDR is cleared to 0, if port 2 is read the corresponding pin level is read.

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

Port 2 Input Pull-Up Control Register (P2PCR)

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

Bit	7	6	5	4	3	2	1	0
	$P2_7$PCR	$P2_6$PCR	$P2_5$PCR	$P2_4$PCR	$P2_3$PCR	$P2_2$PCR	$P2_1$PCR	$P2_0$PCR
Initial value	0	0	0	0	0	0	0	0
Read/Write	R/W	R/W	R/W	R/W	R/W	R/W	R/W	R/W

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

When a P2DDR bit is cleared to 0 (selecting generic input) in modes 7 to 5, if the corresponding bit from $P2_7$PCR to $P2_0$PCR is set to 1, the input pull-up transistor is turned on.

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

7.3.3 Pin Functions in Each Mode

The pin functions of port 2 differ between mode 1, 3 (expanded mode with on-chip ROM disabled), mode 5 (expanded mode with on-chip ROM enabled), mode 6, and 7 (single-chip mode). The pin functions in each mode are described followings.

Modes 1 and 3

Address output can be selected for each pin in port 2. Figure 7.6 shows the pin functions in modes 1 and 3.

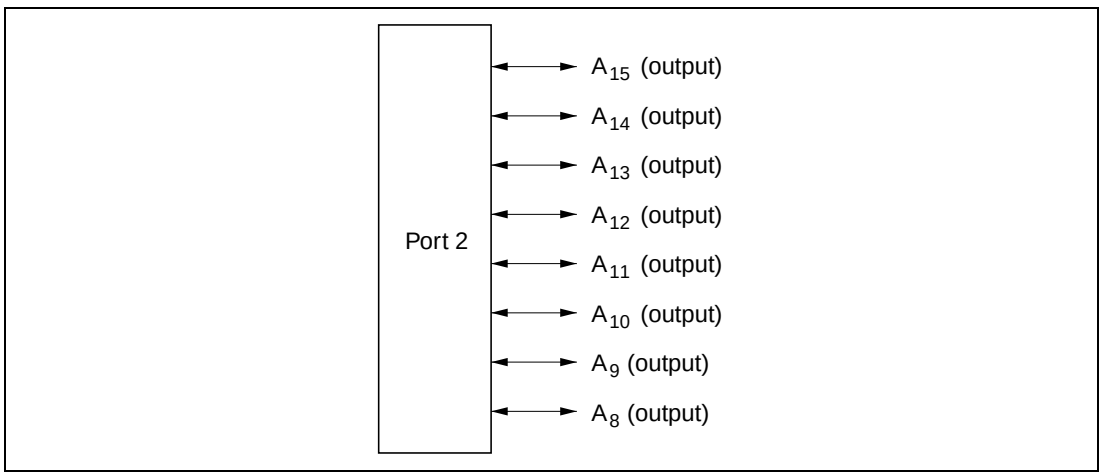


Figure 7.6 Pin Functions in Modes 1 and 3 (Port 2)

Mode 5

Address output or generic input can be selected for each pin in port 2. A pin becomes an address output pin if the corresponding P2DDR bit is set to 1, and a generic input pin if this bit is cleared to 0. Following a reset, all pins are input pins. To use a pin for address output, its P2DDR bit must be set to 1. Figure 7.7 shows the pin functions in modes 5.

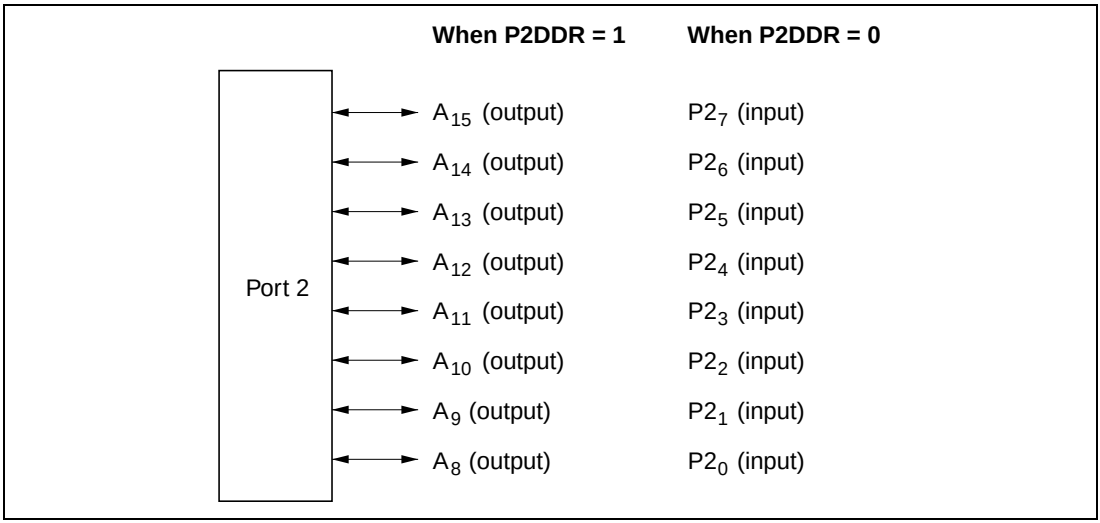


Figure 7.7 Pin Functions in Modes 1 and 3 (Port 2)

Modes 6 and 7

Input or output can be selected separately for each pin in port 2. A pin becomes an output pin if the corresponding P2DDR bit is set to 1, and an input pin if this bit is cleared to 0. Figure 7.8 shows the pin functions in modes 6 and 7.

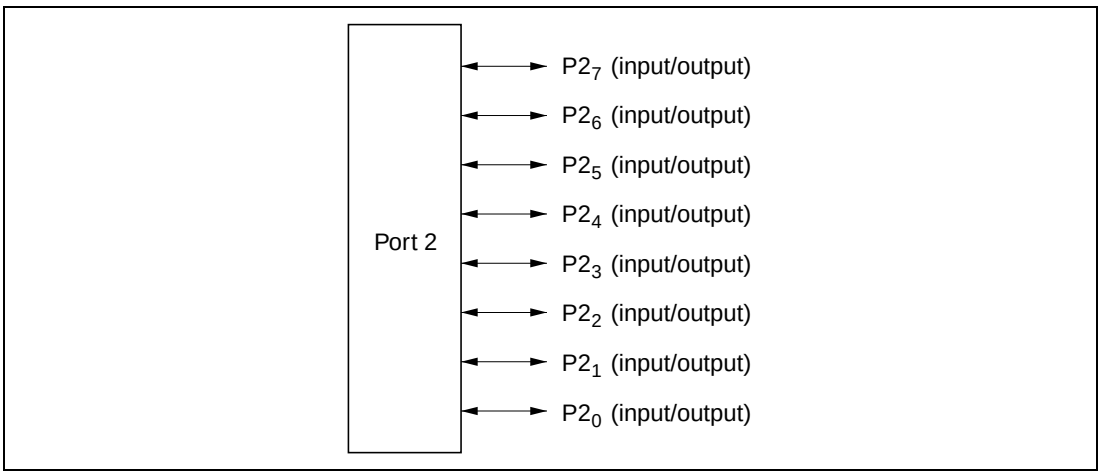


Figure 7.8 Pin Functions in Modes 6 and 7 (Port 2)

7.3.4 Input Pull-Up Transistors

Port 2 has built-in MOS input pull-up transistors that can be controlled by software. These input pull-up transistors can be turned on and off individually.

When a P2PCR bit is set to 1 and the corresponding P2DDR bit is cleared to 0, the input pull-up transistor is turned on.

The input pull-up transistors are turned off by a reset and in hardware standby mode. In software standby mode they retain their previous state.

Table 7.4 summarizes the states of the input pull-up transistors in each mode.

Table 7.4 Input Pull-Up Transistor States (Port 2)

Mode	Reset	Hardware Standby Mode	Software Standby Mode	Other Modes
1 3	Off	Off	Off	Off
5 6 7	Off	Off	On/Off	On/Off

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

7.4 Port 3

7.4.1 Overview

Port 3 is an 8-bit input/output port with the pin configuration shown in figure 7.9. Port 3 is a data bus in modes 1, 3 and 5 (expanded modes) and a generic input/output port in mode 6 and 7 (single-chip mode).

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

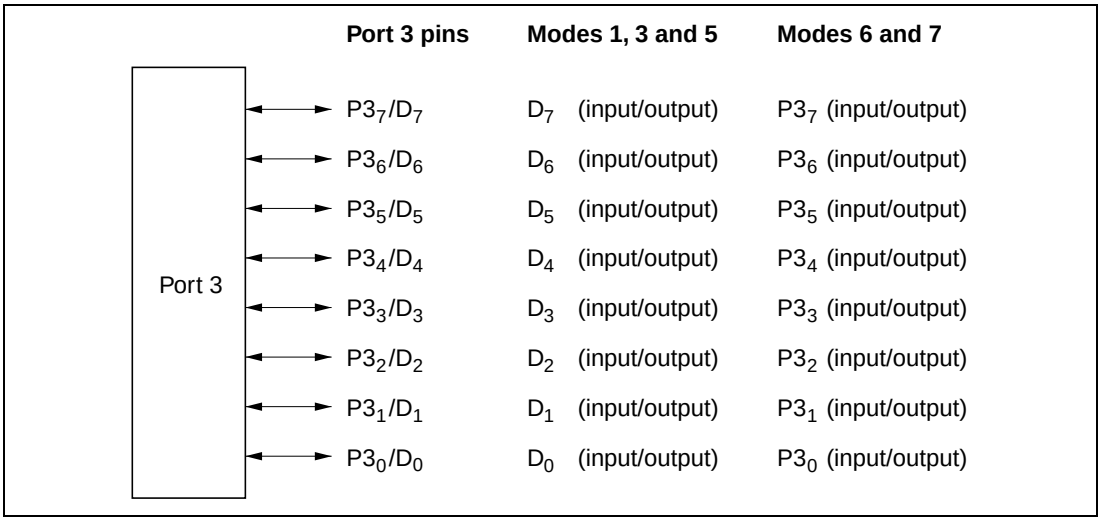


Figure 7.9 Port 3 Pin Configuration

7.4.2 Register Descriptions

Table 7.5 summarizes the registers of port 3.

Table 7.5 Port 3 Registers

Address*	Name	Abbreviation	R/W	Initial Value
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 Register (P3DDR)

P3DDR is an 8-bit write-only register that can select input or output for each pin in port 3.

Bit	7	6	5	4	3	2	1	0
	$P3_7DDR$	$P3_6DDR$	$P3_5DDR$	$P3_4DDR$	$P3_3DDR$	$P3_2DDR$	$P3_1DDR$	$P3_0DDR$
Initial value	0	0	0	0	0	0	0	0
Read/Write	W	W	W	W	W	W	W	W

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

Modes 1, 3, and 5: Port 3 functions as a data bus. P3DDR is ignored.

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

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

P3DDR is initialized to H'00 by a reset and in hardware standby mode. In software standby mode it retains its previous setting. If a P3DDR bit is set to 1, the corresponding pin maintains its output state in software standby mode.

Port 3 Data Register (P3DR)

P3DR is an 8-bit readable/writable register that stores data for pins $P3_7$ to $P3_0$.

Bit	7	6	5	4	3	2	1	0
	$P3_7$	$P3_6$	$P3_5$	$P3_4$	$P3_3$	$P3_2$	$P3_1$	$P3_0$
Initial value	0	0	0	0	0	0	0	0
Read/Write	R/W	R/W	R/W	R/W	R/W	R/W	R/W	R/W

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

When a bit in P3DDR is set to 1, if port 3 is read the value of the corresponding P3DR bit is returned directly, regardless of the actual state of the pin. When a bit in P3DDR is cleared to 0, if port 3 is read the corresponding pin level is read.

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

7.4.3 Pin Functions in Each Mode

The pin functions of port 3 differ between modes 1, 3 and 5 and modes 6 and 7. The pin functions in each mode are described below.

Modes 1, 3 and 5

All pins of port 3 automatically become data input/output pins. Figure 7.10 shows the pin functions in modes 1, 3 and 5.

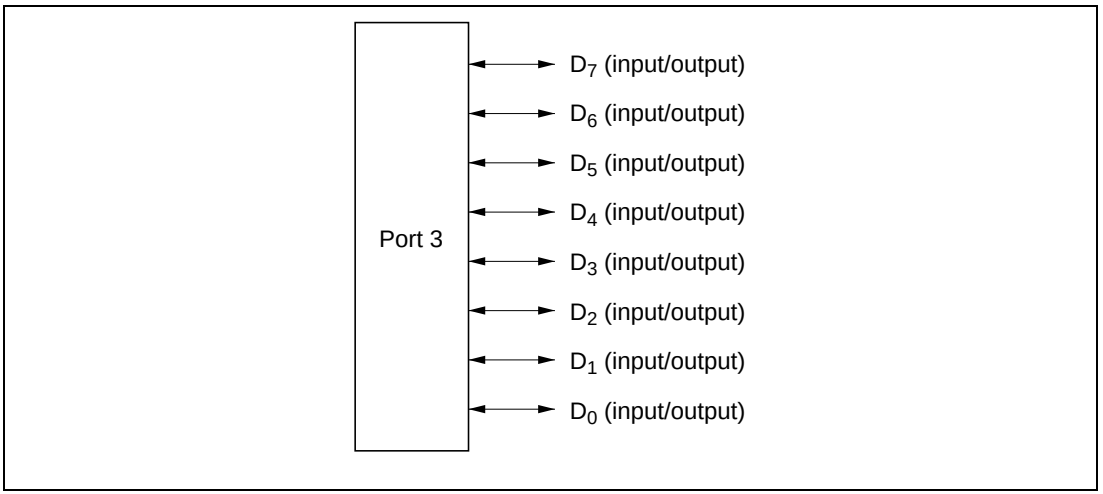


Figure 7.10 Pin Functions in Modes 1, 3 and 5 (Port 3)

Modes 6 and 7

Input or output can be selected separately for each pin in port 3. A pin becomes an output pin if the corresponding P3DDR bit is set to 1, and an input pin if this bit is cleared to 0. Figure 7.11 shows the pin functions in modes 6 and 7.

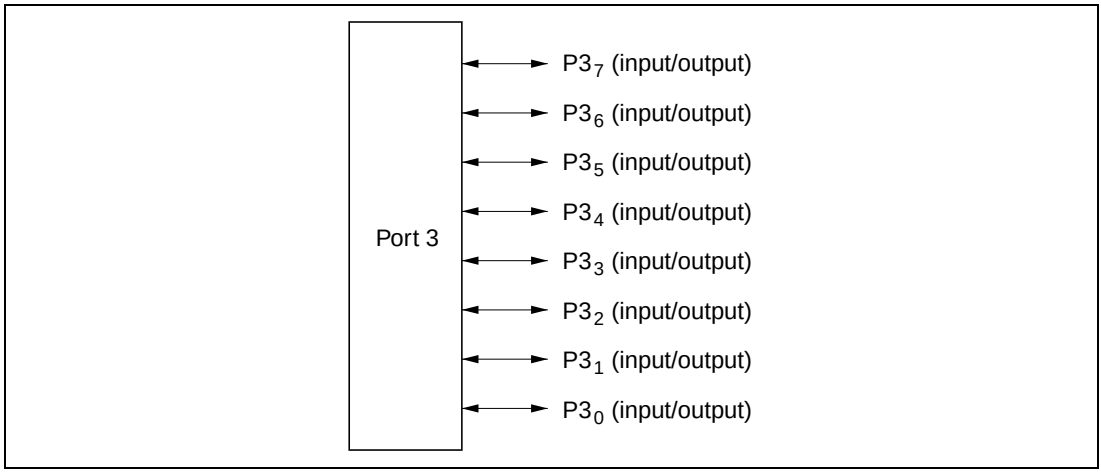


Figure 7.11 Pin Functions in Modes 6 and 7 (Port 3)

7.5 Port 5

7.5.1 Overview

Port 5 is a 4-bit input/output port with the pin configuration shown in figure 7.12. The pin functions differ depending on the operating mode.

In modes 1, 3 (expanded modes with on-chip ROM disabled), port 5 consists of address output pins (A19 to A16). In modes 5 (expanded modes with on-chip ROM enabled), settings in the port 5 data direction register (P5DDR) designate pins for address bus output (A_{19} to A_{16}) or generic input. In mode 6 and 7 (single-chip mode), port 5 is a generic input/output port.

Port 5 has software-programmable built-in pull-up transistors. Port 5 can drive one TTL load and a 90-pF capacitive load. They can also drive an LED or a Darlington transistor pair.

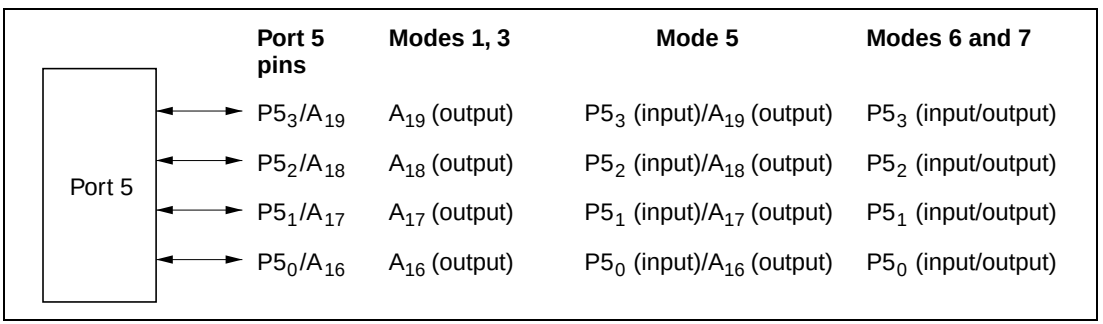


Figure 7.12 Port 5 Pin Configuration

7.5.2 Register Descriptions

Table 7.6 summarizes the registers of port 5.

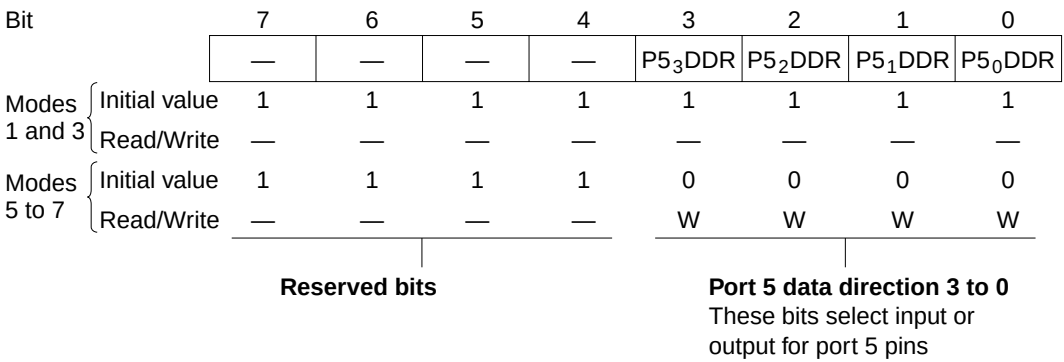
Table 7.6 Port 5 Registers

Address*	Name	Abbreviation	R/W	Initial Value	
				Modes 1 and 3	Modes 5 to 7
H'FFC8	Port 5 data direction register	P5DDR	W	H'FF	H'F0
H'FFCA	Port 5 data register	P5DR	R/W	H'F0	H'F0
H'FFDB	Port 5 input pull-up control register	P5PCR	R/W	H'F0	H'F0

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 port 5.



P5DDR is initialized to H'F0 by a reset and in hardware standby mode. In software standby mode it retains its previous setting. If a P5DDR bit is set to 1, the corresponding pin maintains its output state in software standby mode.

Port 5 Data Register (P5DR)

P5DR is an 8-bit readable/writable register that stores data for pins $P5_3$ to $P5_0$.

Bit	7	6	5	4	3	2	1	0
	—	—	—	—	$P5_3$	$P5_2$	$P5_1$	$P5_0$
Initial value	1	1	1	1	0	0	0	0
Read/Write	—	—	—	—	R/W	R/W	R/W	R/W

Reserved bits

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

When a bit in P5DDR is set to 1, if port 5 is read the value of the corresponding P5DR bit is returned directly, regardless of the actual state of the pin. When a bit in P5DDR is cleared to 0, if port 5 is read the corresponding pin level is read.

Bits $P5_7$ to $P5_4$ are reserved. They cannot be modified and are always read as 1.

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

Port 5 Input Pull-Up Control Register (P5PCR)

P5PCR is an 8-bit readable/writable register that controls the MOS input pull-up transistors in port 5.

Bit	7	6	5	4	3	2	1	0
	—	—	—	—	$P5_3PCR$	$P5_2PCR$	$P5_1PCR$	$P5_0PCR$
Initial value	1	1	1	1	0	0	0	0
Read/Write	—	—	—	—	R/W	R/W	R/W	R/W

Reserved bits

Port 5 input pull-up control 3 to 0
These bits control input pull-up transistors built into port 5

When a P5DDR bit is cleared to 0 (selecting generic input) in modes 5 to 7, if the corresponding bit from $P5_3$PCR to $P5_0$PCR is set to 1, the input pull-up transistor is turned on.

P5PCR is initialized to H'F0 by a reset and in hardware standby mode. In software standby mode it retains its previous setting.

7.5.3 Pin Functions in Each Mode

The pin functions differ between mode 1, 3 (expanded modes with on-chip ROM disabled), mode 5 (expanded modes with on-chip ROM enabled), mode 6, and 7 (single-chip mode). The pin functions in each mode are described below.

Modes 1 and 3

Address output can be selected for each pin in port 5. Figure 7.13 shows the pin functions in modes 1 and 3.

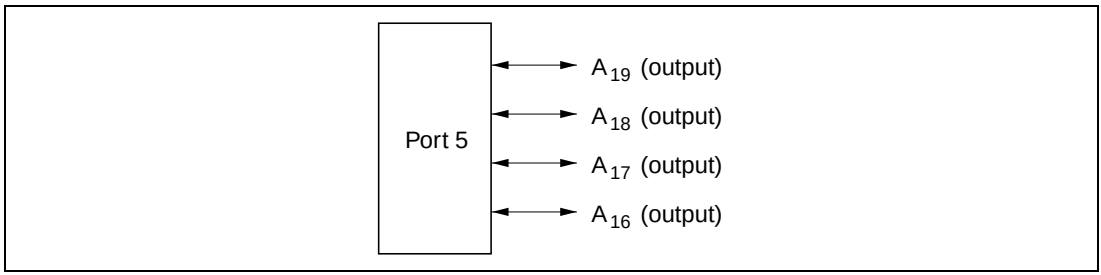


Figure 7.13 Pin Functions in Modes 1 and 3 (Port 5)

Mode 5

Address output or generic input can be selected for each pin in port 5. A pin becomes an address output pin if the corresponding P5DDR bit is set to 1, and a generic input pin if this bit is cleared to 0. Following a reset, all pins are input pins. To use a pin for address output, its P5DDR must be set to 1. Figure 7.14 shows the pin functions in mode 5.

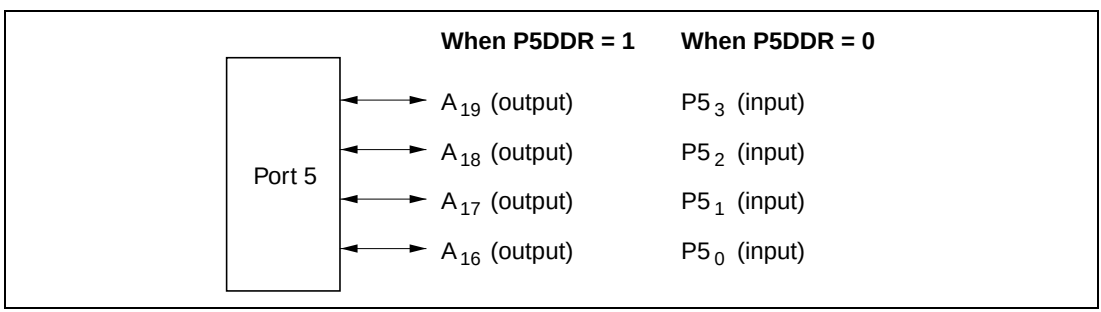


Figure 7.14 Pin Functions in Mode 5 (Port 5)

Modes 6 and 7

Input or output can be selected separately for each pin in port 5. A pin becomes an output pin if the corresponding P5DDR bit is set to 1, and an input pin if this bit is cleared to 0. Figure 7.15 shows the pin functions in modes 6 and 7.

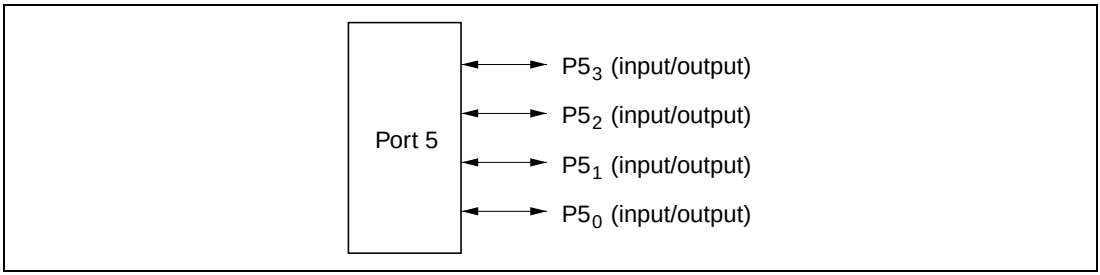


Figure 7.15 Pin Functions in Mode 6 and 7 (Port 5)

7.5.4 Input Pull-Up Transistors

Port 5 has built-in MOS input pull-up transistors that can be controlled by software. These input pull-up transistors can be turned on and off individually.

When a P5PCR bit is set to 1 and the corresponding P5DDR bit is cleared to 0, the input pull-up transistor is turned on.

The input pull-up transistors are turned off by a reset and in hardware standby mode. In software standby mode they retain their previous state.

Table 7.7 summarizes the states of the input pull-up transistors in each mode.

Table 7.7 Input Pull-Up Transistor States (Port 5)

Mode	Reset	Hardware Standby Mode	Software Standby Mode	Other Modes
1 3	Off	Off	Off	Off
5 6 7	Off	Off	On/Off	On/Off

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

7.6 Port 6

7.6.1 Overview

Port 6 is a 4-bit input/output port that is also used for input and output of bus control signals ($\overline{WR}$, $\overline{RD}$, $\overline{AS}$, and $\overline{WAIT}$).

Figure 7.16 shows the pin configuration of port 6. In modes 1, 3 and 5, the pin functions are $\overline{WR}$, $\overline{RD}$, $\overline{AS}$, and $P6_0/\overline{WAIT}$. In modes 6 and 7, port 6 is a generic input/output port.

Pins in port 6 can drive one TTL load and a 90-pF capacitive load. They can also drive a Darlington transistor pair.

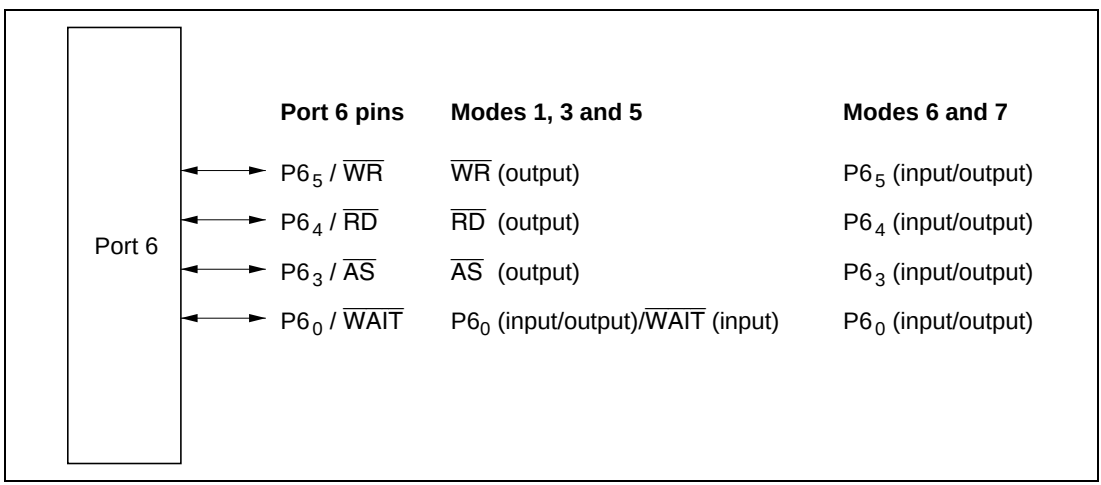


Figure 7.16 Port 6 Pin Configuration

7.6.2 Register Descriptions

Table 7.8 summarizes the registers of port 6.

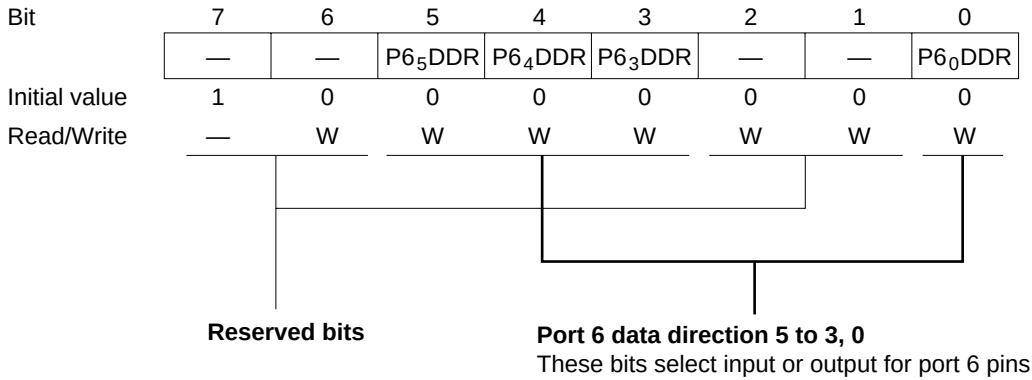
Table 7.8 Port 6 Registers

Address*	Name	Abbreviation	R/W	Initial Value	
				Modes 1, 3, and 5	Modes 6 and 7
H'FFC9	Port 6 data direction register	P6DDR	W	H'F8	H'80
H'FFCB	Port 6 data register	P6DR	R/W	H'80	H'80

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 port 6.



Bits 7, 6, 2, and 1 are reserved.

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 standby mode it retains its previous setting. If a P6DDR bit is set to 1, the corresponding pin maintains its output state in software standby mode.

Port 6 Data Register (P6DR)

P6DR is an 8-bit readable/writable register that stores data for pins $P6_5$ to $P6_3$ and $P6_0$.

Bit	7	6	5	4	3	2	1	0
	—	—	$P6_5$	$P6_4$	$P6_3$	—	—	$P6_0$
Initial value	1	0	0	0	0	0	0	0
Read/Write	—	R/W	R/W	R/W	R/W	R/W	R/W	R/W

Reserved bits

Port 6 data 5 to 3, 0
These bits store data for port 6 pins

When a bit in P6DDR is set to 1, if port 6 is read the value of the corresponding P6DR bit is returned directly. When a bit in P6DDR is cleared to 0, if port 6 is read the corresponding pin level is read. Bits 7, 6, 2, and 1 are reserved. Bit 7 cannot be modified and always reads 1. Bits 6, 2, and 1 can be written and read, but cannot be used as ports. If bit 6, 2, or 1 in P6DDR is read while its value is 1, the value of the corresponding bit in P6DR will be read. If bit 6, 2, or 1 in P6DDR is read while its value is 0, it will always read 1.

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

7.6.3 Pin Functions in Each Mode

Modes 1, 3, and 5

$P6_5$ to $P6_3$ function as bus control output pins. $P6_0$ is either a bus control input pin or generic input/output pin, functioning as an output pin when bit $P6_0$DDR is set to 1 and an input pin when this bit is cleared to 0. Figure 7.17 and table 7.9 indicate the pin functions in modes 1, 3, and 5.

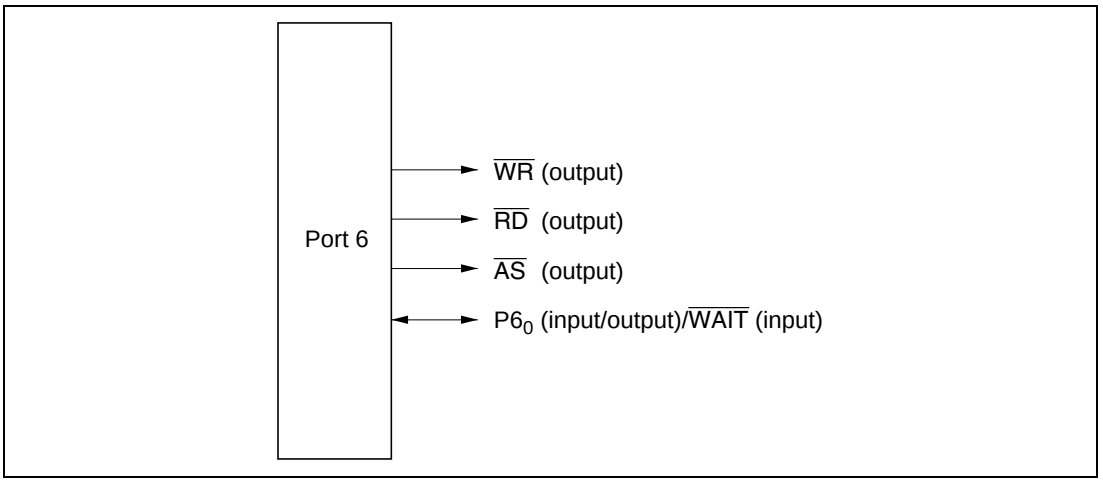


Figure 7.17 Pin Functions in Modes 1, 3, and 5 (Port 6)

Table 7.9 Port 6 Pin Functions in Modes 1, 3, and 5

Pin	Pin Functions and Selection Method			
$P6_5/\overline{WR}$	Functions as follows regardless of $P6_5DDR$			
	$P6_5DDR$	0		1
	Pin function	$\overline{WR}$ output		
$P6_4/\overline{RD}$	Functions as follows regardless of $P6_4DDR$			
	$P6_4DDR$	0		1
	Pin function	$\overline{RD}$ output		
$P6_3/\overline{AS}$	Functions as follows regardless of $P6_3DDR$			
	$P6_3DDR$	0		1
	Pin function	$\overline{AS}$ output		
$P6_0/\overline{WAIT}$	Bits WCE7 to WCE0 in WCER, bit WMS1 in WCR, and bit $P6_0DDR$ select the pin function as follows			
	WCER	All 1s		Not all 1s
	WMS1	0	1	—
	$P6_0DDR$	0	1	0*	0*
	Pin function	$P6_0$ input	$P6_0$ output	$\overline{WAIT}$ input	
	Note: * Do not set bit $P6_0DDR$ to 1.			

Modes 6 and 7

Input or output can be selected separately for each pin in port 6. A pin becomes an output pin if the corresponding P6DDR bit is set to 1, and an input pin if this bit is cleared to 0. Figure 7.18 shows the pin functions in modes 6 and 7.

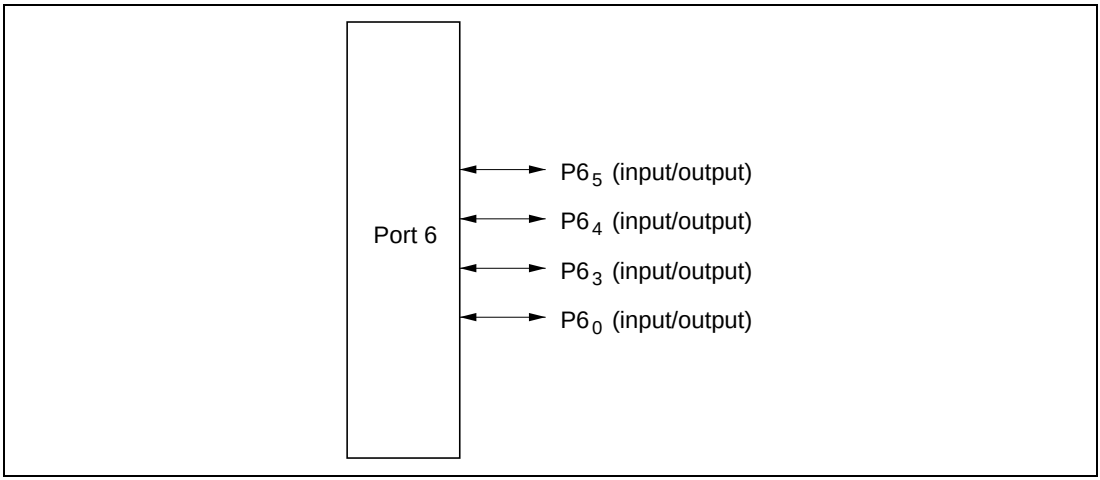


Figure 7.18 Pin Functions in Modes 6 and 7 (Port 6)

7.7 Port 7

7.7.1 Overview

Port 7 is an 8-bit input port that is also used for analog input to the A/D converter. The pin functions are the same in all operating modes. Figure 7.19 shows the pin configuration of port 7.

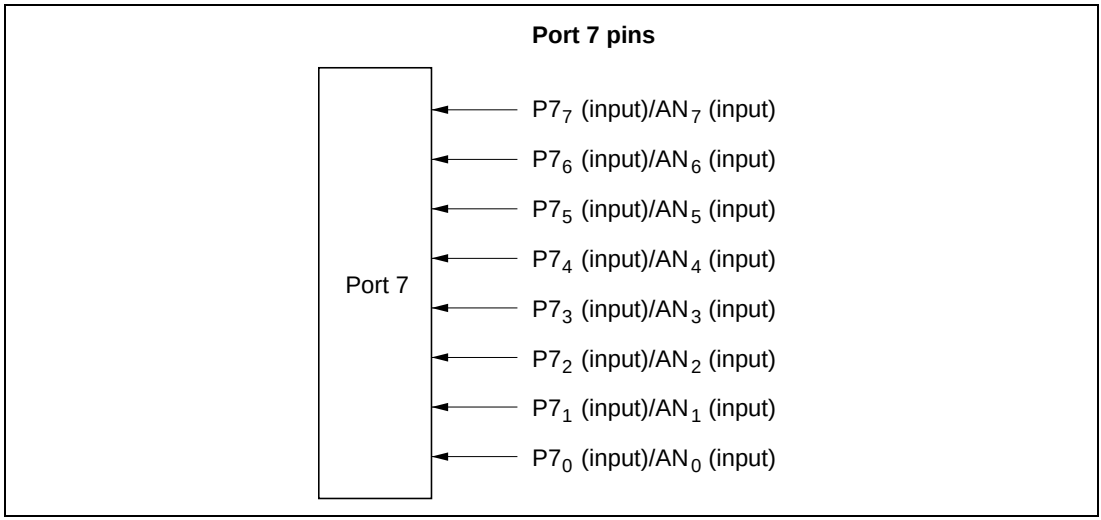


Figure 7.19 Port 7 Pin Configuration

7.7.2 Register Description

Table 7.10 summarizes the port 7 register. Port 7 is an input-only port, so it has no data direction register.

Table 7.10 Port 7 Data Register

Address*	Name	Abbreviation	R/W	Initial Value
H'FFCE	Port 7 data register	P7DR	R	Undetermined

Note: * Lower 16 bits of the address.

Port 7 Data Register (P7DR)

Bit	7	6	5	4	3	2	1	0
	$P7_7$	$P7_6$	$P7_5$	$P7_4$	$P7_3$	$P7_2$	$P7_1$	$P7_0$
Initial value	—*	—*	—*	—*	—*	—*	—*	—*
Read/Write	R	R	R	R	R	R	R	R

Note: * Determined by pins P_7 to P_0.

When P7DR is read, the pin levels are always read.

7.8 Port 8

7.8.1 Overview

Port 8 is a 2-bit input/output port that is also used for $\overline{IRQ}_1$ and $\overline{IRQ}_0$ input. Figure 7.20 shows the pin configuration of port 8.

Pin $P8_0$ functions as input/output pin or as an $\overline{IRQ}_0$ input pin. Pins $P8_1$ function as either input pins or $\overline{IRQ}_1$ input pins in modes 1, 3, and 5, and as input/output pins or $\overline{IRQ}_1$ input pins in modes 6 and 7.

Pins in port 8 can drive one TTL load and a 90-pF capacitive load. They can also drive a Darlington transistor pair. Pins $P8_1$ and $P8_0$ have Schmitt-trigger inputs.

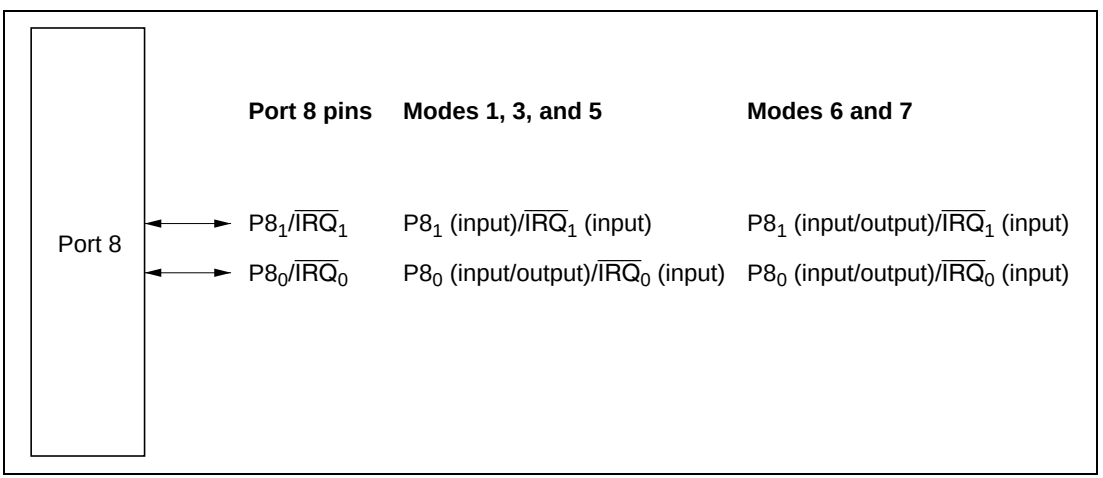


Figure 7.20 Port 8 Pin Configuration

7.8.2 Register Descriptions

Table 7.11 summarizes the registers of port 8.

Table 7.11 Port 8 Registers

Address*	Name	Abbreviation	R/W	Initial Value
H'FFCD	Port 8 data direction register	P8DDR	W	H'E0
H'FFCF	Port 8 data register	P8DR	R/W	H'E0

Note: * Lower 16 bits of the address.

Port 8 Data Direction Register (P8DDR)

P8DDR is an 8-bit write-only register that can select input or output for each pin in port 8.

	7	6	5	4	3	2	1	0
Bit	—	—	—	—	—	—	$P8_1DDR$	$P8_0DDR$
Initial value	1	1	1	0	0	0	0	0
Read/Write	—	—	—	W	W	W	W	W

Reserved bits

Port 8 data direction 1 and 0
These bits select input or output for port 8 pins

P8DDR is initialized to H'E0 by a reset and in hardware standby mode. In software standby mode it retains its previous setting. If a P8DDR bit is set to 1, the corresponding pin maintains its output state in software standby mode.

Port 8 Data Register (P8DR)

P8DR is an 8-bit readable/writable register that stores data for pins $P8_1$ to $P8_0$.

Bit	7	6	5	4	3	2	1	0
	—	—	—	—	—	—	$P8_1$	$P8_0$
Initial value	1	1	1	0	0	0	0	0
Read/Write	—	—	—	R/W	R/W	R/W	R/W	R/W

Reserved bits

Port 8 data 1 and 0
These bits store data for port 8 pins

When a bit in P8DDR is set to 1, if port 8 is read the value of the corresponding P8DR bit is returned directly. When a bit in P8DDR is cleared to 0, if port 8 is read the corresponding pin level is read.

Bits 7 to 2 are reserved. Bits 7 to 5 cannot be modified and always read 1. Bit 4, 3, and 2 can be written and read, but it cannot be used for port input or output. If bit 4, 3, and 2 of P8DDR is read while its value is 1, bit 4, 3 and 2 of P8DR is read directly. If bit 4, 3, and 2 of P8DDR is read while its value is 0, it always reads 1.

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

7.8.3 Pin Functions

The port 8 pins are also used for $\overline{IRQ}_1$ and $\overline{IRQ}_0$. Table 7.12 describes the selection of pin functions.

Table 7.12 Port 8 Pin Functions

Pin	Pin Functions and Selection Method			
$P8_1/\overline{IRQ}_1$	Bit $P8_1$DDR selects the pin function as follows			
	$P8_1$DDR	0	1	
			Modes 1, 3, and 5	Modes 6 and 7
	Pin function	$P8_1$ input	Illegal setting	$P8_1$ output
		$\overline{IRQ}_1$ input		
$P8_0/\overline{IRQ}_0$	Bit $P8_0$DDR selects the pin function as follows			
	$P8_0$DDR	0	1	
	Pin function	$P8_0$ input	$P8_0$ output	
		$\overline{IRQ}_0$ input		

7.9 Port 9

7.9.1 Overview

Port 9 is a 6-bit input/output port that is also used for input and output (TxD_0, TxD_1, RxD_0, RxD_1, SCK_0, SCK_1) by serial communication interface channels 0 and 1 (SCI0 and SCI1), and for $\overline{IRQ}_5$ and $\overline{IRQ}_4$ input.

Port 9 has the same set of pin functions in all operating modes. Figure 7.21 shows the pin configuration of port 9.

Pins in port 9 can drive one TTL load and a 30-pF capacitive load. They can also drive a Darlington transistor pair.

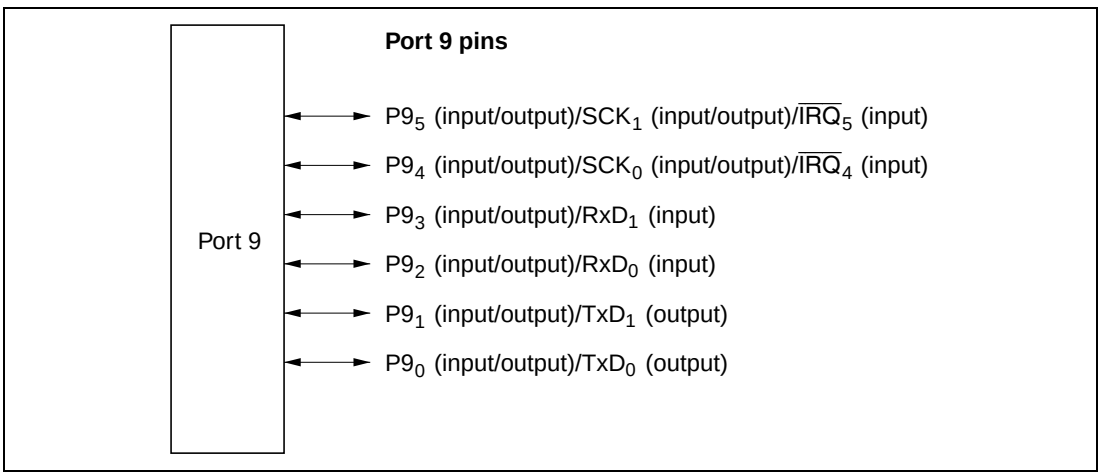


Figure 7.21 Port 9 Pin Configuration

7.9.2 Register Descriptions

Table 7.13 summarizes the registers of port 9.

Table 7.13 Port 9 Registers

Address*	Name	Abbreviation	R/W	Initial Value
H'FFD0	Port 9 data direction register	P9DDR	W	H'C0
H'FFD2	Port 9 data register	P9DR	R/W	H'C0

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 port 9.

Bit	7	6	5	4	3	2	1	0
	—	—	$P9_5$DDR	$P9_4$DDR	$P9_3$DDR	$P9_2$DDR	$P9_1$DDR	$P9_0$DDR
Initial value	1	1	0	0	0	0	0	0
Read/Write	—	—	W	W	W	W	W	W

Reserved bits

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

A pin in port 9 becomes an output pin if the corresponding P9DDR bit is set to 1, and an input pin if this bit is cleared to 0.

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 standby mode it retains its previous setting. If a P9DDR bit is set to 1, the corresponding pin maintains its output state in software standby mode.

Port 9 Data Register (P9DR)

P9DR is an 8-bit readable/writable register that stores output data for pins $P9_5$ to $P9_0$.

Bit	7	6	5	4	3	2	1	0
	—	—	$P9_5$	$P9_4$	$P9_3$	$P9_2$	$P9_1$	$P9_0$
Initial value	1	1	0	0	0	0	0	0
Read/Write	—	—	R/W	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

When a bit in P9DDR is set to 1, if port 9 is read the value of the corresponding P9DR bit is returned. When a bit in P9DDR is cleared to 0, if port 9 is read the corresponding pin level is read.

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 standby mode it retains its previous setting.

7.9.3 Pin Functions

The port 9 pins are also used for SCI input and output (TxD, RxD, SCK), and for $\overline{IRQ}_5$ and $\overline{IRQ}_4$ input. Table 7.14 describes the selection of pin functions.

Table 7.14 Port 9 Pin Functions

Pin	Pin Functions and Selection Method
$P9_5/SCK_1/\overline{IRQ}_5$	Bit C/$\overline{A}$ in SMR of SCI1, bits CKE0 and CKE1 in SCR of SCI1, and bit $P9_5$DDR select the pin function as follows

CKE1	0				1
C/$\overline{A}$	0			1	—
CKE0	0		1	—	—
$P9_5$DDR	0	1	—	—	—
Pin function	$P9_5$ input	$P9_5$ output	SCK_1 output	SCK_1 output	SCK_1 input
	$\overline{IRQ}_5$ input				

$P9_4/SCK_0/\overline{IRQ}_4$	Bit C/$\overline{A}$ in SMR of SCI0, bits CKE0 and CKE1 in SCR of SCI, and bit $P9_4$DDR select the pin function as follows

CKE1	0				1
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	SCK_0 input
	$\overline{IRQ}_4$ input				

Pin	Pin Functions and Selection Method

$P9_3/RxD_1$ Bit RE in SCR of SCI1 and bit $P9_3DDR$ select the pin function as follows

RE	0		1
$P9_3DDR$	0	1	—
Pin function	$P9_3$ input	$P9_3$ output	RxD_1 input

$P9_2/RxD_0$ Bit RE in SCR of SCI_0, bit SMIF in SCMR, and bit $P9_2DDR$ select the pin function as follows

SMIF	0			1
RE	0		1	—
$P9_2DDR$	0	1	—	—
Pin function	$P9_2$ input	$P9_2$ output	RxD_0 input	RxD_0 input

$P9_1/TxD_1$ Bit TE in SCR of SCI1 and bit P91DDR select the pin function as follows

TE	0		1
$P9_1DDR$	0	1	—
Pin function	$P9_1$ input	$P9_1$ output	TxD_1 output

$P9_0/TxD_0$ Bit TE in SCR of SCI0, bit SMIF in SCMR, and bit $P9_0DDR$ select the pin function as follows

SMIF	0			1
TE	0		1	—
$P9_0DDR$	0	1	—	—
Pin function	$P9_0$ input	$P9_0$ output	TxD_0 output	TxD_0 output*

Note: * Functions as the TxD_0 output pin, but there are two states: one in which the pin is driven, and another in which the pin is at high impedance.

7.10 Port A

7.10.1 Overview

Port A is an 8-bit input/output port that is also used for output (TP_7 to TP_0) from the programmable timing pattern controller (TPC), input and output ($TIOCB_2$, $TIOCA_2$, $TIOCB_1$, $TIOCA_1$, $TIOCB_0$, $TIOCA_0$, TCLKD, TCLKC, TCLKB, TCLKA) by the 16-bit integrated timer unit (ITU), and address output (A_{23} to A_{20}). Figure 7.22 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 drive a Darlington transistor pair. Port A has Schmitt-trigger inputs.

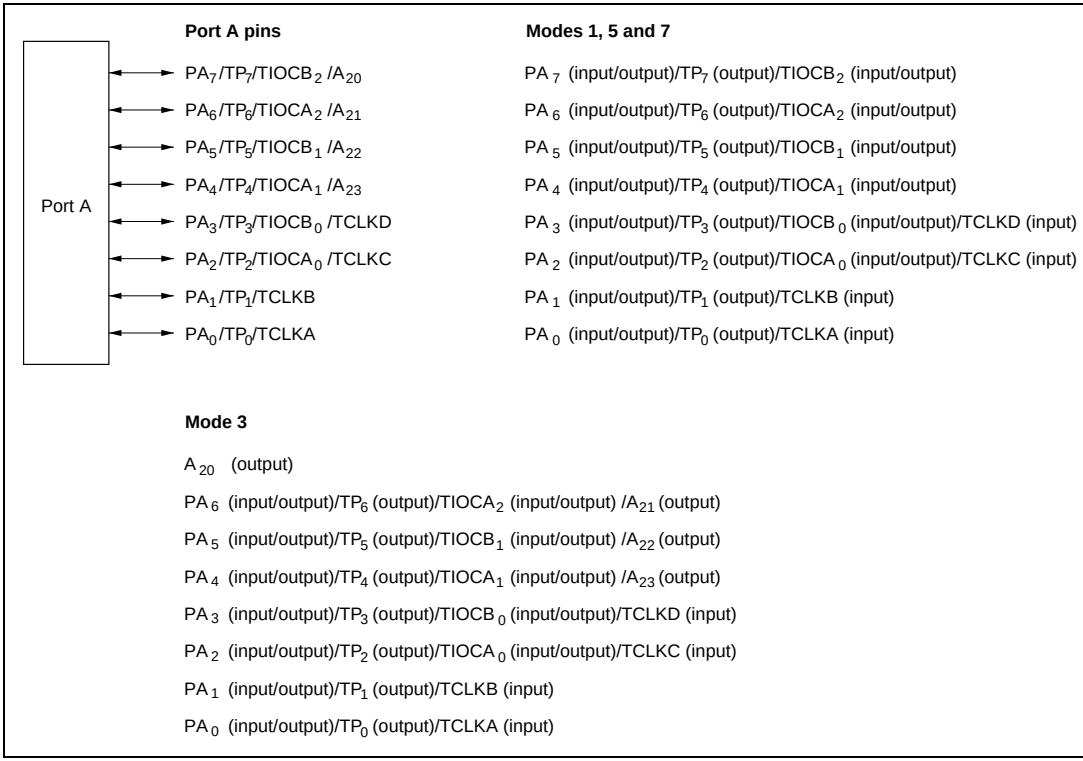


Figure 7.22 Port A Pin Configuration

7.10.2 Register Descriptions

Table 7.15 summarizes the registers of port A.

Table 7.15 Port A Registers

Address*	Name	Abbreviation	R/W	Initial Value: Modes 1, 5, and 7	Initial Value: Mode 3
H'FFD1	Port A data direction register	PADDR	W	H'00	H'80
H'FFD3	Port A data register	PADR	R/W	H'00	H'00

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 port A. The corresponding PADDR bit should also be set when a pin is used as a TPC output.

	Bit	7	6	5	4	3	2	1	0
		PA_7DDR	PA_6DDR	PA_5DDR	PA_4DDR	PA_3DDR	PA_2DDR	PA_1DDR	PA_0DDR
Modes 1, 5, and 7	Initial value	0	0	0	0	0	0	0	0
	Read/Write	W	W	W	W	W	W	W	W
Mode 3	Initial value	1	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

A pin in port A becomes an output pin if the corresponding PADDR bit is set to 1, and an input pin if this bit is cleared to 0. However, in mode 3, PA_7 DDR is fixed at 1, and PA7 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 in modes 1, 5 and 7 and to H'80 in mode 3 by a reset and in hardware standby mode. In software standby mode it retains its previous setting. If a PADDR bit is set to 1, the corresponding pin maintains its output state in software standby mode.

Port A Data Register (PADR)

PADR is an 8-bit readable/writable register that stores data for pins PA_7 to PA_0.

Bit	7	6	5	4	3	2	1	0
	PA_7	PA_6	PA_5	PA_4	PA_3	PA_2	PA_1	PA_0
Initial value	0	0	0	0	0	0	0	0
Read/Write	R/W	R/W	R/W	R/W	R/W	R/W	R/W	R/W

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

When a bit in PADDR is set to 1, if port A is read the value of the corresponding PADR bit is returned directly. When a bit in PADDR is cleared to 0, if port A is read the corresponding pin level is read.

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

When port A pins are used for TPC output, PADR stores output data for TPC output groups 0 and 1. If a bit in the next data enable register (NDERA) is set to 1, the corresponding PADR bit cannot be written. In this case, PADR can be updated only when data is transferred from NDRA.

7.10.3 Pin Functions

The port A pins are also used for TPC output (TP_7 to TP_0), ITU input/output ($TIOCB_2$ to $TIOCB_0$, $TIOCA_2$ to $TIOCA_0$) and input (TCLKD, TCLKC, TCLKB, TCLKA), and as address bus pins (A_{23} to A_{20}). Table 7.16 describes the selection of pin functions.

Table 7.16 Port A Pin Functions

Pin | **Pin Functions and Selection Method**

$PA_7/TP_7/TIOCB_2/A_{20}$

The mode setting, ITU channel 2 settings (bit PWM2 in TMDR and bits IOB2 to IOB0 in TIOR2), bit NDER7 in NDERA, and bit PA7DDR in PADDR select the pin function as follows

Mode	1, 5 to 7				3
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	A_{20} output
		TIOCB2 input*			

Note: * TIOCB2 input when IOB2 = 1 and PWM2 = 0.

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

Pin	Pin Functions and Selection Method
PA_6/TP_6/ $TIOCA_2$/ A_{21}	The mode setting, bit $A_{21}E$ in BRCR, ITU channel 2 settings (bit PWM2 in TMDR and bits IOA2 to IOA0 in TIOR2), bit NDER6 in NDERA, and bit PA_6DDR in PADDR select the pin function as follows

Mode	1, 5 to 7				3				
$A_{21}E$	—				1				0
ITU channel 2 settings	(1) in table below	(2) in table below			(1) in table below	(2) in table below			—
PA_6DDR	—	0	1	1	—	0	1	1	—
NDER6	—	—	0	1	—	—	0	1	—
Pin function	$TIOCA_2$ output	PA_6 input	PA_6 output	TP_6 output	$TIOCA_2$ output	PA_6 input	PA_6 output	TP_6 output	A_{21} output
		$TIOCA_2$ input*				$TIOCA_2$ input*			

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

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

Pin	Pin Functions and Selection Method
$PA_5/TP_5/TIOCB_1/A_{22}$	The mode setting, bit $A_{22}E$ in BRCR, ITU channel 1 settings (bit PWM1 in TMDR and bits IOB2 to IOB0 in TIOR1), bit NDER5 in NDERA, and bit PA_5DDR in PADDR select the pin function as follows

Mode	1, 5 to 7				3				
$A_{22}E$	—				1				0
ITU channel 1 settings	(1) in table below	(2) in table below			(1) in table below	(2) in 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	$TIOCB_1$ output	PA_5 input	PA_5 output	TP_5 output	A_{22} output
		TIOCB1 input*				TIOCB1 input*			

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

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

Pin	Pin Functions and Selection Method
$PA_4/TP_4/TIOCB_1/A_{23}$	The mode setting, bit $A_{23}E$ in BRCR, ITU channel 1 settings (bit PWM1 in TMDR and bits IOA2 to IOA0 in TIOR1), bit NDER4 in NDERA, and bit PA_4DDR in PADDR select the pin function as follows

Mode	1, 5 to 7				3				
$A_{23}E$	—				1				0
ITU channel 1 settings	(1) in table below	(2) in table below			(1) in table below	(2) in table below			—
PA_4DDR	—	0	1	1	—	0	1	1	—
NDER4	—	—	0	1	—	—	0	1	—
Pin function	$TIOCA_1$ output	PA_4 input	PA_4 output	TP_5 output	$TIOCA_1$ output	PA_4 input	PA_4 output	TP_4 output	A_{23} output
		$TIOCA_1$ input*				$TIOCA_1$ input*			

Note: * $TIOCA_1$ input when IOA2 = 1.

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

Pin	Pin Functions and Selection Method
PA_3/TP_3/ $TIOCB_0$/ TCLKD	ITU channel 0 settings (bit PWM0 in TMDR and bits IOB2 to IOB0 in TIOR0), bits TPSC2 to TPSC0 in TCR4 to TCR0, bit NDER3 in NDERA, and bit PA3DDR in PADDR select the pin function as follows

ITU channel 0 settings	(1) in table below	(2) in table below		
PA_3DDR	—	0	1	1
NDER3	—	—	0	1
Pin function	$TIOCB_0$ output	PA_3 input	PA_3 output	TP_3 output
		$TIOCB_0$ input*1		
	TCLKD input*2			

Notes: 1. TIOCB0 input when IOB2 = 1 and PWM0 = 0.
2. TCLKD input when TPSC2 = TPSC1 = TPSC0 = 1 in any of TCR4 to TCR0.

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

Pin	Pin Functions and Selection Method
$PA_2/TP_2/TIOCA_0/TCLKC$	ITU channel 0 settings (bit PWM0 in TMDR and bits IOA2 to IOA0 in TIOR0), bits TPSC2 to TPSC0 in TCR4 to TCR0, bit NDER2 in NDERA, and bit PA_2DDR in PADDR select the pin function as follows

ITU channel 0 settings	(1) in table below	(2) in table below		
PA_2DDR	—	0	1	1
NDER2	—	—	0	1
Pin function	$TIOCA_0$ output	PA_2 input	PA_2 output	TP_2 output
		$TIOCA_0$ 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 of TCR4 to TCR0.

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

Pin	Pin Functions and Selection Method
$PA_1/TP_1/$ TCLKB	Bit NDER1 in NDERA and bit PA1DDR in PADDR select the pin function as follows

PA_1DDR	0	1	1
NDER1	—	0	1
Pin function	PA_1 input	PA_1 output	TP_1 output
	TCLKB input*		

Note: * TCLKB input when MDF = 1 in TMDR, or when TPSC2 = 1, TPSC1 = 0, and TPSC0 = 1 in any of TCR4 to TCR0.

$PA_0/TP_0/$ TCLKA — Bit NDER0 in NDERA and bit PA0DDR in PADDR select the pin function as follows

PA_0DDR	0	1	1
NDER0	—	0	1
Pin function	PA_0 input	PA_0 output	TP_0 output
	TCLKA input*		

Note: * TCLKA input when MDF = 1 in TMDR, or when TPSC2 = 1 and TPSC1 = TPSC0 = 0 in any of TCR4 to TCR0.

7.11 Port B

7.11.1 Overview

Port B is a 7-bit input/output port that is also used for TPC output (TP_{15}, TP_{13} to TP_{8}), ITU input/output ($TIOCB_4$, $TIOCB_3$, $TIOCA_4$, $TIOCA_3$) and ITU output ($TOCXB_4$, $TOCXA_4$), and $\overline{\text{ADTRG}}$ input to the A/D converter. Port B has the same set of pin functions in all operating modes. Figure 7.23 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 drive an LED or a Darlington transistor pair. Pins PB_3 to PB_0 have Schmitt-trigger inputs.

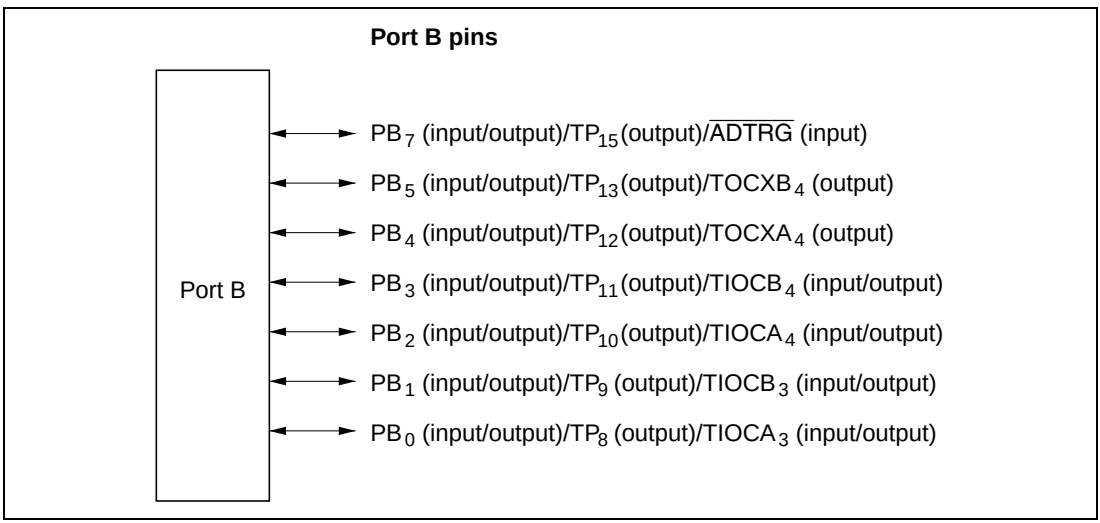


Figure 7.23 Port B Pin Configuration

7.11.2 Register Descriptions

Table 7.17 summarizes the registers of port B.

Table 7.17 Port B Registers

Address*	Name	Abbreviation	R/W	Initial Value
H'FFD4	Port B data direction register	PBDDR	W	H'00
H'FFD6	Port B data register	PBDR	R/W	H'00

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 port B.

Bit	7	6	5	4	3	2	1	0
	PB_7DDR	—	PB_5DDR	PB_4DDR	PB_3DDR	PB_2DDR	PB_1DDR	PB_0DDR
Initial value	0	0	0	0	0	0	0	0
Read/Write	W	W	W	W	W	W	W	W

Reserved bit

Port B data 7, 5 to 0
These bits select input or output for port B pins

A pin in port B becomes an output pin if the corresponding PBDDR bit is set to 1, and an input pin if this bit is cleared to 0.

Bit 6 is reserved.

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 standby mode it retains its previous setting. If a PBDDR bit is set to 1, the corresponding pin maintains its output state in software standby mode.

Port B Data Register (PBDR)

PBDR is an 8-bit readable/writable register that stores data for pins PB_7, PB_5 to PB_0.

Bit	7	6	5	4	3	2	1	0
	PB_7	—	PB_5	PB_4	PB_3	PB_2	PB_1	PB_0
Initial value	0	0	0	0	0	0	0	0
Read/Write	R/W	R/W	R/W	R/W	R/W	R/W	R/W	R/W

Reserved bit

Port B data 7, 5 to 0
These bits store data for port B pins

When a bit in PBDDR is set to 1, if port B is read the value of the corresponding PBDR bit is returned directly. When a bit in PBDDR is cleared to 0, if port B is read the corresponding pin level is read. Bit 6 is reserved. Bit 6 can be written and read, but cannot be used for a port input or output.

If bit 6 in PBDDR is read while its value is 1, the value of bit 6 in PBDR will be read directly. If bit 6 in PBDDR is read while its value is 0, it will always be read as 1.

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

When port B pins are used for TPC output, PBDR stores output data for TPC output groups 2 and 3. If a bit in the next data enable register (NDERB) is set to 1, the corresponding PBDR bit cannot be written. In this case, PBDR can be updated only when data is transferred from NDRB.

7.11.3 Pin Functions

The port B pins are also used for TPC output (TP_{15}, TP_{13} to TP_8), ITU input/output ($TIOCB_4$, $TIOCB_3$, $TIOCA_4$, $TIOCA_3$) and output ($TOCXB_4$, $TOCXA_4$), and $\overline{\text{ADTRG}}$ input. Table 7.18 describes the selection of pin functions.

Table 7.18 Port B Pin Functions

Pin	Pin Functions and Selection Method

PB_7/ TP_{15}/ $\overline{ADTRG}$

Bit TRGE in ADCR, bit NDER15 in NDERB and bit PB7DDR in PBDDR select the pin function as follows

PB_7DDR	0	1	1
NDER15	—	0	1
Pin function	PB_7 input	PB_7 output	TP_{15} output
	$\overline{ADTRG}$ input*		

Notes: * $\overline{ADTRG}$ input when TRGE = 1.

PB_5/ TP_{13}/ $TOCXB_4$

ITU channel 4 settings (bit CMD1 in TFCR and bit EXB4 in TOER), bit NDER13 in NDERB, and bit PB_5DDR in PBDDR select the pin function as follows

EXB4, CMD1	Not both 1			Both 1
PB_5DDR	0	1	1	—
NDER13	—	0	1	—
Pin function	PB_5 input	PB_5 output	TP_{13} output	$TOCXB_4$ output

PB_4/ TP_{12}/ $TOCXA_4$

ITU channel 4 settings (bit CMD1 in TFCR and bit EXA4 in TOER), bit NDER12 in NDERB, and bit PB4DDR in PBDDR select the pin function as follows

EXA4, CMD1	Not both 1			Both 1
PB_4DDR	0	1	1	—
NDER12	—	0	1	—
Pin function	PB_4 input	PB_4 output	TP_{12} output	$TOCXA_4$ output

Pin	Pin Functions and Selection Method
PB_3/ TP_{11}/ $TIOCB_4$	ITU channel 4 settings (bit PWM4 in TMDR, bit CMD1 in TFCR, bit EB4 in TOER, and bits IOB2 to IOB0 in TIOR4), bit NDER11 in NDERB, and bit PB_3DDR in PBDDR select the pin function as follows

ITU channel 4 settings	(1) in table below	(2) in table below		
PB_3DDR	—	0	1	1
NDER11	—	—	0	1
Pin function	$TIOCB_4$ output	PB_3 input	PB_3 output	TP_{11} output
		$TIOCB_4$ input*		

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

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

Pin	Pin Functions and Selection Method
PB_2/ TP_{10}/ $TIOCA_4$	ITU channel 4 settings (bit CMD1 in TFCR, bit EA4 in TOER, bit PWM4 in TMDR, and bits IOA2 to IOA0 in TIOR4), bit NDER10 in NDERB, and bit PB_2DDR in PBDDR select the pin function as follows

ITU channel 4 settings	(1) in table below	(2) in table below		
PB_2DDR	—	0	1	1
NDER11	—	—	0	1
Pin function	$TIOCA_4$ output	PB_2 input	PB_2 output	TP_{10} output
		$TIOCA_4$ input*		

Note: * $TIOCA_4$ input when CMD1 = PWM4 = 0 and IOB2 = 1.

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

Pin	Pin Functions and Selection Method
$PB_1/TP_9/TIOCB_3$	ITU channel 3 settings (bit PWM3 in TMDR, bit CMD1 in TFCR, bit EB3 in TOER, and bits IOB2 to IOB0 in TIOR3), bit NDER11 in NDERB, and bit PB_1DDR in PBDDR select the pin function as follows

ITU channel 3 settings	(1) in table below	(2) in table below		
PB_1DDR	—	0	1	1
NDER9	—	—	0	1
Pin function	$TIOCB_3$ output	PB_1 input	PB_1 output	TP_9 output
		$TIOCB_3$ input*		

Note: * TIOCB3 input when CMD1 = PWM3 = 0 and IOB2 = 1.

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

Pin	Pin Functions and Selection Method
$PB_0/TP_8/TIOCA_3$	ITU channel 3 settings (bit CMD1 in TFCR, bit EA3 in TOER, bit PWM3 in TMDR, and bits IOA2 to IOA0 in TIOR3), bit NDER8 in NDERB, and bit PB_0DDR in PBDDR select the pin function as follows

ITU channel 3 settings	(1) in table below	(2) in table below		
PB_0DDR	—	0	1	1
NDER8	—	—	0	1
Pin function	$TIOCA_3$ output	PB_0 input	PB_0 output	TP_8 output
		$TIOCA_3$ input*		

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

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

Section 8 16-Bit Integrated Timer Unit (ITU)

8.1 Overview

The H8/3039 Group has a built-in 16-bit integrated timer-pulse unit (ITU) with five 16-bit timer channels.

8.1.1 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 output compare or input capture functions
- Selection of eight counter clock sources for each channel:
  Internal clocks: ϕ, $\phi/2$, $\phi/4$, $\phi/8$
  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 2)
  - — 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 enables synchronous register input and output.
  - — PWM mode
    PWM output can be provided with an arbitrary duty cycle. With synchronization, up to five-phase PWM output is possible
- Phase counting mode selectable in channel 2
  Two-phase encoder output can be counted automatically.

- Three additional modes selectable in channels 3 and 4
  - — Reset-synchronized PWM mode

    If channels 3 and 4 are combined, three-phase PWM output is possible with three pairs of complementary waveforms.
  - — Complementary PWM mode

    If channels 3 and 4 are combined, three-phase PWM output is possible with three pairs of non-overlapping complementary waveforms.
  - — Buffering

    Input capture registers can be double-buffered. Output compare registers can be updated automatically.
- High-speed access via internal 16-bit bus

  The 16-bit timer counters, general registers, and buffer registers can be accessed at high speed via a 16-bit bus.
- Fifteen interrupt sources

  Each channel has two compare match/input capture interrupts and an overflow interrupt. All interrupts can be requested independently.
- Output triggering of programmable pattern controller (TPC)

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

Table 8.1 summarizes the ITU functions.

Table 8.1 ITU Functions

Item		Channel 0	Channel 1	Channel 2	Channel 3	Channel 4
Clock sources		Internal clocks: φ, φ/2, φ/4, φ/8 External clocks: TCLKA, TCLKB, TCLKC, TCLKD, selectable independently				
General registers (output compare/ input capture registers)		GRA0, GRB0	GRA1, GRB1	GRA2, GRB2	GRA3, GRB3	GRA4, GRB4
Buffer registers		—	—	—	BRA3, BRB3	BRA4, BRB4
Input/output pins		$TIOCA_0$, $TIOCB_0$	$TIOCA_1$, $TIOCB_1$	$TIOCA_2$, $TIOCB_2$	$TIOCA_3$, $TIOCB_3$	$TIOCA_4$, $TIOCB_4$
Output pins		—	—	—	—	$TOCXA_4$, $TOCXB_4$
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	GRA4/GRB4 compare match or input capture
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
Interrupt sources		Three sources • Compare match/input capture A0 • Compare match/input capture B0 • Overflow	Three sources • Compare match/input capture A1 • Compare match/input capture B1 • Overflow	Three 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	Three sources • Compare match/input capture A4 • Compare match/input capture B4 • Overflow

Legend:

O: Available

—: Not available

8.1.2 Block Diagrams

ITU Block Diagram (Overall)

Figure 8.1 is a block diagram of the ITU.

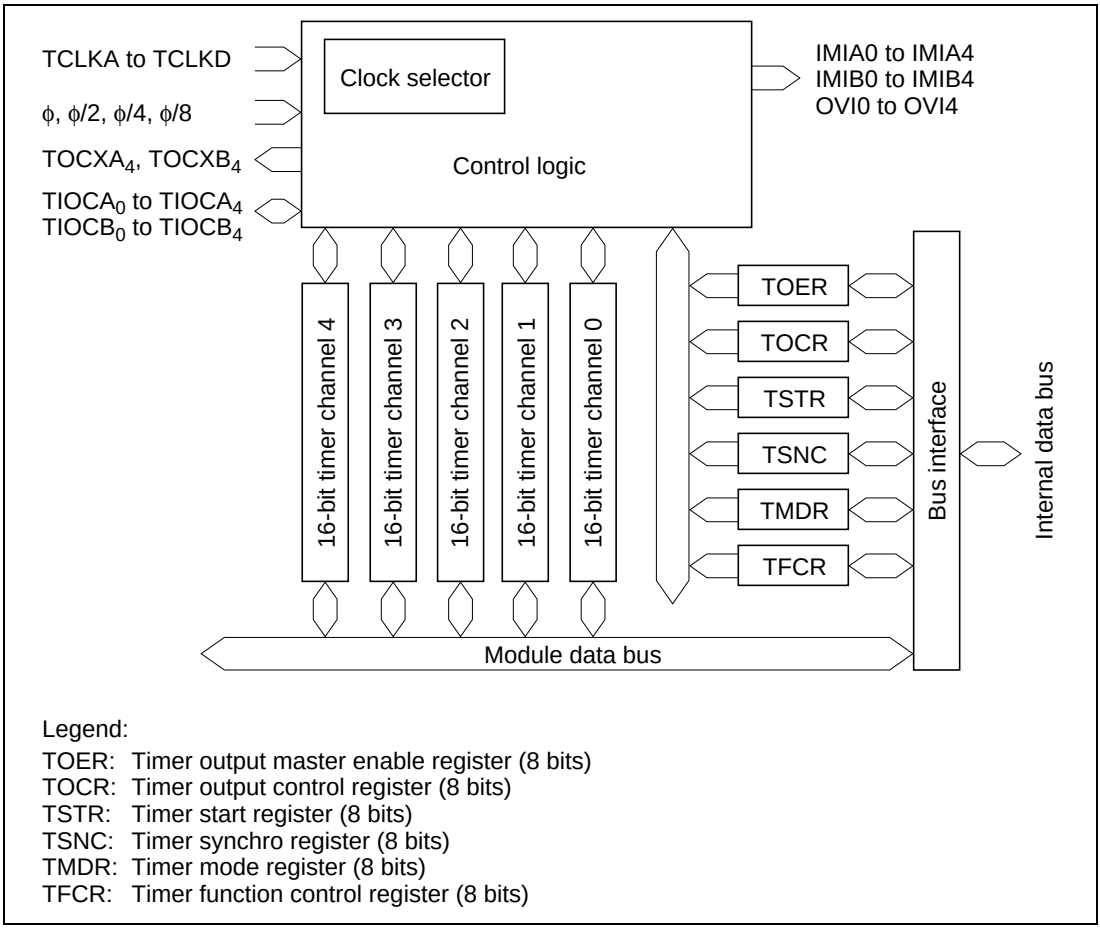
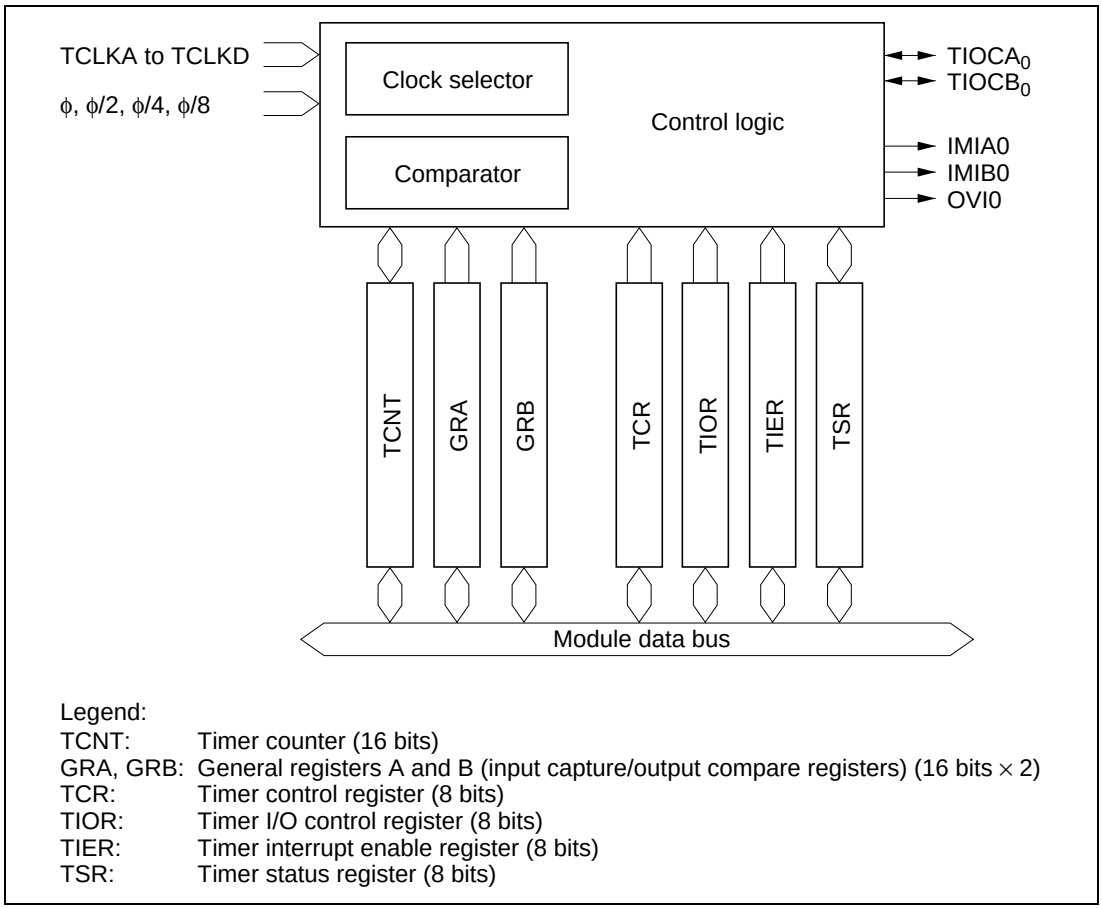


Figure 8.1 ITU Block Diagram (Overall)

Block Diagram of Channels 0 and 1

ITU channels 0 and 1 are functionally identical. Both have the structure shown in figure 8.2.



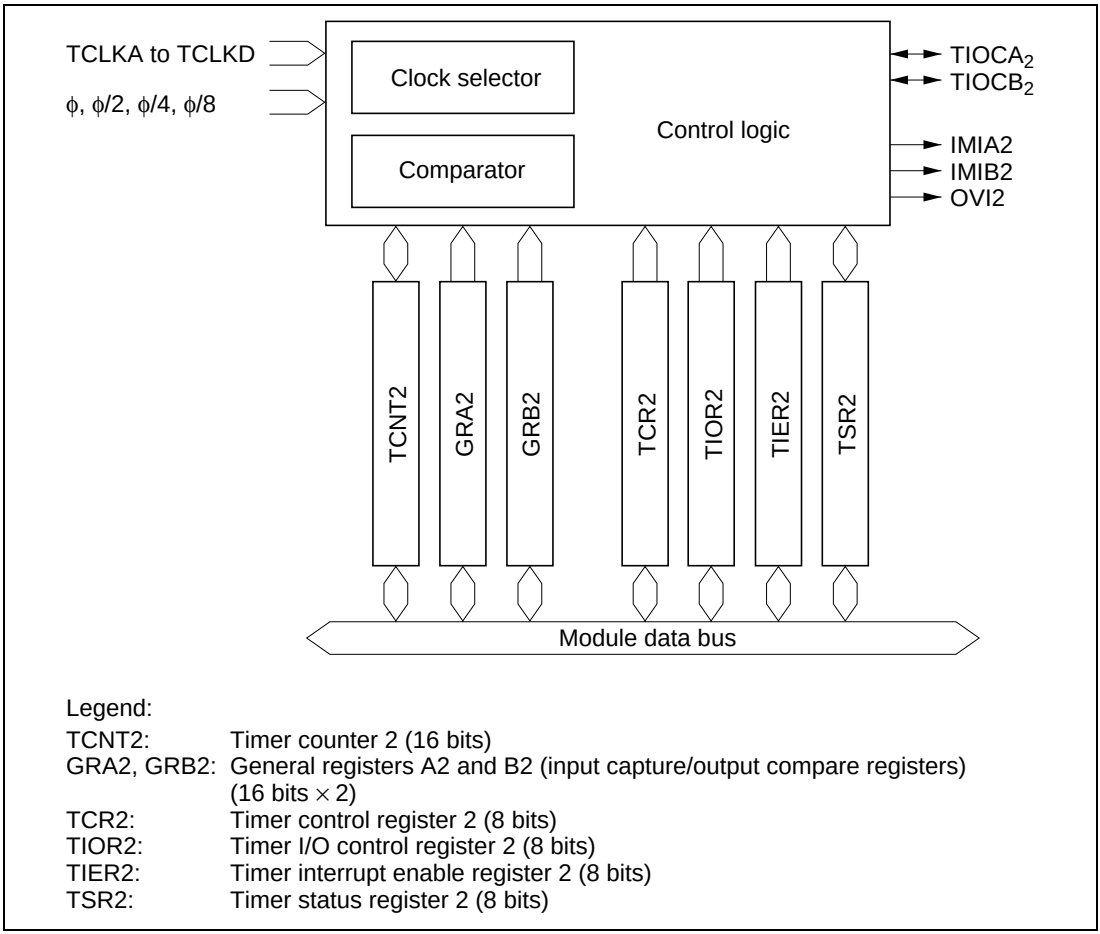
Legend:

TCNT:	Timer counter (16 bits)
GRA, GRB:	General registers A and B (input capture/output compare registers) (16 bits × 2)
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 8.2 Block Diagram of Channels 0 and 1 (for Channel 0)

Block Diagram of Channel 2

Figure 8.3 is a block diagram of channel 2. This is the channel that provides only 0 output and 1 output.



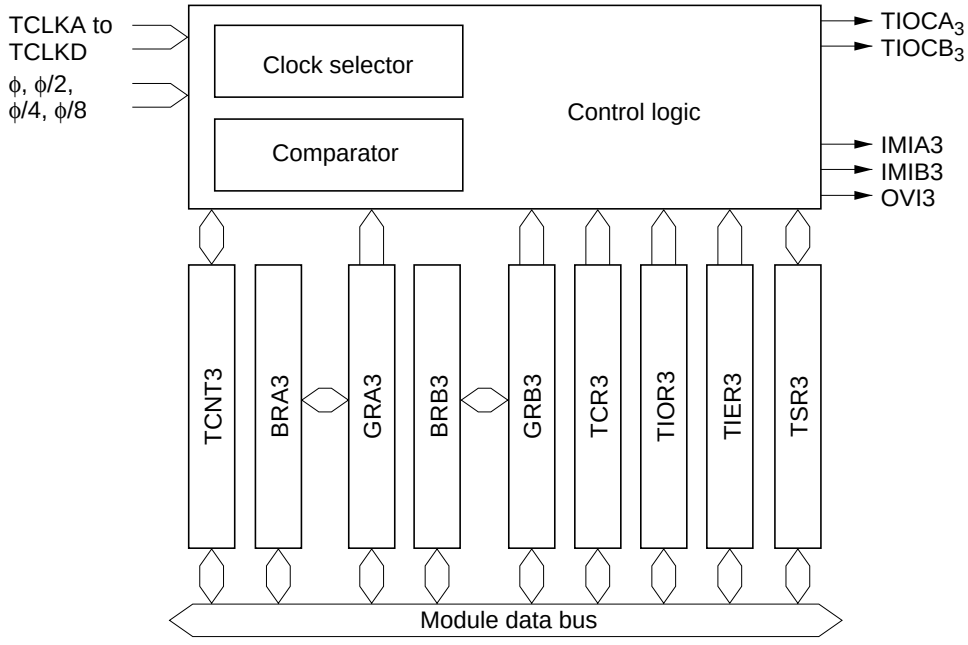
Legend:

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 8.3 Block Diagram of Channel 2

Block Diagrams of Channels 3 and 4

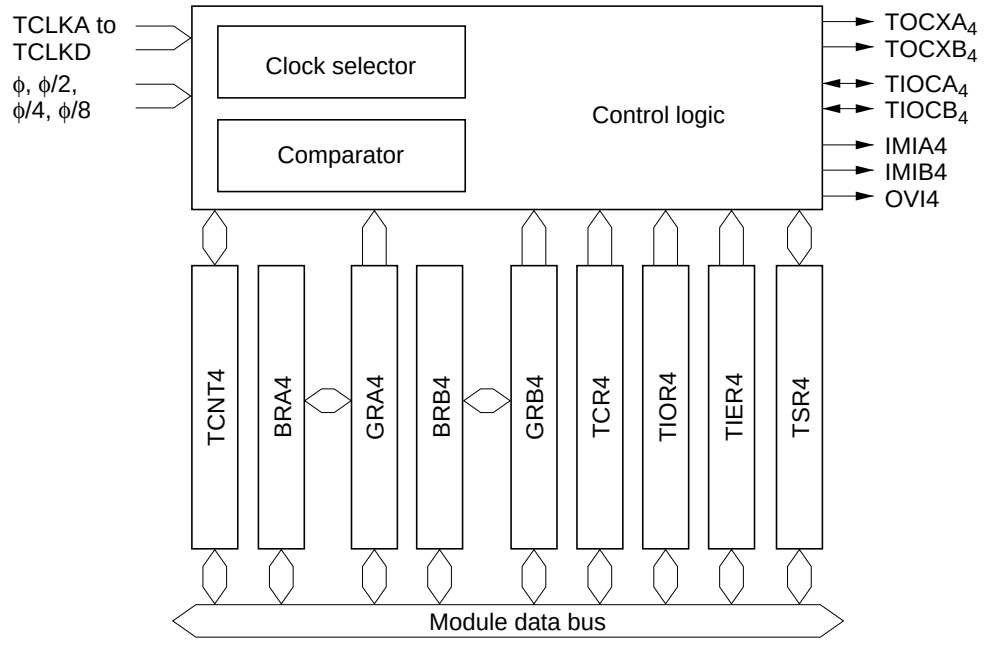
Figure 8.4 is a block diagram of channel 3. Figure 8.5 is a block diagram of channel 4.



Legend:

TCNT3:	Timer counter 3 (16 bits)
GRA3, GRB3:	General registers A3 and B3 (input capture/output compare registers) (16 bits × 2)
BRA3, BRB3:	Buffer registers A3 and B3 (input capture/output compare buffer registers) (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 8.4 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 registers) (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 8.5 Block Diagram of Channel 4

8.1.3 Input/Output Pins

Table 8.2 summarizes the ITU pins.

Table 8.2 ITU Pins

Channel	Name	Abbre-viation	Input/ Output	Function
Common	Clock input A	TCLKA	Input	External clock A input pin (phase-A input pin in phase counting mode)
	Clock input B	TCLKB	Input	External clock B input pin (phase-B input pin in phase counting mode)
	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 capture pin PWM output pin in PWM mode
	Input capture/output compare B0	$TIOCB_0$	Input/ output	GRB0 output compare or input capture pin
1	Input capture/output compare A1	$TIOCA_1$	Input/ output	GRA1 output compare or input capture pin PWM output pin in PWM mode
	Input capture/output compare B1	$TIOCB_1$	Input/ output	GRB1 output compare or input capture pin
2	Input capture/output compare A2	$TIOCA_2$	Input/ output	GRA2 output compare or input capture pin PWM output pin in PWM mode
	Input capture/output compare B2	$TIOCB_2$	Input/ output	GRB2 output compare or input capture pin
3	Input capture/output compare A3	$TIOCA_3$	Input/ output	GRA3 output compare or input capture pin PWM output pin in PWM mode, complementary PWM mode, or reset-synchronized PWM mode
	Input capture/output compare B3	$TIOCB_3$	Input/ output	GRB3 output compare or input capture pin PWM output pin in complementary PWM mode or reset-synchronized PWM mode

Channel	Name	Abbre-viation	Input/ Output	Function
4	Input capture/output compare A4	$TIOCA_4$	Input/ output	GRA4 output compare or input capture pin PWM output pin in PWM mode, complementary PWM mode, or reset-synchronized PWM mode
	Input capture/output compare B4	$TIOCB_4$	Input/ output	GRB4 output compare or input capture pin PWM output pin in complementary PWM mode or reset-synchronized PWM mode
	Output compare XA4	$TOCXA_4$	Output	PWM output pin in complementary PWM mode or reset-synchronized PWM mode
	Output compare XB4	$TOCXB_4$	Output	PWM output pin in complementary PWM mode or reset-synchronized PWM mode

8.1.4 Register Configuration

Table 8.3 summarizes the ITU registers.

Table 8.3 ITU Registers

Channel	Address*[1]	Name	Abbre-viation	R/W	Initial Value
Common	H'FF60	Timer start register	TSTR	R/W	H'E0
	H'FF61	Timer synchro register	TSNC	R/W	H'E0
	H'FF62	Timer mode register	TMDR	R/W	H'80
	H'FF63	Timer function control register	TFCR	R/W	H'C0
	H'FF90	Timer output master enable register	TOER	R/W	H'FF
	H'FF91	Timer output control register	TOCR	R/W	H'FF
0	H'FF64	Timer control register 0	TCR0	R/W	H'80
	H'FF65	Timer I/O control register 0	TIOR0	R/W	H'88
	H'FF66	Timer interrupt enable register 0	TIER0	R/W	H'F8
	H'FF67	Timer status register 0	TSR0	R/(W)*[2]	H'F8
	H'FF68	Timer counter 0 (high)	TCNT0H	R/W	H'00
	H'FF69	Timer counter 0 (low)	TCNT0L	R/W	H'00
	H'FF6A	General register A0 (high)	GRA0H	R/W	H'FF
	H'FF6B	General register A0 (low)	GRA0L	R/W	H'FF
	H'FF6C	General register B0 (high)	GRB0H	R/W	H'FF
	H'FF6D	General register B0 (low)	GRB0L	R/W	H'FF
1	H'FF6E	Timer control register 1	TCR1	R/W	H'80
	H'FF6F	Timer I/O control register 1	TIOR1	R/W	H'88
	H'FF70	Timer interrupt enable register 1	TIER1	R/W	H'F8
	H'FF71	Timer status register 1	TSR1	R/(W)*[2]	H'F8
	H'FF72	Timer counter 1 (high)	TCNT1H	R/W	H'00
	H'FF73	Timer counter 1 (low)	TCNT1L	R/W	H'00
	H'FF74	General register A1 (high)	GRA1H	R/W	H'FF
	H'FF75	General register A1 (low)	GRA1L	R/W	H'FF
	H'FF76	General register B1 (high)	GRB1H	R/W	H'FF
	H'FF77	General register B1 (low)	GRB1L	R/W	H'FF

Channel	Address*[1]	Name	Abbre-viation	R/W	Initial Value
2	H'FF78	Timer control register 2	TCR2	R/W	H'80
	H'FF79	Timer I/O control register 2	TIOR2	R/W	H'88
	H'FF7A	Timer interrupt enable register 2	TIER2	R/W	H'F8
	H'FF7B	Timer status register 2	TSR2	R/(W)*[2]	H'F8
	H'FF7C	Timer counter 2 (high)	TCNT2H	R/W	H'00
	H'FF7D	Timer counter 2 (low)	TCNT2L	R/W	H'00
	H'FF7E	General register A2 (high)	GRA2H	R/W	H'FF
	H'FF7F	General register A2 (low)	GRA2L	R/W	H'FF
	H'FF80	General register B2 (high)	GRB2H	R/W	H'FF
	H'FF81	General register B2 (low)	GRB2L	R/W	H'FF
3	H'FF82	Timer control register 3	TCR3	R/W	H'80
	H'FF83	Timer I/O control register 3	TIOR3	R/W	H'88
	H'FF84	Timer interrupt enable register 3	TIER3	R/W	H'F8
	H'FF85	Timer status register 3	TSR3	R/(W)*[2]	H'F8
	H'FF86	Timer counter 3 (high)	TCNT3H	R/W	H'00
	H'FF87	Timer counter 3 (low)	TCNT3L	R/W	H'00
	H'FF88	General register A3 (high)	GRA3H	R/W	H'FF
	H'FF89	General register A3 (low)	GRA3L	R/W	H'FF
	H'FF8A	General register B3 (high)	GRB3H	R/W	H'FF
	H'FF8B	General register B3 (low)	GRB3L	R/W	H'FF
	H'FF8C	Buffer register A3 (high)	BRA3H	R/W	H'FF
	H'FF8D	Buffer register A3 (low)	BRA3L	R/W	H'FF
	H'FF8E	Buffer register B3 (high)	BRB3H	R/W	H'FF
	H'FF8F	Buffer register B3 (low)	BRB3L	R/W	H'FF

Channel	Address*[1]	Name	Abbre-viation	R/W	Initial Value
4	H'FF92	Timer control register 4	TCR4	R/W	H'80
	H'FF93	Timer I/O control register 4	TIOR4	R/W	H'88
	H'FF94	Timer interrupt enable register 4	TIER4	R/W	H'F8
	H'FF95	Timer status register 4	TSR4	R/(W)*[2]	H'F8
	H'FF96	Timer counter 4 (high)	TCNT4H	R/W	H'00
	H'FF97	Timer counter 4 (low)	TCNT4L	R/W	H'00
	H'FF98	General register A4 (high)	GRA4H	R/W	H'FF
	H'FF99	General register A4 (low)	GRA4L	R/W	H'FF
	H'FF9A	General register B4 (high)	GRB4H	R/W	H'FF
	H'FF9B	General register B4 (low)	GRB4L	R/W	H'FF
	H'FF9C	Buffer register A4 (high)	BRA4H	R/W	H'FF
	H'FF9D	Buffer register A4 (low)	BRA4L	R/W	H'FF
	H'FF9E	Buffer register B4 (high)	BRB4H	R/W	H'FF
	H'FF9F	Buffer register B4 (low)	BRB4L	R/W	H'FF

Notes: 1. The lower 16 bits of the address are indicated.
2. Only 0 can be written, to clear flags.

8.2 Register Descriptions

8.2.1 Timer Start Register (TSTR)

TSTR is an 8-bit readable/writable register that starts and stops the timer counter (TCNT) in channels 0 to 4.

Bit	7	6	5	4	3	2	1	0
	—	—	—	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

Reserved bits

Counter start 4 to 0
These bits start and stop TCNT4 to TCNT0

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

Bits 7 to 5—Reserved: These bits cannot be modified and are always read as 1.

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

Bit4 STR4	Description	
0	TCNT4 is halted	(Initial value)
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	(Initial value)
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	(Initial value)
1	TCNT2 is counting	

Bit 1—Counter Start 1 (STR1): Starts and stops timer counter 1 (TCNT1).

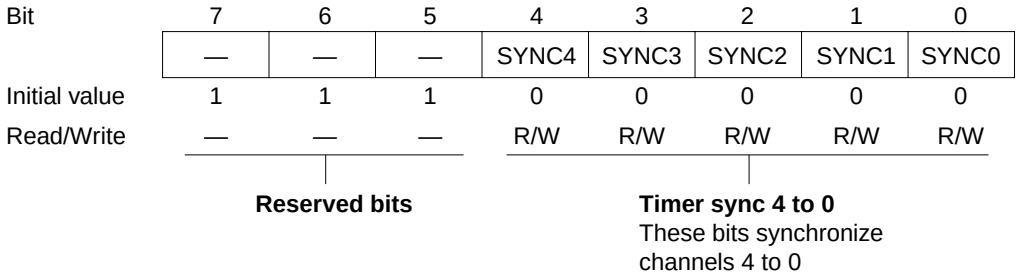
Bit 1 STR1	Description	
0	TCNT1 is halted	(Initial value)
1	TCNT1 is counting	

Bit 0—Counter Start 0 (STR0): Starts and stops timer counter 0 (TCNT0).

Bit 0 STR0	Description	
0	TCNT0 is halted	(Initial value)
1	TCNT0 is counting	

8.2.2 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 corresponding bits to 1.



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

Bits 7 to 5—Reserved: These bits cannot be modified and are 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 (Initial value)
1	Channel 4 operates synchronously TCNT4 can be synchronously preset and cleared

Bit 3—Timer Sync 3 (SYNC3): Selects whether channel 3 operates independently or synchronously.

Bit 3 SYNC3	Description
0	Channel 3's timer counter (TCNT3) operates independently TCNT3 is preset and cleared independently of other channels (Initial value)
1	Channel 3 operates synchronously TCNT3 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 (Initial value)
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.

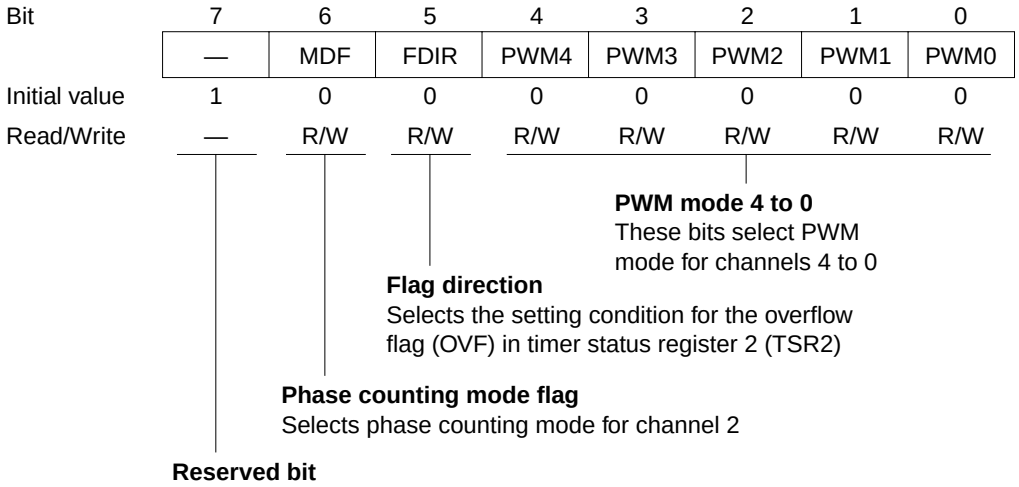
Bit 1 SYNC1	Description
0	Channel 1's timer counter (TCNT1) operates independently TCNT1 is preset and cleared independently of other channels (Initial value)
1	Channel 1 operates synchronously TCNT1 can be synchronously preset and cleared

Bit 0—Timer Sync 0 (SYNC0): Selects whether channel 0 operates independently or synchronously.

Bit 0 SYNC0	Description
0	Channel 0's timer counter (TCNT0) operates independently TCNT0 is preset and cleared independently of other channels (Initial value)
1	Channel 0 operates synchronously TCNT0 can be synchronously preset and cleared

8.2.3 Timer Mode Register (TMDR)

TMDR is an 8-bit readable/writable register that selects PWM mode for channels 0 to 4. It also selects phase counting mode and the overflow flag (OVF) setting conditions for channel 2.



TMDR is initialized to H'80 by a reset and in standby mode.

Bit 7—Reserved: This bit cannot be modified and is always read as 1.

Bit 6—Phase Counting Mode Flag (MDF): Selects whether channel 2 operates normally or in phase counting mode.

Bit 6 MDF	Description	
0	Channel 2 operates normally	(Initial value)
1	Channel 2 operates in phase counting mode	

When MDF is set to 1 to select phase counting mode, timer counter 2 (TCNT2) operates as an up/down-counter and pins TCLKA and TCLKB become counter clock input pins. TCNT2 counts both rising and falling edges of TCLKA and TCLKB, and counts up or down as follows.

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

In phase counting mode channel 2 operates as above regardless of the external clock edges selected by bits CKEG1 and CKEG0 and the clock source selected by bits TPSC2 to TPSC0 in timer control register 2 (TCR2). Phase counting mode takes precedence over these settings.

The counter clearing condition selected by the CCLR1 and CCLR0 bits in TCR2 and the compare match/input capture settings and interrupt functions of timer I/O control register 2 (TIOR2), timer interrupt enable register 2 (TIER2), and timer status register 2 (TSR2) remain effective in phase counting mode.

Bit 5—Flag Direction (FDIR): Designates the setting condition for the overflow flag (OVF) in timer status register 2 (TSR2). The 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	(Initial value)
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 PWM mode.

Bit 4 PWM4	Description	
0	Channel 4 operates normally	(Initial value)
1	Channel 4 operates in PWM mode	

When bit PWM4 is set to 1 to select PWM mode, pin TIOCA4 becomes a PWM output pin. The output goes to 1 at compare match with general register A4 (GRA4), and to 0 at compare match with general register B4 (GRB4).

If complementary PWM mode or reset-synchronized PWM mode is selected by bits CMD1 and CMD0 in the timer function control register (TFCR), the CMD1 and CMD0 setting takes precedence and the PWM4 setting is ignored.

Bit 3—PWM Mode 3 (PWM3): Selects whether channel 3 operates normally or in PWM mode.

Bit 3 PWM3	Description	
0	Channel 3 operates normally	(Initial value)
1	Channel 3 operates in PWM mode	

When bit PWM3 is set to 1 to select PWM mode, pin TIOCA3 becomes a PWM output pin. The output goes to 1 at compare match with general register A3 (GRA3), and to 0 at compare match with general register B3 (GRB3).

If complementary PWM mode or reset-synchronized PWM mode is selected by bits CMD1 and CMD0 in the timer function control register (TFCR), the CMD1 and CMD0 setting takes precedence and the PWM3 setting is ignored.

Bit 2—PWM Mode 2 (PWM2): Selects whether channel 2 operates normally or in PWM mode.

Bit 2 PWM2	Description	
0	Channel 2 operates normally	(Initial value)
1	Channel 2 operates in PWM mode	

When bit PWM2 is set to 1 to select PWM mode, pin TIOCA2 becomes a PWM output pin. The output goes to 1 at compare match with general register A2 (GRA2), and to 0 at compare match with general register B2 (GRB2).

Bit 1—PWM Mode 1 (PWM1): Selects whether channel 1 operates normally or in PWM mode.

Bit 1 PWM1	Description	
0	Channel 1 operates normally	(Initial value)
1	Channel 1 operates in PWM mode	

When bit PWM1 is set to 1 to select PWM mode, pin TIOCA1 becomes a PWM output pin. The output goes to 1 at compare match with general register A1 (GRA1), and to 0 at compare match with general register B1 (GRB1).

Bit 0—PWM Mode 0 (PWM0): Selects whether channel 0 operates normally or in PWM mode.

Bit 0 PWM0	Description	
0	Channel 0 operates normally	(Initial value)
1	Channel 0 operates in PWM mode	

When bit PWM0 is set to 1 to select PWM mode, pin TIOCA0 becomes a PWM output pin. The output goes to 1 at compare match with general register A0 (GRA0), and to 0 at compare match with general register B0 (GRB0).

8.2.4 Timer Function Control Register (TFCR)

TFCR is an 8-bit readable/writable register that selects complementary PWM mode, reset-synchronized PWM mode, and buffering for channels 3 and 4.

Bit	7	6	5	4	3	2	1	0
	—	—	CMD1	CMD0	BFB4	BFA4	BFB3	BFA3
Initial value	1	1	0	0	0	0	0	0
Read/Write	—	—	R/W	R/W	R/W	R/W	R/W	R/W

Reserved bits

Combination mode 1/0
These bits select complementary PWM mode or reset-synchronized PWM mode for channels 3 and 4

Buffer mode B4 and A4
These bits select buffering of general registers (GRB4 and GRA4) by buffer registers (BRB4 and BRA4) in channel 4

Buffer mode B3 and A3
These bits select buffering of general registers (GRB3 and GRA3) by buffer registers (BRB3 and BRA3) in channel 3

TFCR is initialized to H'C0 by a reset and in standby mode.

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

Bits 5 and 4—Combination Mode 1 and 0 (CMD1, CMD0): These bits select whether channels 3 and 4 operate in normal mode, complementary PWM mode, or reset-synchronized PWM mode.

Bit 5 CMD1	Bit 4 CMD0	Description	
0	0	Channels 3 and 4 operate normally	(Initial value)
	1		
1	0	Channels 3 and 4 operate together in complementary PWM mode	
	1	Channels 3 and 4 operate together in reset-synchronized PWM mode	

Before selecting reset-synchronized PWM mode or complementary PWM mode, halt the timer counter or counters that will be used in these modes.

When these bits select complementary PWM mode or reset-synchronized PWM mode, they take precedence over the setting of the PWM mode bits (PWM4 and PWM3) in TMDR. Settings of timer sync bits SYNC4 and SYNC3 in the timer synchro register (TSNC) are valid in complementary PWM mode and reset-synchronized PWM mode, however. When complementary PWM mode is selected, channels 3 and 4 must not be synchronized (do not set bits SYNC3 and SYNC4 both to 1 in TSNC).

Bit 3—Buffer Mode B4 (BFB4): Selects whether GRB4 operates normally in channel 4, or whether GRB4 is buffered by BRB4.

Bit 3 BFB4	Description	
0	GRB4 operates normally	(Initial value)
1	GRB4 is buffered by BRB4	

Bit 2—Buffer Mode A4 (BFA4): Selects whether GRA4 operates normally in channel 4, or whether GRA4 is buffered by BRA4.

Bit 2 BFA4	Description	
0	GRA4 operates normally	(Initial value)
1	GRA4 is buffered by BRA4	

Bit 1—Buffer Mode B3 (BFB3): Selects whether GRB3 operates normally in channel 3, or whether GRB3 is buffered by BRB3.

Bit 1 BFB3	Description	
0	GRB3 operates normally	(Initial value)
1	GRB3 is buffered by BRB3	

Bit 0—Buffer Mode A3 (BFA3): Selects whether GRA3 operates normally in channel 3, or whether GRA3 is buffered by BRA3.

Bit 0 BFA3	Description	
0	GRA3 operates normally	(Initial value)
1	GRA3 is buffered by BRA3	

8.2.5 Timer Output Master Enable Register (TOER)

TOER is an 8-bit readable/writable register that enables or disables output settings for channels 3 and 4.

Bit	7	6	5	4	3	2	1	0
	—	—	EXB4	EXA4	EB3	EB4	EA4	EA3
Initial value	1	1	1	1	1	1	1	1
Read/Write	—	—	R/W	R/W	R/W	R/W	R/W	R/W

Reserved bits

Master enable $TOCXA_4$, $TOCXB_4$
These bits enable or disable output settings for pins $TOCXA_4$ and $TOCXB_4$

Master enable $TIOCA_3$, $TIOCB_3$, $TIOCA_4$, $TIOCB_4$
These bits enable or disable output settings for pins $TIOCA_3$, $TIOCB_3$, $TIOCA_4$, and $TIOCB_4$

TOER is initialized to H'FF by a reset and in standby mode.

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

Bit 5—Master Enable $TOCXB_4$ (EXB4): Enables or disables ITU output at pin $TOCXB_4$.

Bit 5 EXB4	Description
0	$TOCXB_4$ output is disabled regardless of TFCR settings ($TOCXB_4$ operates as a generic input/output pin). If XTGD = 0, EXB4 is cleared to 0 when input capture A occurs in channel 1.
1	$TOCXB_4$ is enabled for output according to TFCR settings (Initial value)

Bit 4—Master Enable $TOCXA_4$ (EXA4): Enables or disables ITU output at pin $TOCXA_4$.

Bit 4 EXA4	Description
0	$TOCXA_4$ output is disabled regardless of TFCR settings ($TOCXA_4$ operates as a generic input/output pin). If XTGD = 0, EXA4 is cleared to 0 when input capture A occurs in channel 1.
1	$TOCXA_4$ is enabled for output according to TFCR settings (Initial value)

Bit 3—Master Enable TIOCB$_3$ (EB3): Enables or disables ITU output at pin TIOCB$_3$.

Bit 3 EB3	Description
0	TIOCB$_3$ output is disabled regardless of TIOR3 and TFCR settings (TIOCB$_3$ operates as a generic input/output pin). If XTGD = 0, EB3 is cleared to 0 when input capture A occurs in channel 1.
1	TIOCB$_3$ is enabled for output according to TIOR3 and TFCR settings (Initial value)

Bit 2—Master Enable TIOCB$_4$ (EB4): Enables or disables ITU output at pin TIOCB$_4$.

Bit 2 EB4	Description
0	TIOCB$_4$ output is disabled regardless of TIOR4 and TFCR settings (TIOCB$_4$ operates as a generic input/output pin). If XTGD = 0, EB4 is cleared to 0 when input capture A occurs in channel 1.
1	TIOCB$_4$ is enabled for output according to TIOR4 and TFCR settings (Initial value)

Bit 1—Master Enable TIOCA$_4$ (EA4): Enables or disables ITU output at pin TIOCA$_4$.

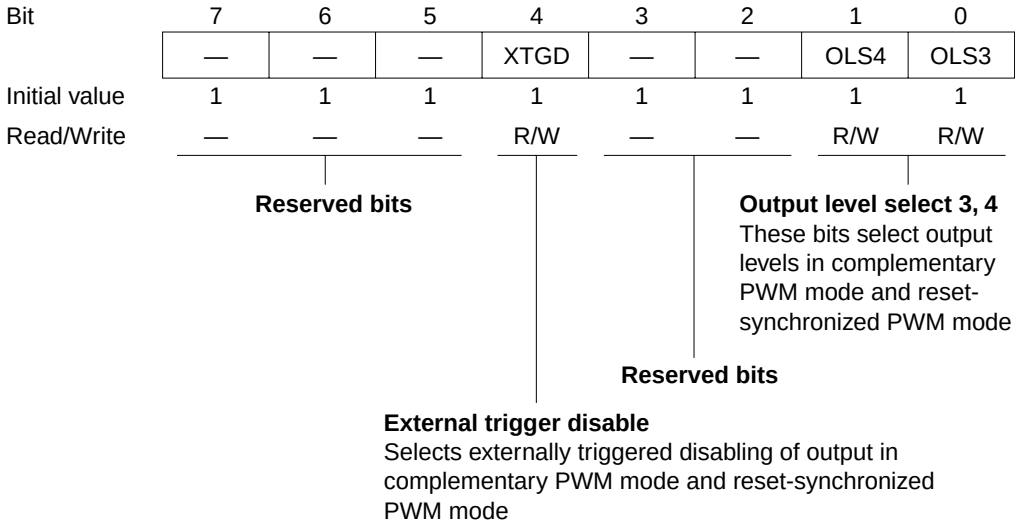
Bit 1 EA4	Description
0	TIOCA$_4$ output is disabled regardless of TIOR4, TMDR, and TFCR settings (TIOCA$_4$ operates as a generic input/output pin). If XTGD = 0, EA4 is cleared to 0 when input capture A occurs in channel 1.
1	TIOCA$_4$ is enabled for output according to TIOR4, TMDR, and TFCR settings (Initial value)

Bit 0—Master Enable TIOCA$_3$ (EA3): Enables or disables ITU output at pin TIOCA$_3$.

Bit 0 EA3	Description
0	TIOCA$_3$ output is disabled regardless of TIOR3, TMDR, and TFCR settings (TIOCA$_3$ operates as a generic input/output pin). If XTGD = 0, EA3 is cleared to 0 when input capture A occurs in channel 1.
1	TIOCA$_3$ is enabled for output according to TIOR3, TMDR, and TFCR settings (Initial value)

8.2.6 Timer Output Control Register (TOCR)

TOCR is an 8-bit readable/writable register that selects externally triggered disabling of output in complementary PWM mode and reset-synchronized PWM mode, and inverts the output levels.



The settings of the XTGD, OLS4, and OLS3 bits are valid only in complementary PWM mode 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: These bits cannot be modified and are always read as 1.

Bit 4—External Trigger Disable (XTGD): Selects externally triggered disabling of ITU output 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 in complementary PWM mode and reset-synchronized PWM mode. When an external trigger occurs, bits 5 to 0 in the timer output master enable register (TOER) are cleared to 0, disabling ITU output.	
1	External triggering is disabled	(Initial value)

Bits 3 and 2—Reserved: These bits cannot be modified and are always read as 1.

Bit 1—Output Level Select 4 (OLS4): Selects output levels in complementary PWM mode and reset-synchronized PWM mode.

Bit 1 OLS4	Description	
0	$TIOCA_3$, $TIOCA_4$, and $TIOCB_4$ pin outputs are inverted	
1	$TIOCA_3$, $TIOCA_4$, and $TIOCB_4$ pin outputs are not inverted	(Initial value)

Bit 0—Output Level Select 3 (OLS3): Selects output levels in complementary PWM mode and reset-synchronized PWM mode.

Bit 0 OLS3	Description	
0	$TIOCB_3$, $TOCXA_4$, and $TOCXB_4$ pin outputs are inverted	
1	$TIOCB_3$, $TOCXA_4$, and $TOCXB_4$ pin outputs are not inverted	(Initial value)

8.2.7 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 Other modes: up-counter
4	TCNT4	

Bit	15	14	13	12	11	10	9	8	7	6	5	4	3	2	1	0
Initial value	0	0	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	R/W	R/W

Each TCNT is a 16-bit readable/writable register that counts pulse inputs from a clock source. The clock source is selected by bits TPSC2 to TPSC0 in the timer control register (TCR).

TCNT0 and TCNT1 are up-counters. TCNT2 is an up/down-counter in phase counting mode and an up-counter in other modes. TCNT3 and TCNT4 are up/down-counters in complementary PWM mode and up-counters in other modes.

TCNT can be cleared to H'0000 by compare match with general register A or B (GRA or GRB) or by input capture to GRA or GRB (counter clearing function) in the same channel.

When TCNT overflows (changes from H'FFFF to H'0000), the overflow flag (OVF) is set to 1 in the timer status register (TSR) of the corresponding channel.

When TCNT underflows (changes from H'0000 to H'FFFF), the overflow flag (OVF) is set to 1 in TSR of the corresponding channel.

The TCNTs are linked to the CPU by an internal 16-bit bus and can be written or read by either word access or byte access.

Each TCNT is initialized to H'0000 by a reset and in standby mode.

8.2.8 General Registers (GRA, GRB)

The general registers are 16-bit registers. The ITU has 10 general registers, two in each channel.

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 by buffer registers BRA and BRB
4	GRA4, GRB4	

Bit	15	14	13	12	11	10	9	8	7	6	5	4	3	2	1	0
Initial value	1	1	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	R/W	R/W

A general register is a 16-bit readable/writable register that can function as either an output compare register or an input capture register. The function is selected by settings in the timer I/O control register (TIOR).

When a general register is used as an output compare register, its value is constantly compared with the TCNT value. When the two values match (compare match), the IMFA or IMFB flag is set to 1 in the timer status register (TSR). Compare match output can be selected in TIOR.

When a general register is used as an input capture register, rising edges, falling edges, or both edges of an external input capture signal are detected and the current TCNT value is stored in the general register. The corresponding IMFA or IMFB flag in TSR is set to 1 at the same time. The valid edge or edges of the input capture signal are selected in TIOR.

TIOR settings are ignored in PWM mode, complementary PWM mode, and reset-synchronized PWM mode.

General registers are linked to the CPU by an internal 16-bit bus and can be written or read by either word access or byte access.

General registers are initialized to the output compare function (with no output signal) by a reset and in standby mode. The initial value is H'FFFF.

8.2.9 Buffer Registers (BRA, BRB)

The buffer registers are 16-bit registers. The ITU has four buffer registers, two each in channels 3 and 4.

Channel	Abbreviation	Function
3	BRA3, BRB3	Used for buffering
4	BRA4, BRB4	• When the corresponding GRA or GRB functions as an output compare register, BRA or BRB can function as an output compare buffer register: the BRA or BRB value is automatically transferred to GRA or GRB at compare match • When the corresponding GRA or GRB functions as an input capture register, BRA or BRB can function as an input capture buffer register: the GRA or GRB value is automatically transferred to BRA or BRB at input capture

Bit	15	14	13	12	11	10	9	8	7	6	5	4	3	2	1	0
Initial value	1	1	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	R/W	R/W

A buffer register is a 16-bit readable/writable register that is used when buffering is selected. Buffering can be selected independently by bits BFB4, BFA4, BFB3, and BFA3 in TFCR.

The buffer register and general register operate as a pair. When the general register functions as an output compare register, the buffer register functions as an output compare buffer register. When the general register functions as an input capture register, the buffer register functions as an input capture buffer register.

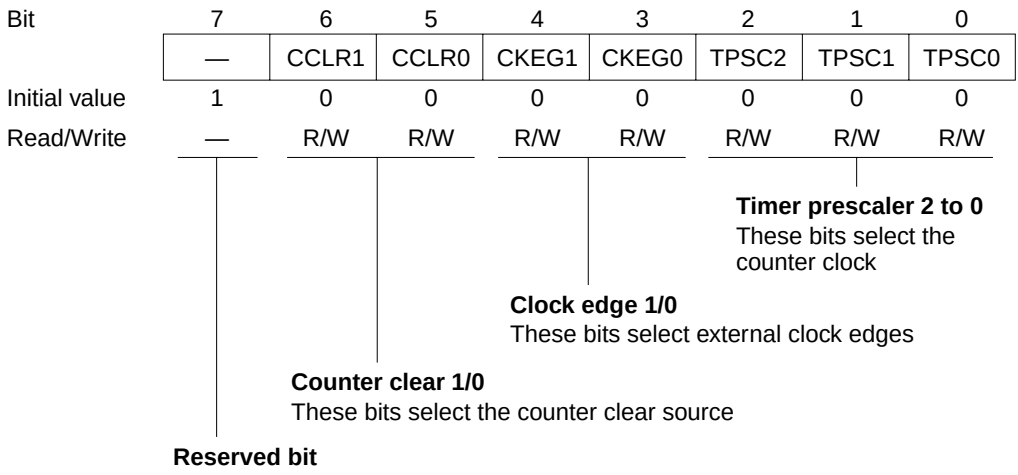
The buffer registers are linked to the CPU by an internal 16-bit bus and can be written or read by either word or byte access.

Buffer registers are initialized to H'FFFF by a reset and in standby mode.

8.2.10 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 channels are functionally identical. When phase counting mode is selected in channel 2, the settings of bits CKEG1 and CKEG0 and TPSC2 to TPSC0 in TCR2 are ignored.
1	TCR1	
2	TCR2	
3	TCR3	
4	TCR4	



Each TCR is an 8-bit readable/writable register that selects the timer counter clock source, selects 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: This bit cannot be modified and is always read as 1.

Bits 6 and 5—Counter Clear 1/0 (CCLR1, CCLR0): These bits select how TCNT is cleared.

Bit 6 CCLR1	Bit 5 CCLR0	Description	
0	0	TCNT is not cleared	(Initial value)
	1	TCNT is cleared by GRA compare match or input capture*[1]	
1	0	TCNT is cleared by GRB compare match or input capture*[1]	
	1	Synchronous clear: TCNT is cleared in synchronization with other synchronized timers*[2]	

Notes: 1. TCNT is cleared by compare match when the general register functions as an output compare match register, and by input capture when the general register functions as an input capture register.
2. Selected in the timer synchro register (TSNC).

Bits 4 and 3—Clock Edge 1/0 (CKEG1, CKEG0): These bits select external clock input edges when an external clock source is used.

Bit 4 CKEG1	Bit 3 CKEG0	Description	
0	0	Count rising edges	(Initial value)
	1	Count falling edges	
1	—	Count both edges	

When channel 2 is set to phase counting mode, bits CKEG1 and CKEG0 in TCR2 are ignored. Phase counting takes precedence.

Bits 2 to 0—Timer Prescaler 2 to 0 (TPSC2 to TPSC0): These bits select the counter clock source.

Bit 2 TPSC2	Bit 1 TPSC1	Bit 0 TPSC0	Function	
0	0	0	Internal clock: ϕ	(Initial value)
		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	

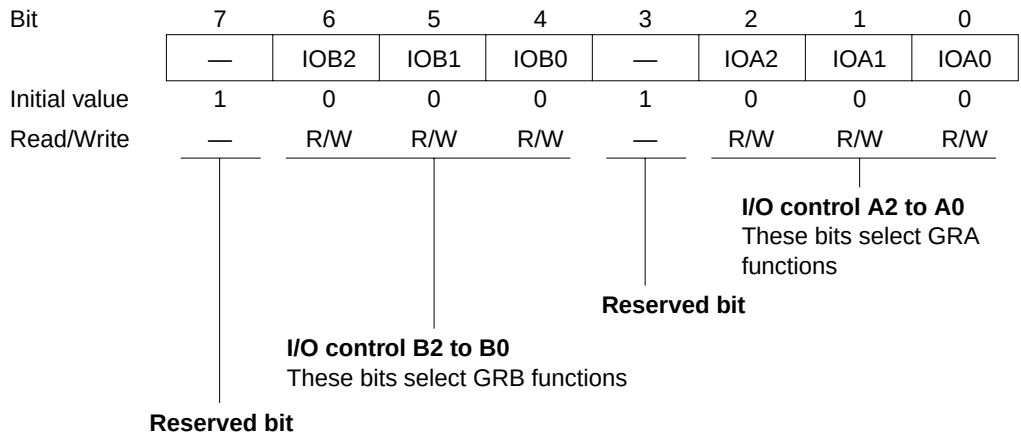
When bit TPSC2 is cleared to 0 an internal clock source is selected, and the timer counts only falling edges. When bit TPSC2 is set to 1 an external clock source is selected, and the timer counts the edge or edges selected by bits CKEG1 and CKEG0.

When channel 2 is set to phase counting mode (MDF = 1 in TMDR), the settings of bits TPSC2 to TPSC0 in TCR2 are ignored. Phase counting takes precedence.

8.2.11 Timer I/O Control Register (TIOR)

TIOR is an 8-bit register. The ITU has five TIORs, one in each channel.

Channel	Abbreviation	Function
0	TIOR0	TIOR controls the general registers. Some functions differ in PWM mode. TIOR3 and TIOR4 settings are ignored when complementary PWM mode or reset-synchronized PWM mode is selected in channels 3 and 4.
1	TIOR1	
2	TIOR2	
3	TIOR3	
4	TIOR4	



Each TIOR is an 8-bit readable/writable register that selects the output compare or input capture function for GRA and GRB, and specifies the functions of the TIOCA and TIOCB pins. If the output compare function is selected, TIOR also selects the type of output. If input capture is 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: This bit cannot be modified and is always read as 1.

Bits 6 to 4—I/O Control B2 to B0 (IOB2 to IOB0): These bits select the GRB function.

Bit 6 IOB2	Bit 5 IOB1	Bit 4 IOB0	Function	
0	0	0	GRB is an output compare register	No output at compare match (Initial value)
		1		0 output at GRB compare match*[1]
	1	0		1 output at GRB compare match*[1]
		1		Output toggles at GRB compare match (1 output in channel 2)*[1]*[2]
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		

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 1 output instead.

Bit 3—Reserved: This bit cannot be modified and is always read as 1.

Bits 2 to 0—I/O Control A2 to A0 (IOA2 to IOA0): These bits select the GRA function.

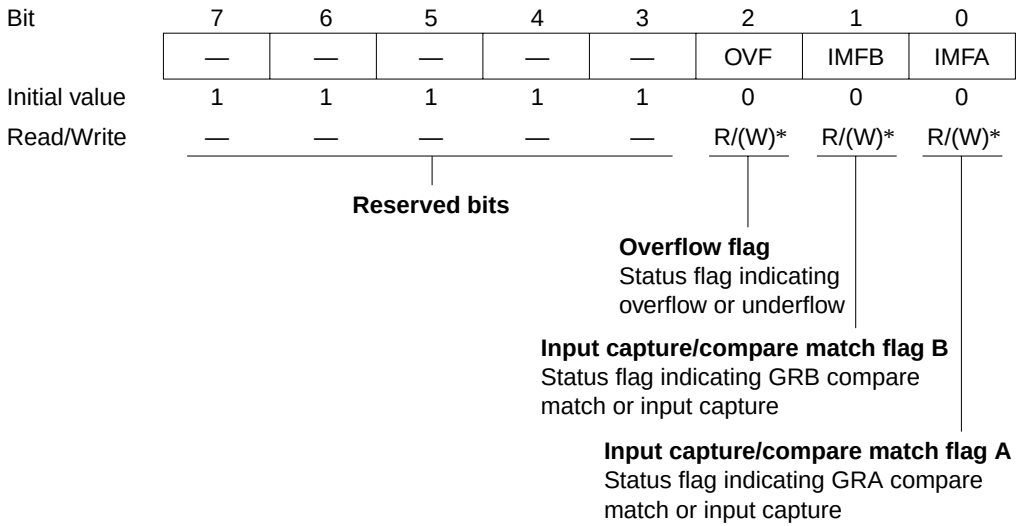
Bit 2 IOA2	Bit 1 IOA1	Bit 0 IOA0	Function	
0	0	0	GRA is an output compare register	No output at compare match (Initial value)
		1		0 output at GRA compare match*[1]
	1	0		1 output at GRA compare match*[1]
		1		Output toggles at GRA compare match (1 output in channel 2)*[1]*[2]
1	0	0	GRA is an input capture register	GRA captures rising edge of input
		1		GRA captures falling edge of input
	1	0		GRA captures both edges of input
		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 1 output instead.

8.2.12 Timer Status Register (TSR)

TSR is an 8-bit register. The ITU has five TSRs, one in each channel.

Channel	Abbreviation	Function
0	TSR0	Indicates input capture, compare match, and overflow status
1	TSR1	
2	TSR2	
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 overflow or underflow and GRA or GRB compare match or input capture. These flags are interrupt sources and generate CPU interrupts if enabled by corresponding bits in the timer interrupt enable register (TIER).

TSR is initialized to H'F8 by a reset and in standby mode.

Bits 7 to 3—Reserved: These bits cannot be modified and are always read as 1.

Bit 2—Overflow Flag (OVF): This status flag indicates TCNT overflow or underflow.

Bit 2 OVF	Description	
0	[Clearing condition]	(Initial value)
	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*	

Notes: * TCNT underflow occurs when TCNT operates as an up/down-counter. Underflow 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 and CMD0 = 0 in TFCR)

Bit 1—Input Capture/Compare Match Flag B (IMFB): This status flag indicates GRB compare match or input capture events.

Bit 1 IMFB	Description	
0	[Clearing condition]	
	Read IMFB when IMFB = 1, then write 0 in IMFB	(Initial value)
1	[Setting conditions]	
	• TCNT = GRB when GRB functions as a compare match register.	
	• TCNT value is transferred to GRB by an input capture signal, when GRB functions as an input capture register.	

Bit 0—Input Capture/Compare Match Flag A (IMFA): This status flag indicates GRA compare match or input capture events.

Bit 0 IMFA	Description	
0	[Clearing condition]	
	Read IMFA when IMFA = 1, then write 0 in IMFA.	(Initial value)
1	[Setting conditions]	
	• TCNT = GRA when GRA functions as a compare match register.	
	• TCNT value is transferred to GRA by an input capture signal, when GRA functions as an input capture register.	

8.2.13 Timer Interrupt Enable Register (TIER)

TIER is an 8-bit register. The ITU has five TIERs, one in each channel.

Channel	Abbreviation	Function
0	TIER0	Enables or disables interrupt requests.
1	TIER1	
2	TIER2	
3	TIER3	
4	TIER4	

Bit	7	6	5	4	3	2	1	0
	—	—	—	—	—	OVIE	IMIEB	IMIEA
Initial value	1	1	1	1	1	0	0	0
Read/Write	—	—	—	—	—	R/W	R/W	R/W

Reserved bits

Overflow interrupt enable
Enables or disables OVF interrupts

Input capture/compare match interrupt enable B
Enables or disables IMFB interrupts

Input capture/compare match interrupt enable A
Enables or disables IMFA interrupts

Each TIER is an 8-bit readable/writable register that enables and disables overflow interrupt 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: These bits cannot be modified and are always read as 1.

Bit 2—Overflow Interrupt Enable (OVIE): Enables or disables the interrupt requested by the overflow flag (OVF) in TSR when OVF is set to 1.

Bit 2 OVIE	Description	
0	OVI interrupt requested by OVF is disabled	(Initial value)
1	OVI interrupt requested by OVF is enabled	

Bit 1—Input Capture/Compare Match Interrupt Enable B (IMIEB): Enables or disables the 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	(Initial value)
1	IMIB interrupt requested by IMFB is enabled	

Bit 0—Input Capture/Compare Match Interrupt Enable A (IMIEA): Enables or disables the 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	(Initial value)
1	IMIA interrupt requested by IMFA is enabled	

8.3 CPU Interface

8.3.1 16-Bit Accessible Registers

The timer counters (TCNTs), general registers A and B (GRAs and GRBs), and buffer registers A and B (BRAs and BRBs) are 16-bit registers, and are linked to the CPU by an internal 16-bit data bus. These registers can be written or read a word at a time, or a byte at a time.

Figures 8.6 and 8.7 show examples of word access to a timer counter (TCNT). Figures 8.8, 8.9, 8.10, and 8.11 show examples of byte access to TCNTH and TCNTL.

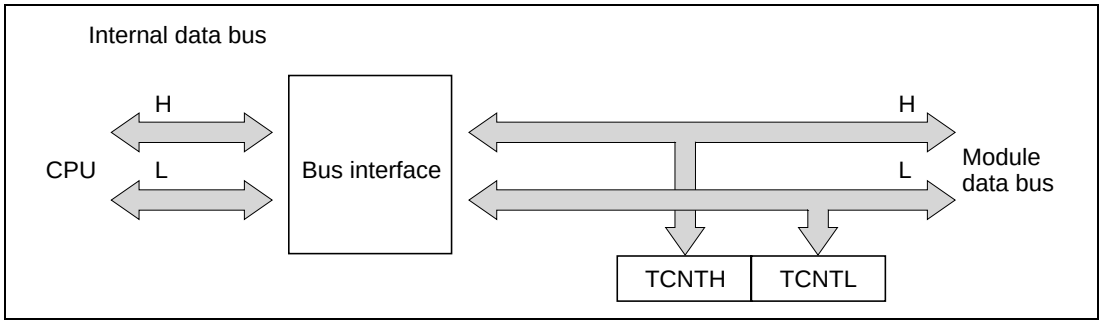


Figure 8.6 Access to Timer Counter (CPU Writes to TCNT, Word)

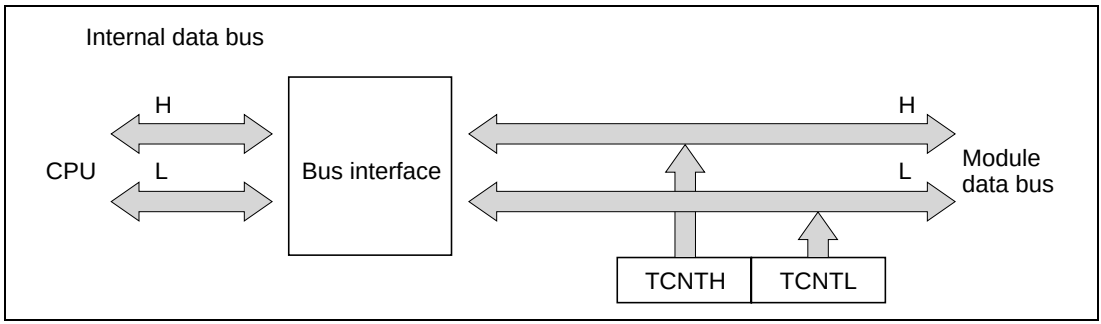


Figure 8.7 Access to Timer Counter (CPU Reads TCNT, Word)

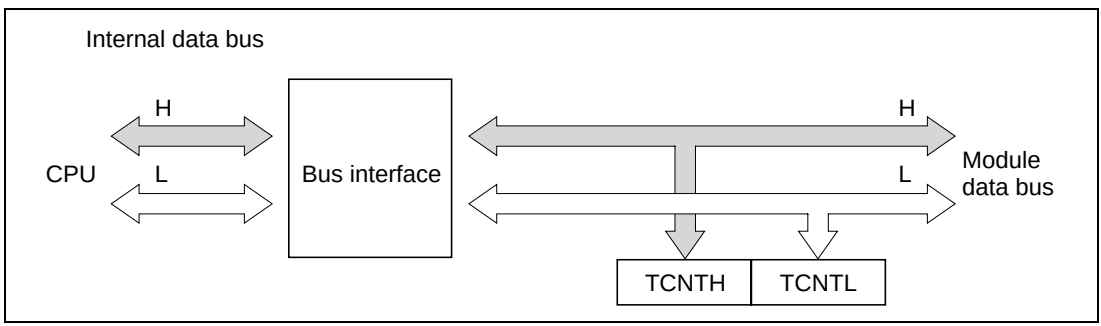


Figure 8.8 Access to Timer Counter (CPU Writes to TCNT, Upper Byte)

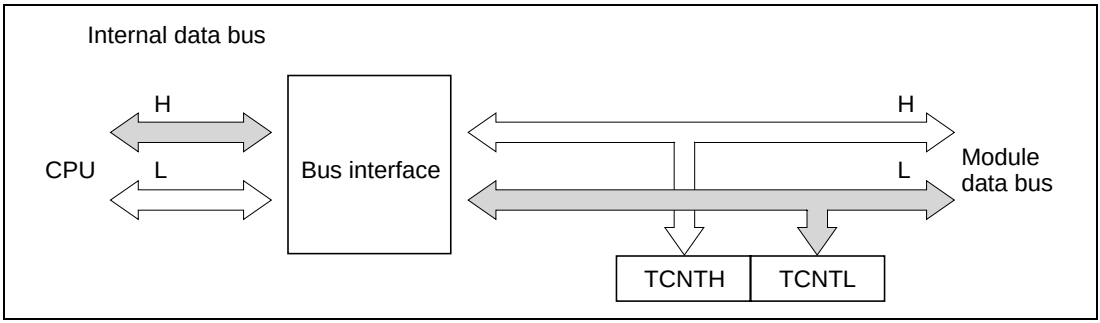


Figure 8.9 Access to Timer Counter (CPU Writes to TCNT, Lower Byte)

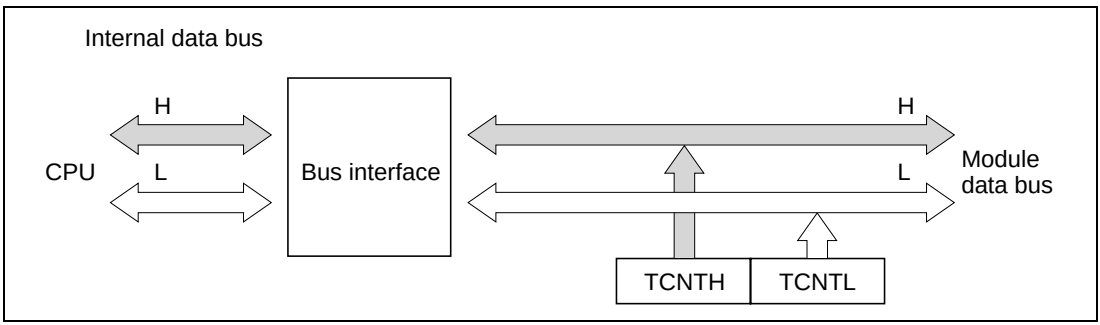


Figure 8.10 Access to Timer Counter (CPU Reads TCNT, Upper Byte)

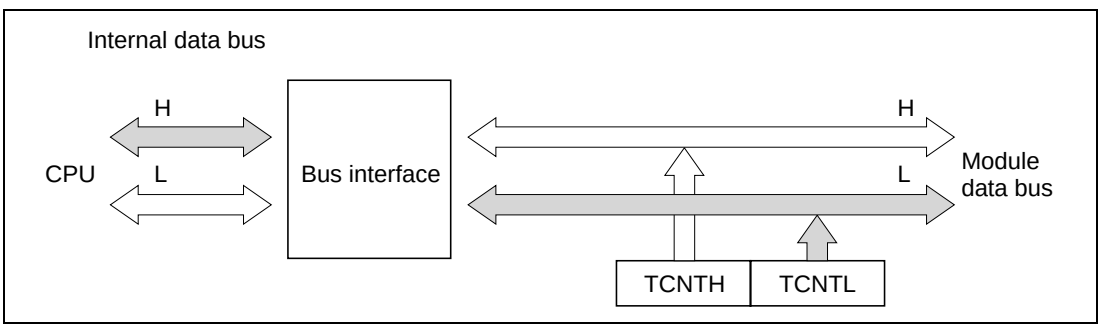


Figure 8.11 Access to Timer Counter (CPU Reads TCNT, Lower Byte)

8.3.2 8-Bit Accessible Registers

The registers other than the timer counters, general registers, and buffer registers are 8-bit registers. These registers are linked to the CPU by an internal 8-bit data bus.

Figures 8.12 and 8.13 show examples of byte read and write access to a TCR.

If a word-size data transfer instruction is executed, two byte transfers are performed.

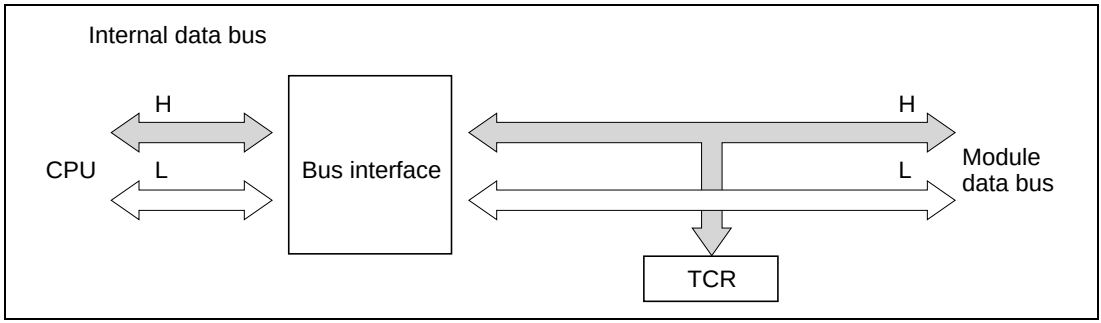


Figure 8.12 TCR Access (CPU Writes to TCR)

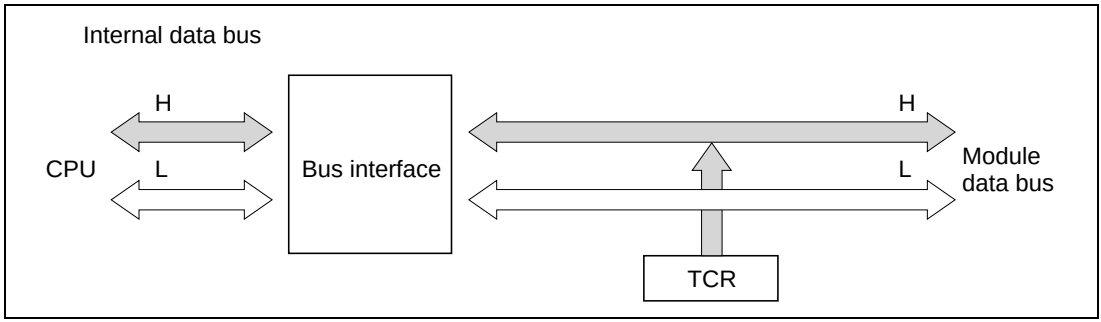


Figure 8.13 TCR Access (CPU Reads TCR)

8.4 Operation

8.4.1 Overview

A summary of operations in the various modes is given below.

Normal Operation: Each channel has a timer counter and general registers. The timer counter counts up, and can operate as a free-running counter, periodic counter, or external event counter. General registers A 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 the timer counter in any one of these channels is simultaneously written to the timer counters in the other channels as well. The timer counters can also be cleared synchronously if so designated by the CCLR1 and CCLR0 bits in the TCRs.

PWM Mode: A PWM waveform is output from the TIOCA pin. The output goes to 1 at compare match A and to 0 at compare match B. The duty cycle can be varied from 0% to 100% depending on the settings of GRA and GRB. When a channel is set to PWM mode, its GRA and GRB automatically become output compare registers.

Reset-Synchronized PWM Mode: Channels 3 and 4 are paired for three-phase PWM output with complementary waveforms. (The three phases are related by having a common transition point.) When reset-synchronized PWM mode is selected GRA3, GRB3, GRA4, and GRB4 automatically function as output compare registers, $TIOCA_3$, $TIOCB_3$, $TIOCA_4$, $TOCXA_4$, $TIOCB_4$, and $TOCXB_4$ function as PWM output pins, and $TCNT_3$ operates as an up-counter. TCNT4 operates independently, and is not compared with GRA4 or GRB4.

Complementary PWM Mode: Channels 3 and 4 are paired for three-phase PWM output with non-overlapping complementary waveforms. When complementary PWM mode is selected GRA3, GRB3, GRA4, and GRB4 automatically function as output compare registers, and $TIOCA_3$, $TIOCB_3$, $TIOCA_4$, $TOCXA_4$, $TIOCB_4$, and $TOCXB_4$ function as PWM output pins. TCNT3 and TCNT4 operate as up/down-counters.

Phase Counting Mode: The phase relationship between two clock signals input at TCLKA and TCLKB is detected and TCNT2 counts up or down accordingly. When phase counting mode is selected TCLKA and 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 register.
- If the general register is an input capture register

  When input capture occurs the TCNT value is transferred to the general register, and the 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 TCNT4 change counting direction.
- Reset-synchronized PWM mode

  The buffer register value is transferred to the general register at GRA3 compare match.

8.4.2 Basic Functions

Counter Operation

When one of bits STR0 to STR4 is set to 1 in the timer start register (TSTR), the timer counter (TCNT) in the corresponding channel starts counting. The counting can be free-running or periodic.

Sample setup procedure for counter: Figure 8.14 shows a sample procedure for setting up a counter.

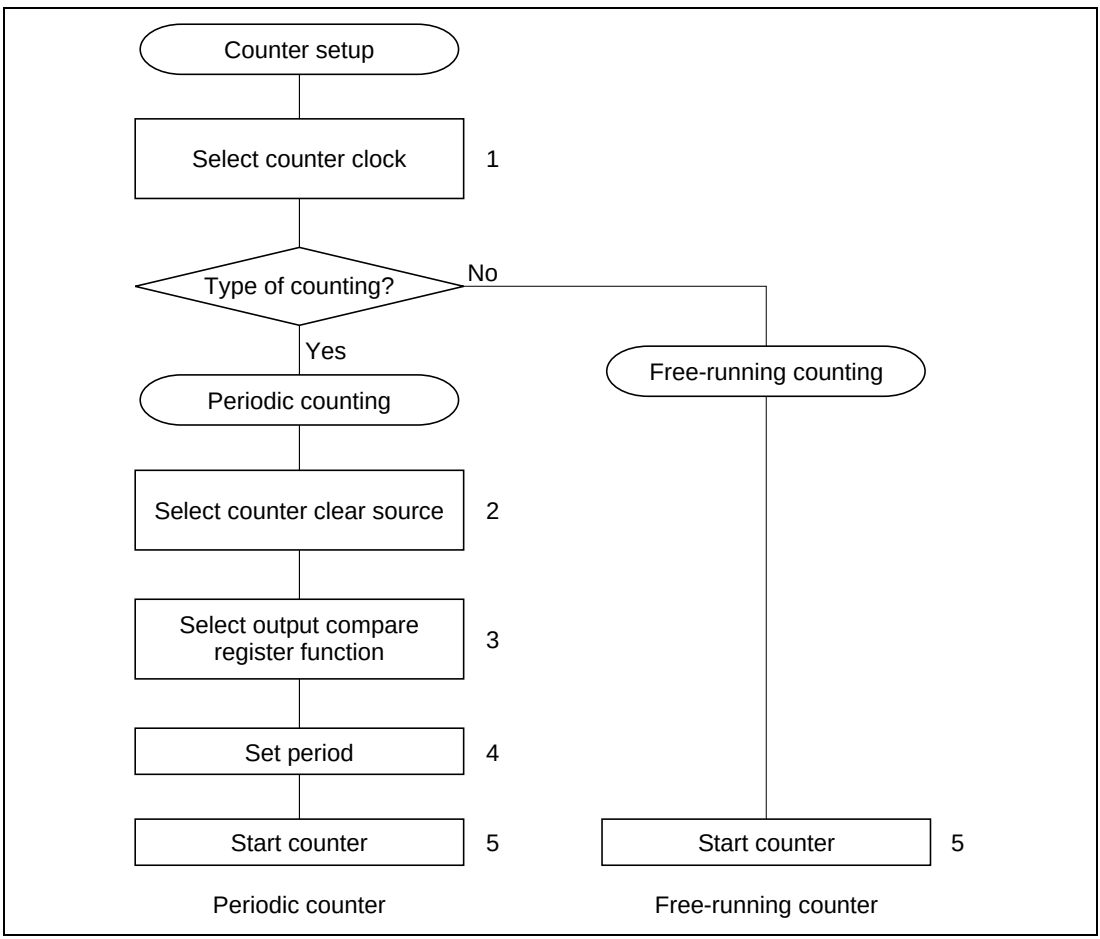


Figure 8.14 Counter Setup Procedure (Example)

1. Set bits TPSC2 to TPSC0 in TCR to select the counter clock source. If an external clock source is selected, set bits CKEG1 and CKEG0 in TCR to select the desired edge(s) of the external clock signal.
2. For periodic counting, set CCLR1 and CCLR0 in TCR to have TCNT cleared at GRA compare match or GRB compare match.
3. Set TIOR to select the output compare function of GRA or GRB, whichever was selected in step 2.
4. Write the count period in GRA or GRB, whichever was selected in step 2.
5. Set the STR bit to 1 in TSTR to start the timer counter.

Free-running and periodic counter operation: A reset leaves the counters (TCNTs) in ITU channels 0 to 4 all set as free-running counters. A free-running counter starts counting up when the corresponding bit in TSTR is set to 1. When the count overflows from H'FFFF to H'0000, the overflow flag (OVF) is set to 1 in the timer status register (TSR). If the corresponding OVIE bit is set to 1 in the timer interrupt enable register, a CPU interrupt is requested. After the overflow, the counter continues counting up from H'0000. Figure 8.15 illustrates free-running counting.

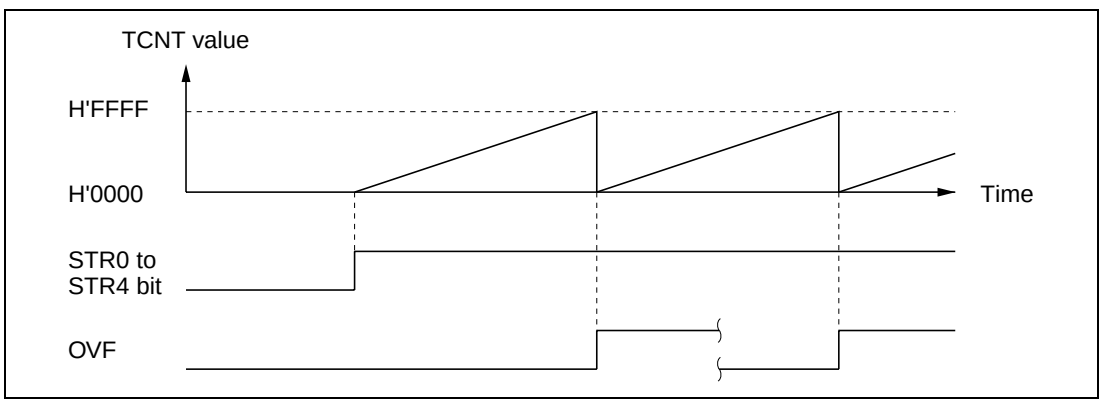


Figure 8.15 Free-Running Counter Operation

When a channel is set to have its counter cleared by compare match, in that channel TCNT operates as a periodic counter. Select the output compare function of GRA or GRB, set bit CCLR1 or CCLR0 in the timer control register (TCR) to have the counter cleared by compare match, and set the count period in GRA or GRB. After these settings, the counter starts counting up as a periodic counter when the corresponding bit is set to 1 in TSTR. When the count matches GRA or GRB, the IMFA or IMFB flag is set to 1 in TSR and the counter is cleared to H'0000. If the corresponding IMIEA or IMIEB bit is set to 1 in TIER, a CPU interrupt is requested at this time. After the compare match, TCNT continues counting up from H'0000. Figure 8.16 illustrates periodic counting.

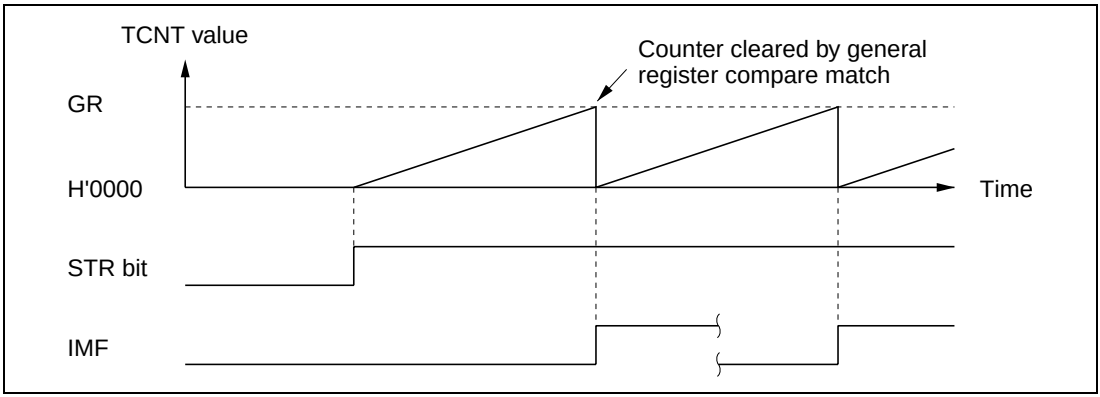


Figure 8.16 Periodic Counter Operation

Count timing:

- Internal clock source

  Bits TPSC2 to TPSC0 in TCR select the system clock (ϕ) or one of three internal clock sources obtained by prescaling the system clock (ϕ/2, ϕ/4, ϕ/8).
  Figure 8.17 shows the timing.

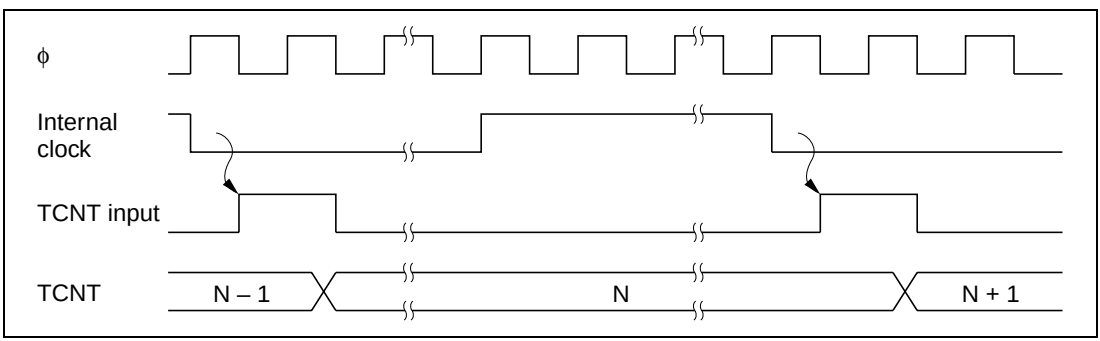


Figure 8.17 Count Timing for Internal Clock Sources

- External clock source

  Bits TPSC2 to TPSC0 in TCR select an external clock input pin (TCLKA to TCLKD), and its valid edge or edges are selected by bits CKEG1 and CKEG0. The rising edge, falling edge, or both edges can be selected.

  The pulse width of the external clock signal must be at least 1.5 system clocks when a single edge is selected, and at least 2.5 system clocks when both edges are selected. Shorter pulses will not be counted correctly.

  Figure 8.18 shows the timing when both edges are detected.

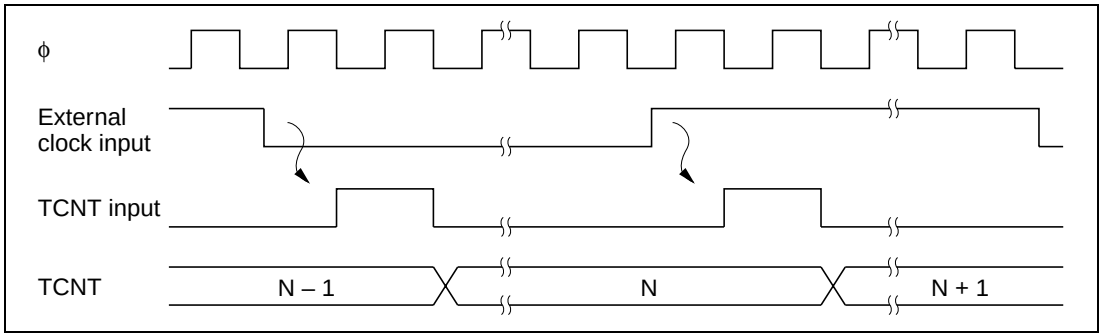


Figure 8.18 Count Timing for External Clock Sources (when Both Edges are Detected)

Waveform Output by Compare Match

In ITU channels 0, 1, 3, and 4, compare match A or B can cause the output at the TIOCA or TIOCB pin to go to 0, go to 1, or toggle. In channel 2 the output can only go to 0 or go to 1.

Sample setup procedure for waveform output by compare match: Figure 8.19 shows a sample procedure for setting up waveform output by compare match.

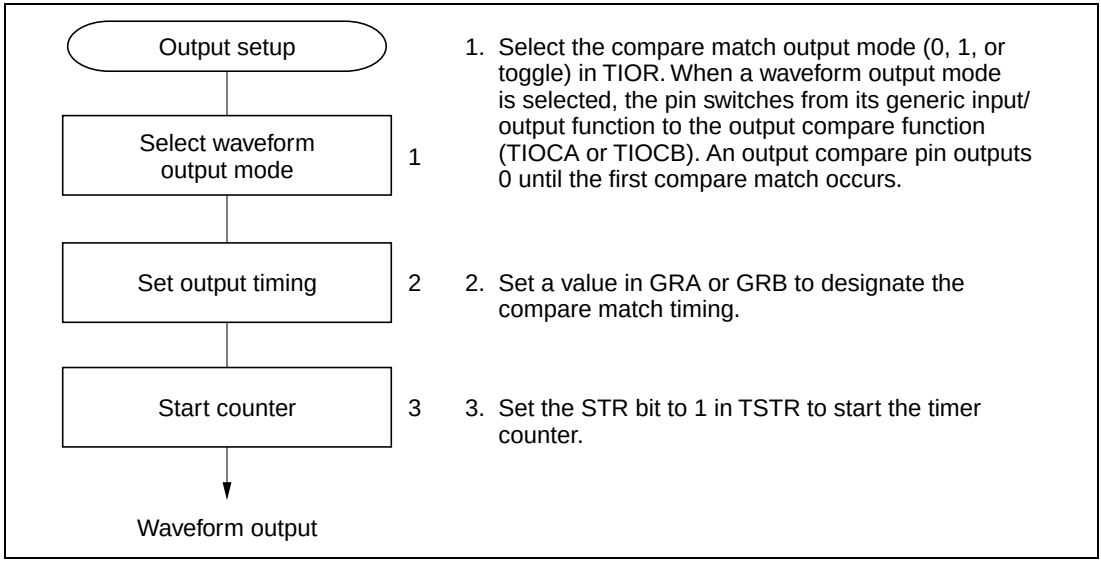


Figure 8.19 Setup Procedure for Waveform Output by Compare Match (Example)

Examples of waveform output: Figure 8.20 shows examples of 0 and 1 output. TCNT operates as a free-running counter, 0 output is selected for compare match A, and 1 output is selected for compare match B. When the pin is already at the selected output level, the pin level does not change.

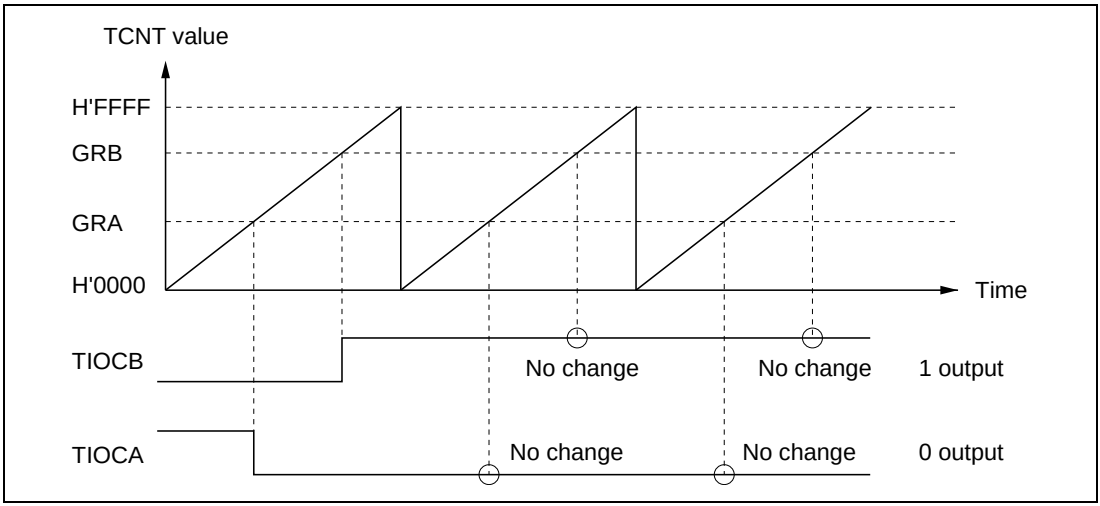


Figure 8.20 0 and 1 Output (Examples)

Figure 8.21 shows examples of toggle output. TCNT operates as a periodic counter, cleared by compare match B. Toggle output is selected for both compare match A and B.

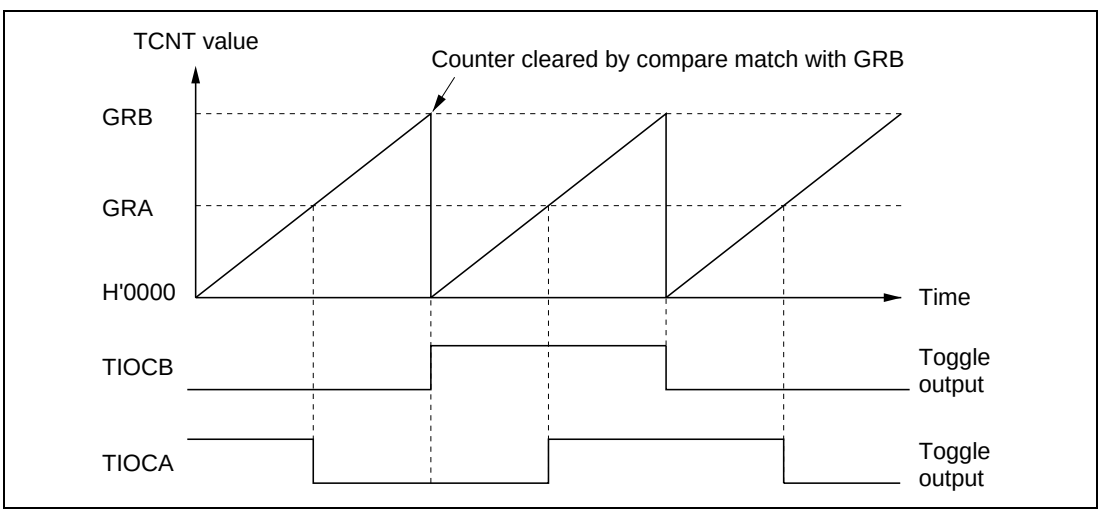


Figure 8.21 Toggle Output (Example)

Output compare timing: The compare match signal is generated in the last state in which TCNT and the general register match (when TCNT changes from the matching value to the next value). When the compare match signal is generated, the output value selected in TIOR is output at the output compare pin (TIOCA or TIOCB). When TCNT matches a general register, the compare match signal is not generated until the next counter clock pulse.

Figure 8.22 shows the output compare timing.

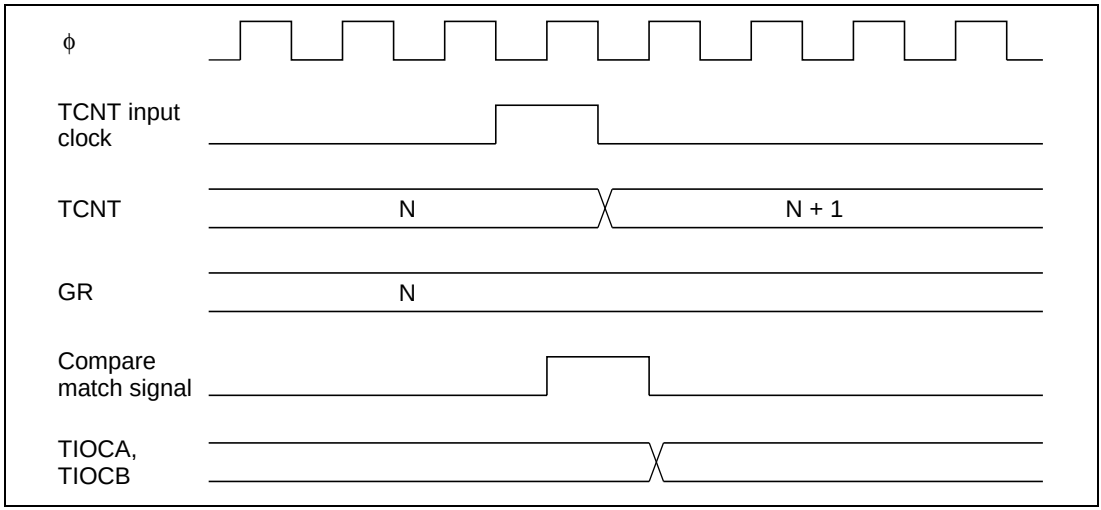


Figure 8.22 Output Compare Timing

Input Capture Function

The TCNT value can be captured into a general register when a transition occurs at an input capture/output compare pin (TIOCA or TIOCB). Capture can take place on the rising edge, falling edge, or both edges. The input capture function can be used to measure pulse width or period.

Sample setup procedure for input capture: Figure 8.23 shows a sample procedure for setting up input capture.

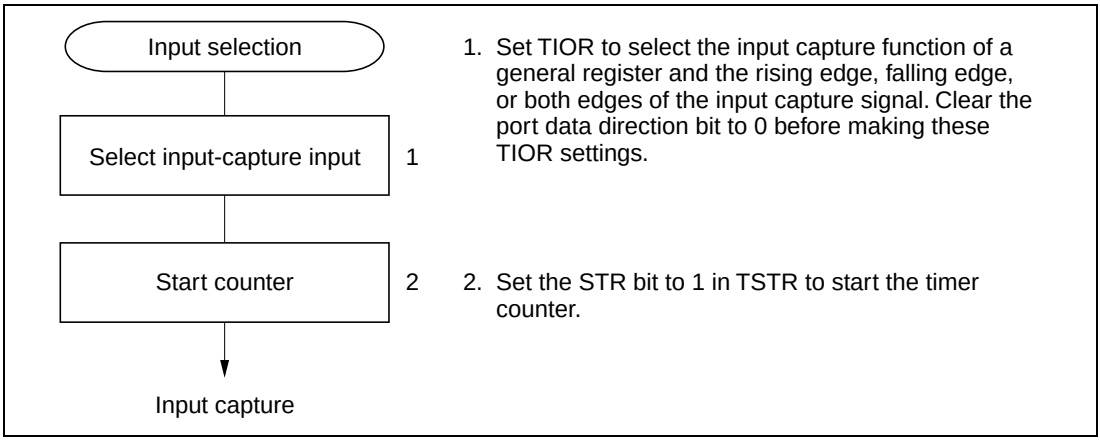


Figure 8.23 Setup Procedure for Input Capture (Example)

Examples of input capture: Figure 8.24 illustrates input capture when the falling edge of TIOCB and both edges of TIOCA are selected as capture edges. TCNT is cleared by input capture into GRB.

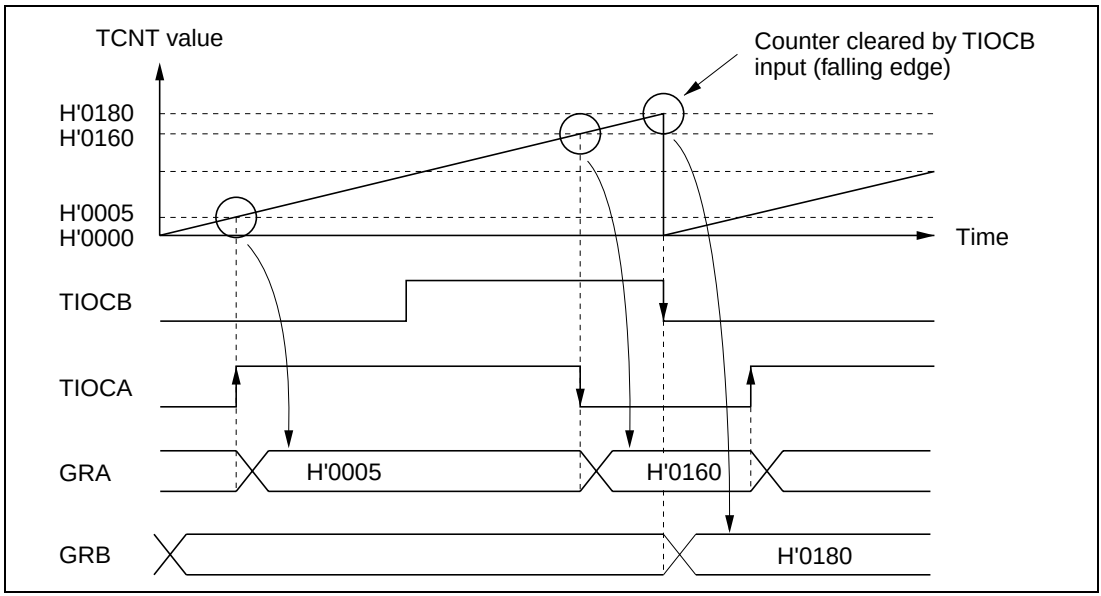


Figure 8.24 Input Capture (Example)

Input capture signal timing: Input capture on the rising edge, falling edge, or both edges can be selected by settings in TIOR. Figure 8.25 shows the timing when the rising edge is selected. The pulse width of the input capture signal must be at least 1.5 system clocks for single-edge capture, and 2.5 system clocks for capture of both edges.

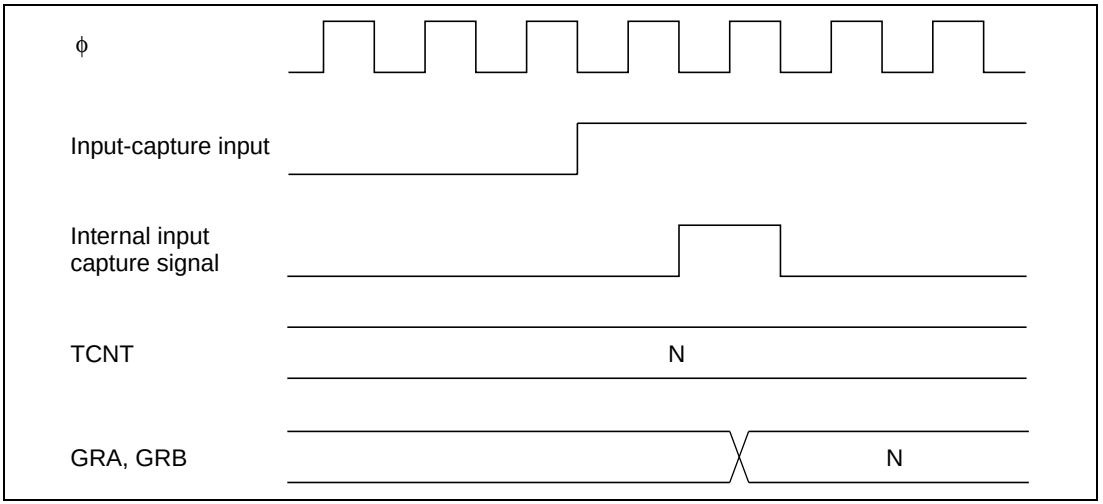


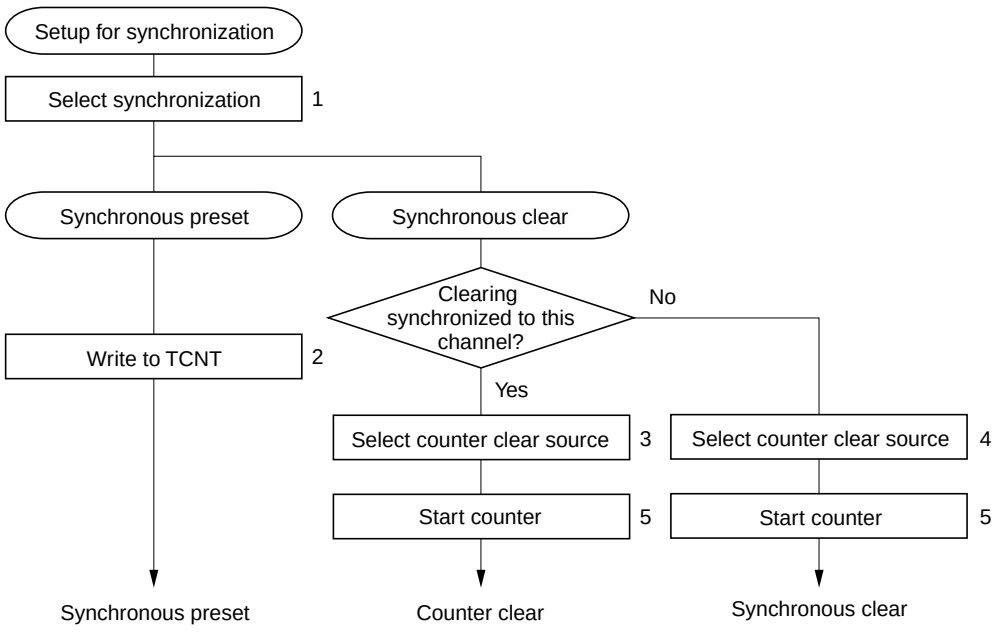
Figure 8.25 Input Capture Signal Timing

8.4.3 Synchronization

The synchronization function enables two or more timer counters to be synchronized by writing the same data to them simultaneously (synchronous preset). With appropriate TCR settings, two or more timer counters can also be cleared simultaneously (synchronous clear). Synchronization enables additional general registers to be associated with a single time base. Synchronization can be selected for all channels (0 to 4).

Sample Setup Procedure for Synchronization

Figure 8.26 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.
3. Set the CCLR1 or CCLR0 bit in TCR to have the counter cleared by compare match or input capture.
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 8.26 Setup Procedure for Synchronization (Example)

Example of Synchronization

Figure 8.27 shows an example of synchronization. Channels 0, 1, and 2 are synchronized, and are set to operate in PWM mode. Channel 0 is set for counter clearing by compare match with GRB0. Channels 1 and 2 are set for synchronous counter clearing. The timer counters in channels 0, 1, and 2 are synchronously preset, and are synchronously cleared by compare match with GRB0. A three-phase PWM waveform is output from pins $TIOCA_0$, $TIOCA_1$, and $TIOCA_2$. For further information on PWM mode, see section 8.4.4, PWM Mode.

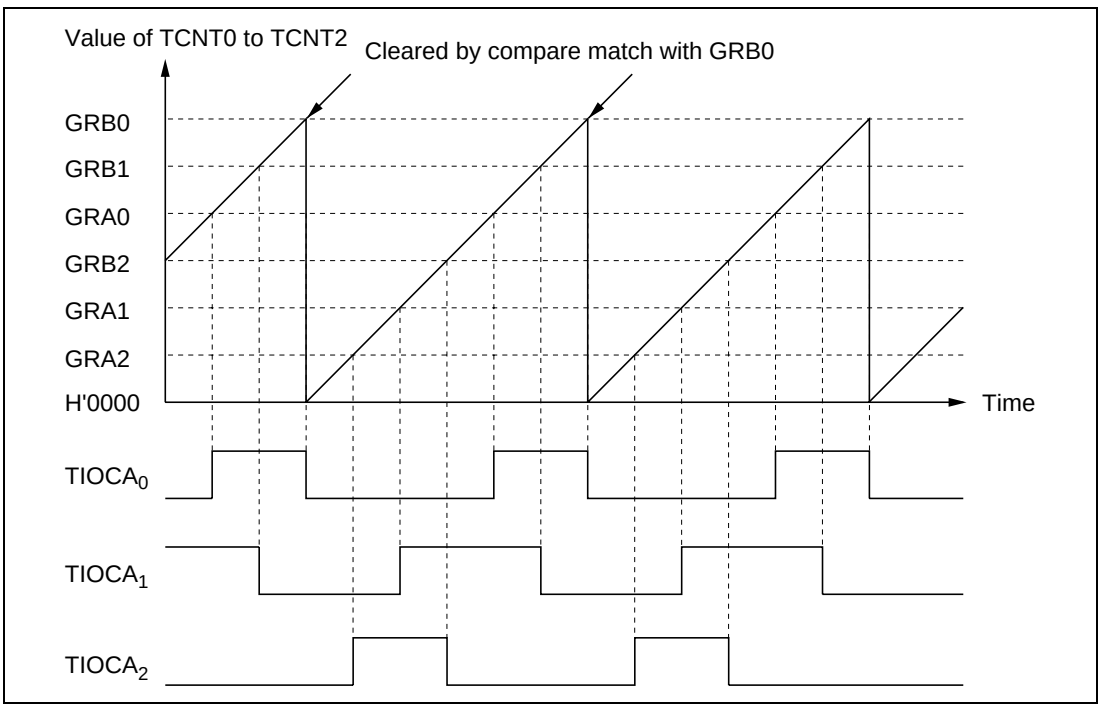


Figure 8.27 Synchronization (Example)

8.4.4 PWM Mode

In PWM mode GRA and GRB are paired and a PWM waveform is output from the TIOCA pin. GRA specifies the time at which the PWM output changes to 1. GRB specifies the time at which the PWM output changes to 0. If either GRA or GRB is selected as the counter clear source, a PWM waveform with a duty cycle from 0% to 100% is output at the TIOCA pin. PWM mode can be selected in all channels (0 to 4).

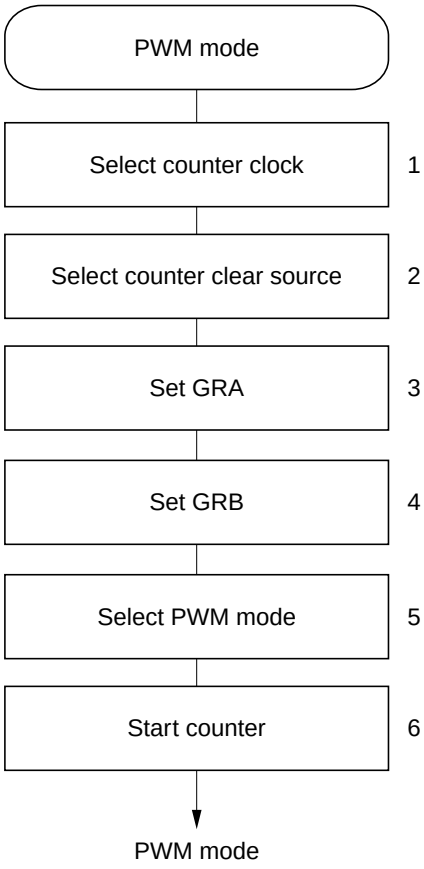
Table 8.4 summarizes the PWM output pins and corresponding registers. If the same value is set in GRA and GRB, the output does not change when compare match occurs.

Table 8.4 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

Sample Setup Procedure for PWM Mode

Figure 8.28 shows a sample procedure for setting up PWM mode.



1. Set bits TPSC2 to TPSC0 in TCR to select the counter clock source. If an external clock source is selected, set bits CKEG1 and CKEG0 in TCR to select the desired edge(s) of the external clock signal.
2. Set bits CCLR1 and CCLR0 in TCR 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 select PWM mode. When PWM mode is selected, regardless of the TIOR contents, GRA and GRB become output compare registers specifying the times at which the PWM goes to 1 and 0. The TIOCA pin automatically becomes the PWM output pin. The TIOCB pin conforms to the settings of bits IOB1 and IOB0 in TIOR. If TIOCB output is not desired, clear both IOB1 and IOB0 to 0.
6. Set the STR bit to 1 in TSTR to start the timer counter.

Figure 8.28 Setup Procedure for PWM Mode (Example)

Examples of PWM Mode

Figure 8.29 shows examples of operation in PWM mode. The PWM waveform is output from the TIOCA pin. The output goes to 1 at compare match with GRA, and to 0 at compare match with GRB.

In the examples shown, TCNT is cleared by compare match with GRA or GRB. Synchronized operation and free-running counting are also possible.

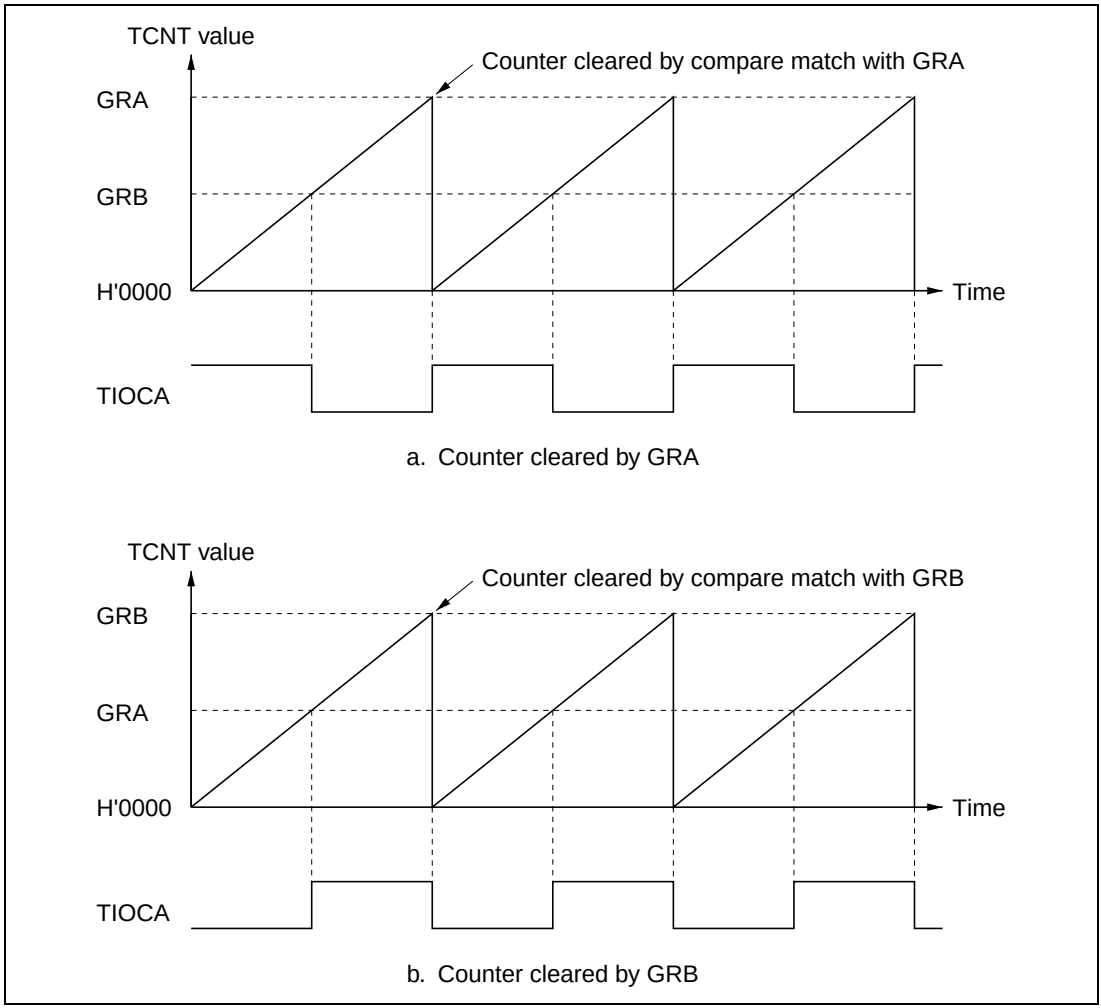


Figure 8.29 PWM Mode (Example 1)

Figure 8.30 shows examples of the output of PWM waveforms with duty cycles of 0% and 100%. If the counter is cleared by compare match with GRB, and GRA is set to a higher value than GRB, the duty cycle is 0%. If the counter is cleared by compare match with GRA, and GRB is set to a higher value than GRA, the duty cycle is 100%.

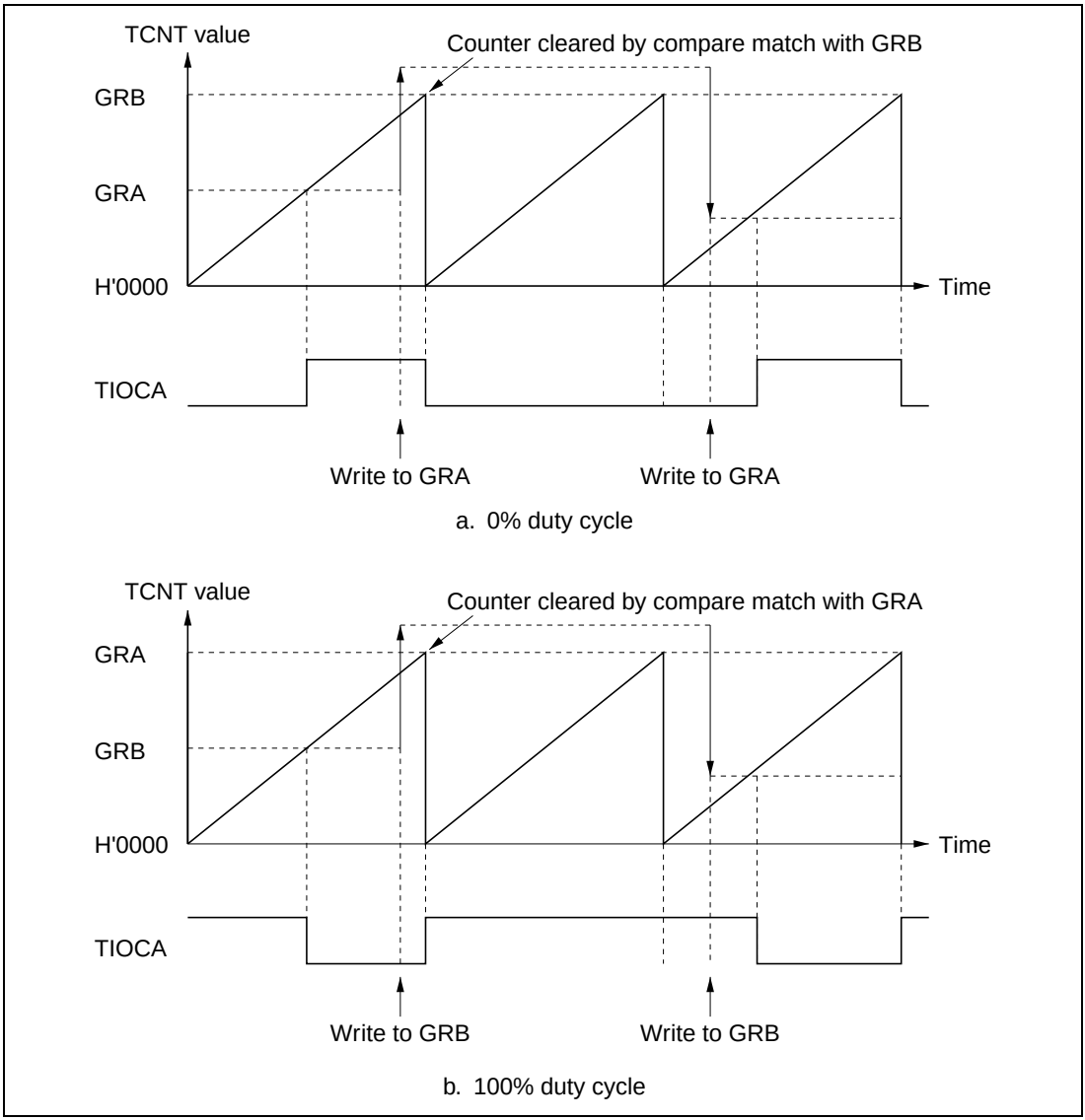


Figure 8.30 PWM Mode (Example 2)

8.4.5 Reset-Synchronized PWM Mode

In reset-synchronized PWM mode channels 3 and 4 are combined to produce three pairs of complementary PWM waveforms, all having one waveform transition point in common.

When reset-synchronized PWM mode is selected $TIOCA_3$, $TIOCB_3$, $TIOCA_4$, $TOCXA_4$, $TIOCB_4$, and $TOCXB_4$ automatically become PWM output pins, and $TCNT_3$ functions as an up-counter.

Table 8.5 lists the PWM output pins. Table 8.6 summarizes the register settings.

Table 8.5 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 output 1)
4	$TIOCA_4$	PWM output 2
	$TOCXA_4$	PWM output 2' (complementary waveform to PWM output 2)
	$TIOCB_4$	PWM output 3
	$TOCXB_4$	PWM output 3' (complementary waveform to PWM output 3)

Table 8.6 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 $TIOCB_3$
GRA4	Specifies a transition point of PWM waveforms output from $TIOCA_4$ and $TOCXA_4$
GRB4	Specifies a transition point of PWM waveforms output from $TIOCB_4$ and $TOCXB_4$

Sample Setup Procedure for Reset-Synchronized PWM Mode

Figure 8.31 shows a sample procedure for setting up reset-synchronized PWM mode.

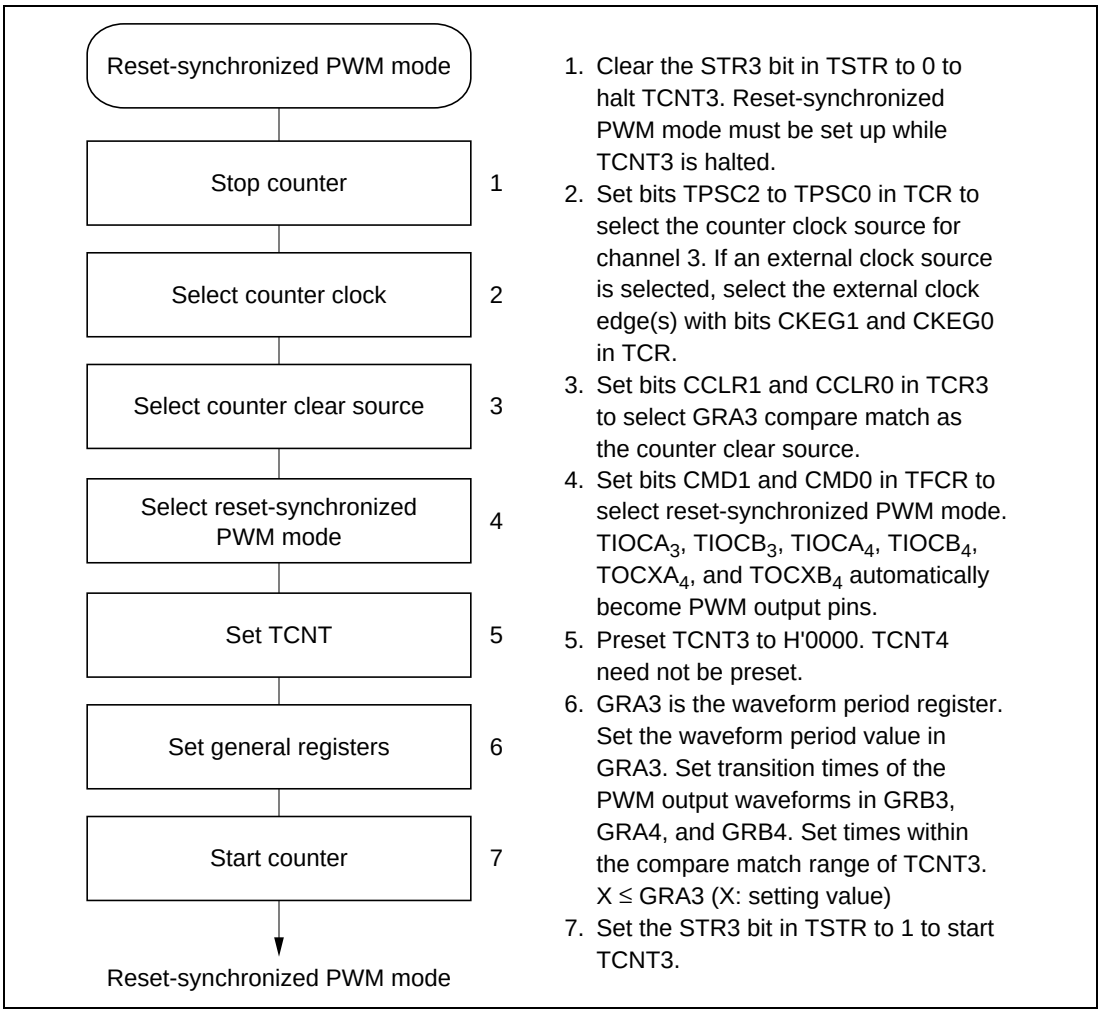


Figure 8.31 Setup Procedure for Reset-Synchronized PWM Mode (Example)

Example of Reset-Synchronized PWM Mode

Figure 8.32 shows an example of operation in reset-synchronized PWM mode. TCNT3 operates as an up-counter in this mode. TCNT4 operates independently, detached from GRA4 and GRB4. When TCNT3 matches GRA3, TCNT3 is cleared and resumes counting from H'0000. The PWM outputs toggle at compare match with GRB3, GRA4, GRB4, and TCNT3 respectively, and when the counter is cleared.

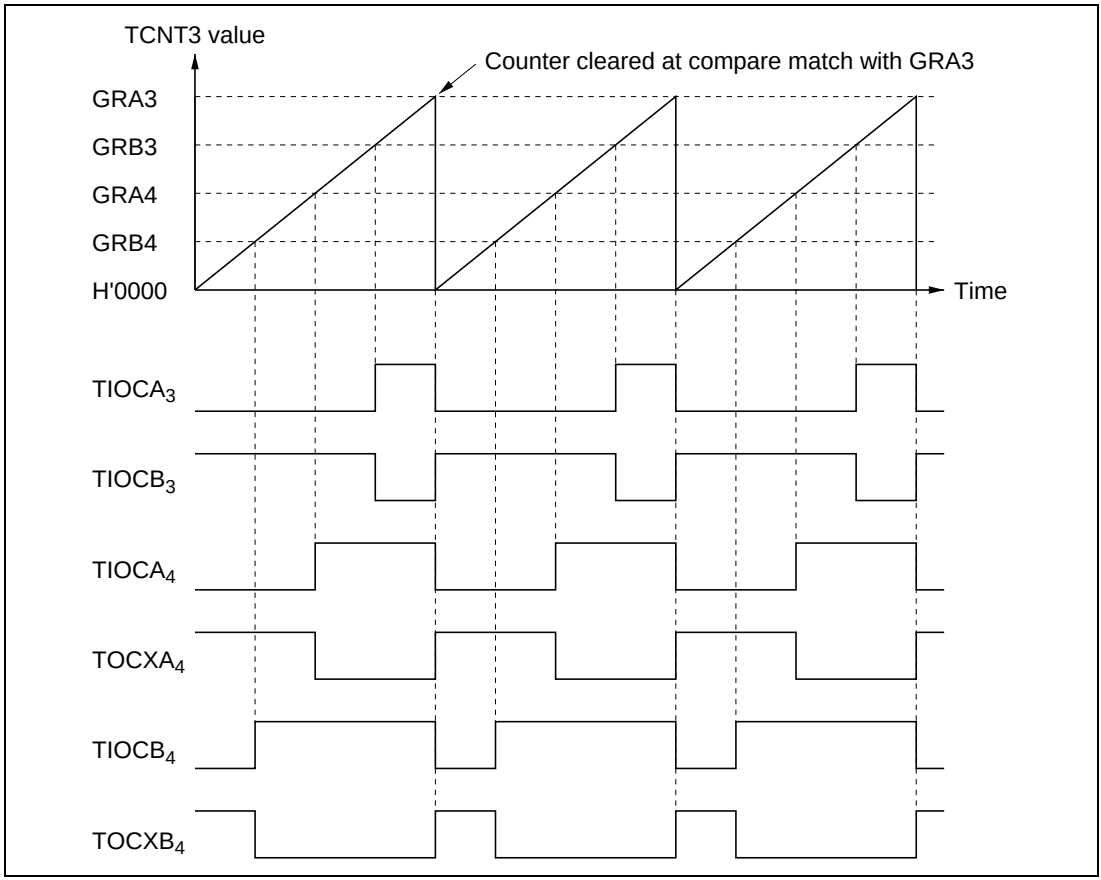


Figure 8.32 Operation in Reset-Synchronized PWM Mode (Example) (when OLS3 = OLS4 = 1)

For the settings and operation when reset-synchronized PWM mode and buffer mode are both selected, see section 8.4.8, Buffering.

8.4.6 Complementary PWM Mode

In complementary PWM mode channels 3 and 4 are combined to output three pairs of complementary, non-overlapping PWM waveforms.

When complementary PWM mode is selected $TIOCA_3$, $TIOCB_3$, $TIOCA_4$, $TOCXA_4$, $TIOCB_4$, and $TOCXB_4$ automatically become PWM output pins, and TCNT3 and TCNT4 function as up/down-counters.

Table 8.7 lists the PWM output pins. Table 8.8 summarizes the register settings.

Table 8.7 Output Pins in Complementary PWM Mode

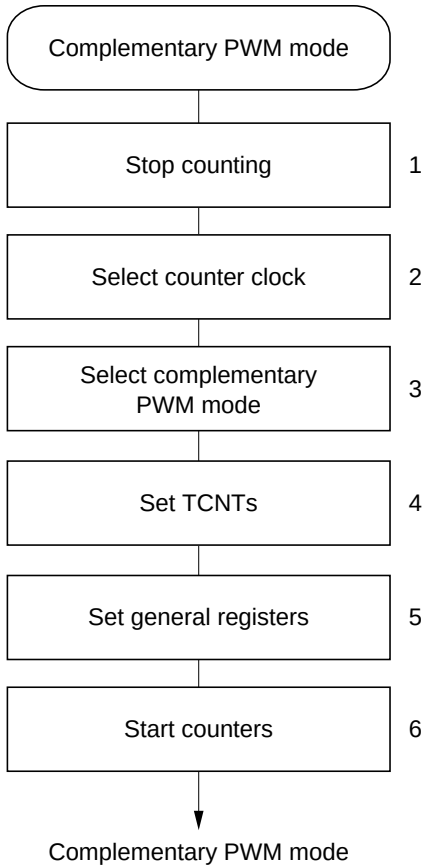
Channel	Output Pin	Description
3	$TIOCA_3$	PWM output 1
	$TIOCB_3$	PWM output 1' (non-overlapping complementary waveform to PWM output 1)
4	$TIOCA_4$	PWM output 2
	$TOCXA_4$	PWM output 2' (non-overlapping complementary waveform to PWM output 2)
	$TIOCB_4$	PWM output 3
	$TOCXB_4$	PWM output 3' (non-overlapping complementary waveform to PWM output 3)

Table 8.8 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 $TIOCB_3$
GRA4	Specifies a transition point of PWM waveforms output from $TIOCA_4$ and $TOCXA_4$
GRB4	Specifies a transition point of PWM waveforms output from $TIOCB_4$ and $TOCXB_4$

Setup Procedure for Complementary PWM Mode

Figure 8.33 shows a sample procedure for setting up complementary PWM mode.



1. Clear bits STR3 and STR4 to 0 in TSTR to halt the timer counters. Complementary PWM mode must be set up while TCNT3 and TCNT4 are halted.
2. Set bits TPSC2 to TPSC0 in TCR to select the same counter clock source for channels 3 and 4. If an external clock source is selected, select the external clock edge(s) with bits CKEG1 and CKEG0 in TCR. Do not select any counter clear source with bits CCLR1 and CCLR0 in TCR.
3. Set bits CMD1 and CMD0 in TFCR to select complementary PWM mode. $TIOCA_3$, $TIOCB_3$, $TIOCA_4$, $TIOCB_4$, $TOCXA_4$, and $TOCXB_4$ automatically become PWM output pins.
4. Clear TCNT4 to H'0000. Set the non-overlap margin in TCNT3. Do not set TCNT3 and TCNT4 to the same value.
5. GRA3 is the waveform period register. Set the upper limit value of TCNT3 minus 1 in GRA3. Set transition times of the PWM output waveforms in GRB3, GRA4, and GRB4. Set times within the compare match range of TCNT3 and TCNT4. $T \leq X$ (X: initial setting of GRB3, GRA4, or GRB4. T: initial setting of TCNT3)
6. Set bits STR3 and STR4 in TSTR to 1 to start TCNT3 and TCNT4.

Note: After exiting complementary PWM mode, to resume operating in complementary PWM mode, follow the entire setup procedure from step 1 again.

Figure 8.33 Setup Procedure for Complementary PWM Mode (Example)

Clearing Complementary PWM Mode

Figure 8.34 shows a sample procedure for clearing complementary PWM mode.

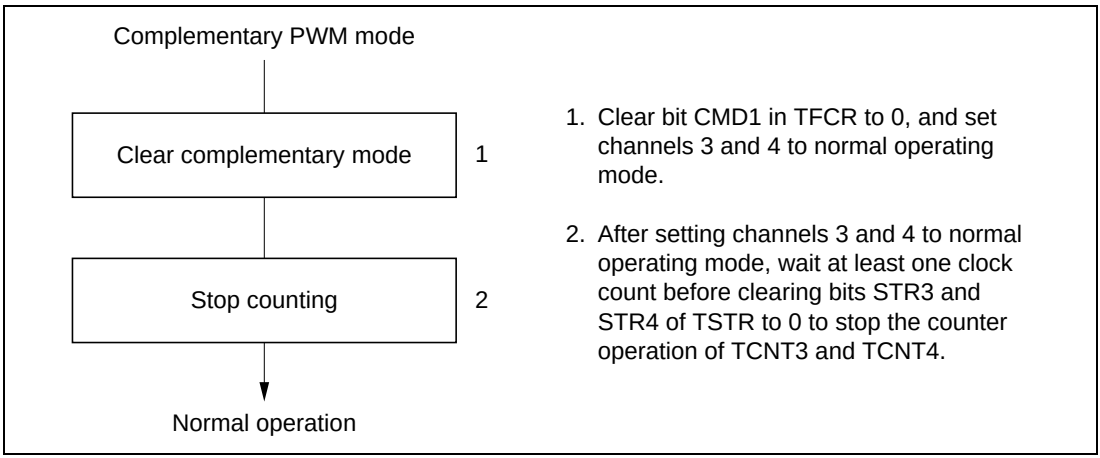


Figure 8.34 Clearing Procedure for Complementary PWM Mode (Example)

Examples of Complementary PWM Mode

Figure 8.35 shows an example of operation in complementary PWM mode. TCNT3 and TCNT4 operate as up/down-counters, counting down from compare match between TCNT3 and GRA3 and counting up from the point at which TCNT4 underflows. During each up-and-down counting cycle, PWM waveforms are generated by compare match with general registers GRB3, GRA4, and GRB4. Since TCNT3 is initially set to a higher value than TCNT4, compare match events occur in the sequence TCNT3, TCNT4, TCNT4, TCNT3.

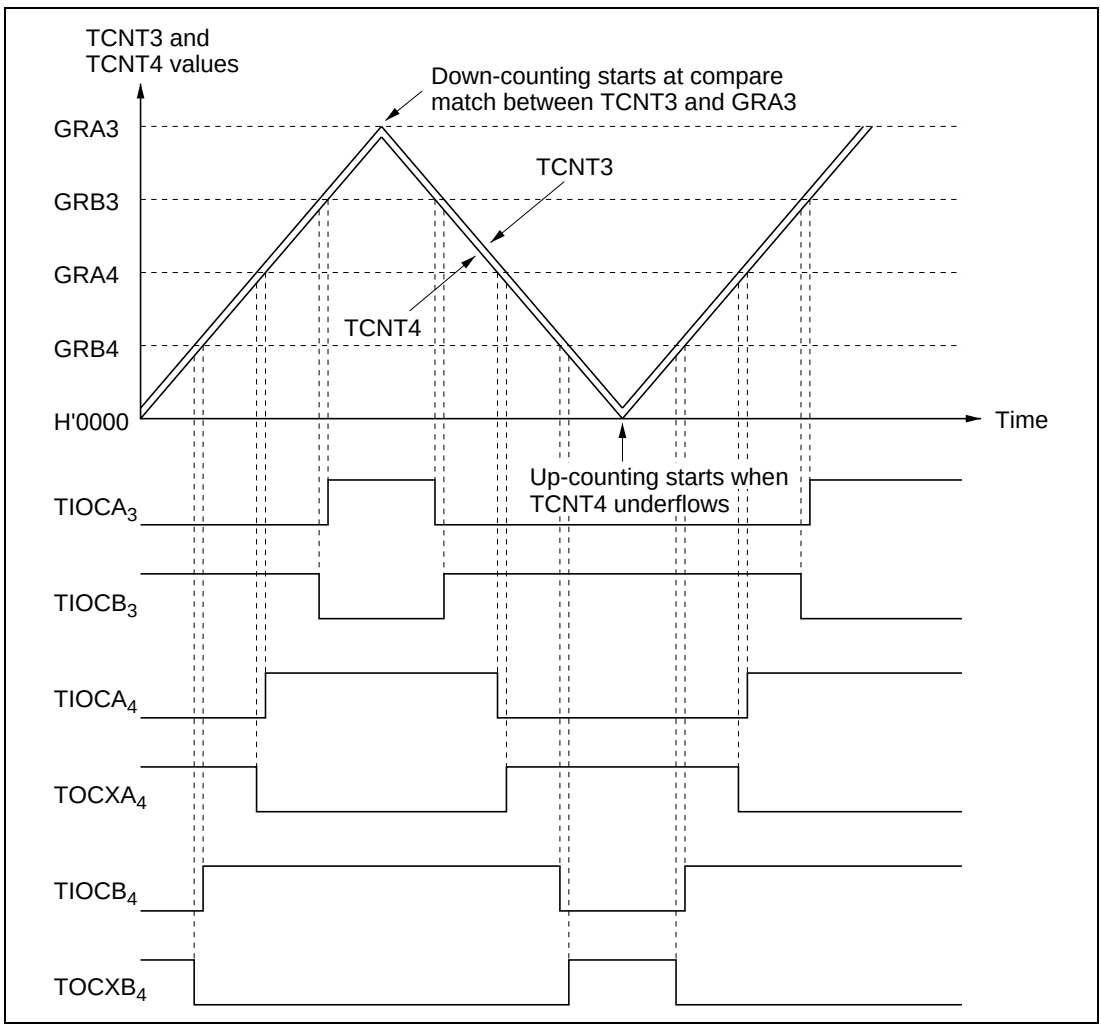


Figure 8.35 Operation in Complementary PWM Mode (Example 1)
(when OLS3 = OLS4 = 1)

Figure 8.36 shows examples of waveforms with 0% and 100% duty cycles (in one phase) in complementary PWM mode. In this example the outputs change at compare match with GRB3, so waveforms with duty cycles of 0% or 100% can be output by setting GRB3 to a value larger than GRA3. The duty cycle can be changed easily during operation by use of the buffer registers. For further information see section 8.4.8, Buffering.

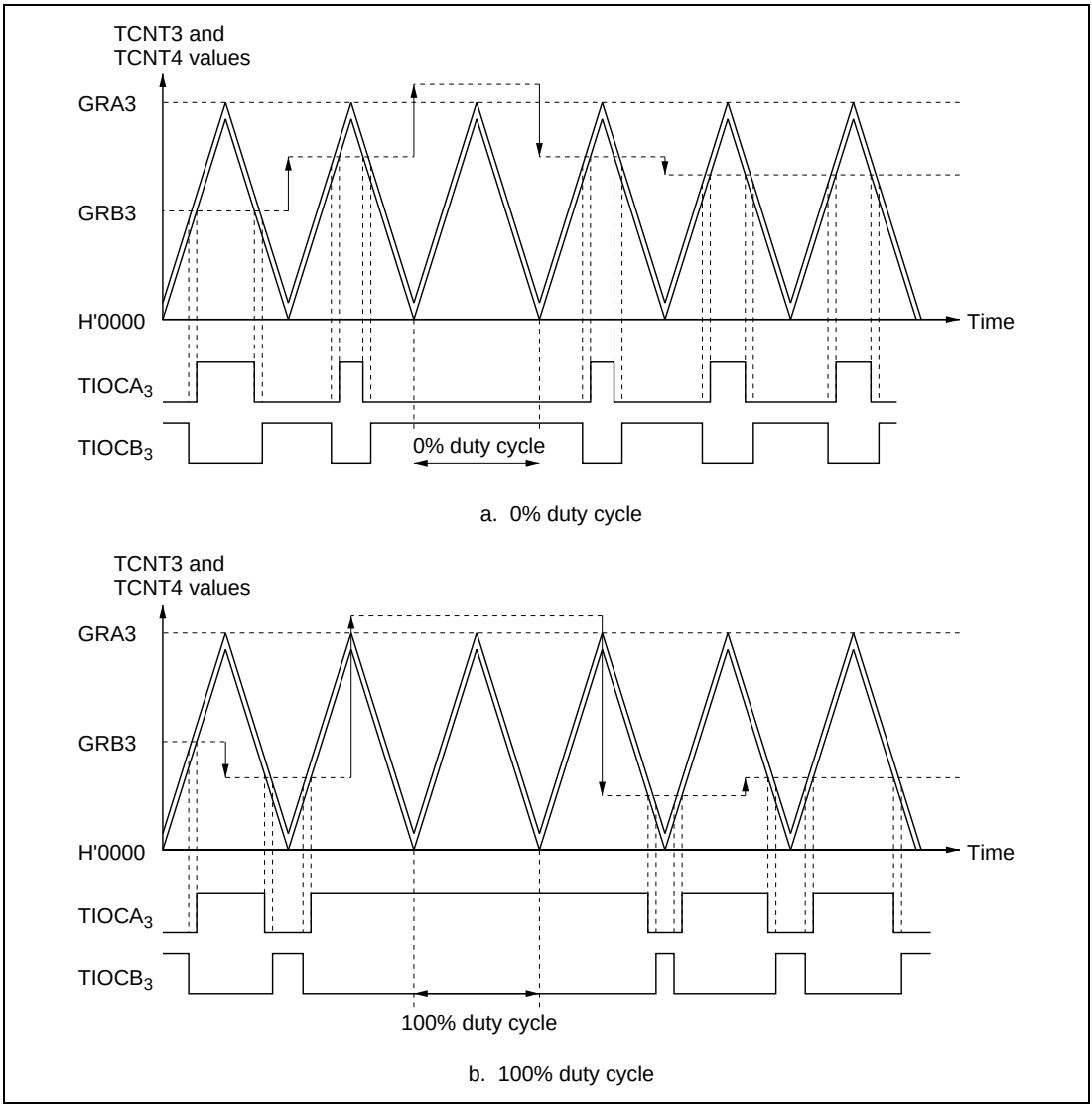


Figure 8.36 Operation in Complementary PWM Mode (Example 2) (when OLS3 = OLS4 = 1)

In complementary PWM mode, TCNT3 and TCNT4 overshoot and undershoot at the transitions between up-counting and down-counting. The setting conditions for the IMFA bit in channel 3 and the OVF bit in channel 4 differ from the usual conditions. In buffered operation the buffer transfer conditions also differ. Timing diagrams are shown in figures 8.37 and 8.38.

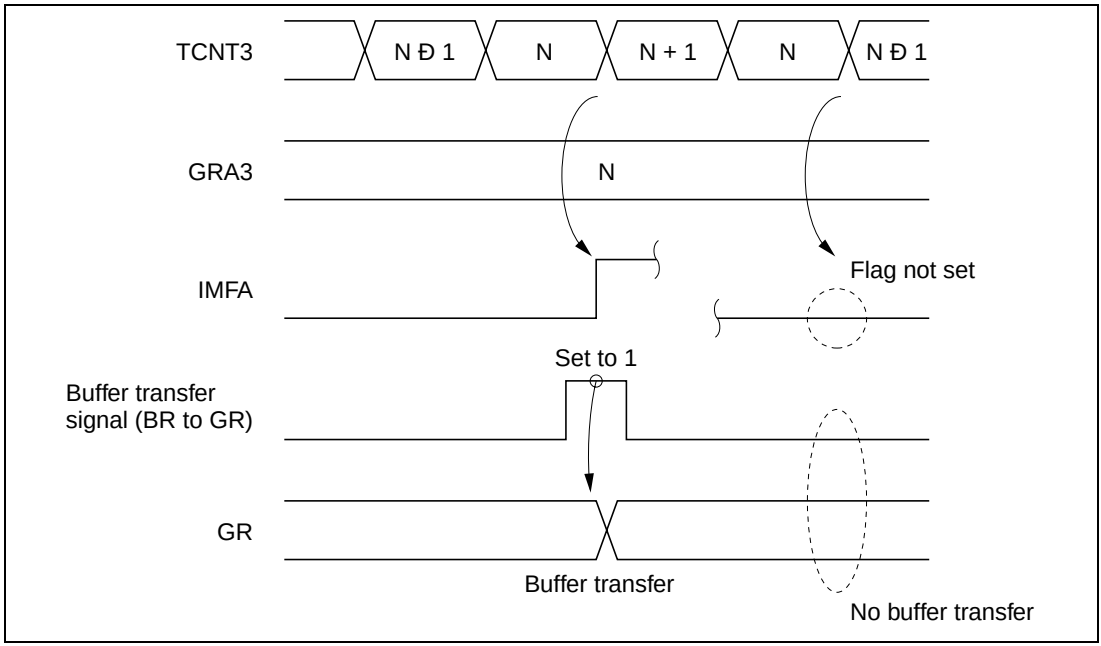


Figure 8.37 Overshoot Timing

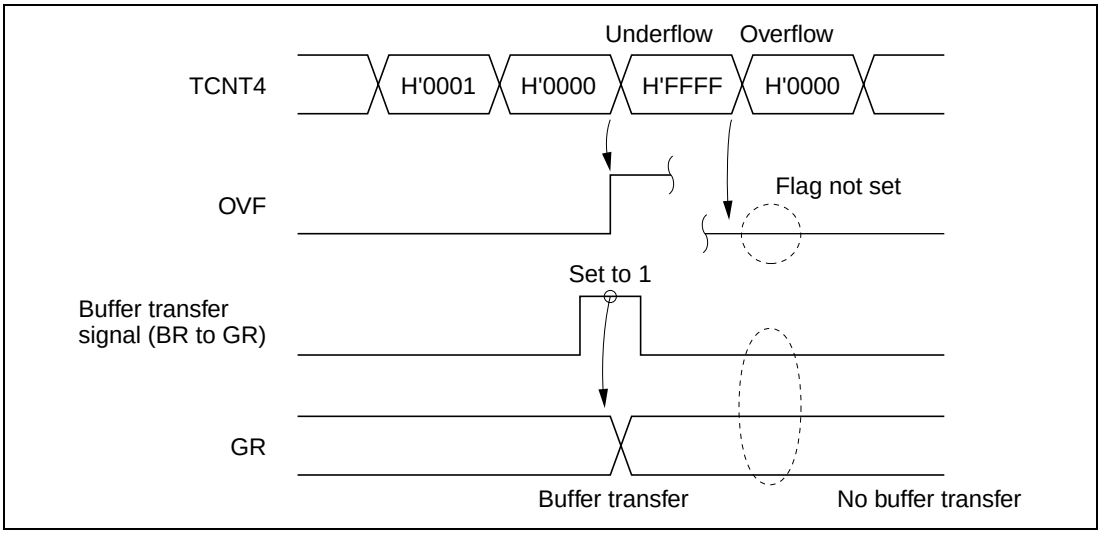


Figure 8.38 Undershoot Timing

In channel 3, IMFA is set to 1 only during up-counting. In channel 4, OVF is set to 1 only when an underflow occurs. When buffering is selected, buffer register contents are transferred to the general register at compare match A3 during up-counting, and when TCNT4 underflows.

General Register Settings in Complementary PWM Mode

When setting up general registers for complementary PWM mode or changing their settings 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). After the counters start and the first compare match A3 event has occurred, however, settings in this range also become possible.
- Changing settings

  Use the buffer registers. Correct waveform output may not be obtained if a general register is written to directly.
- Cautions on changes of general register settings

  Figure 8.39 shows six correct examples and one incorrect example.

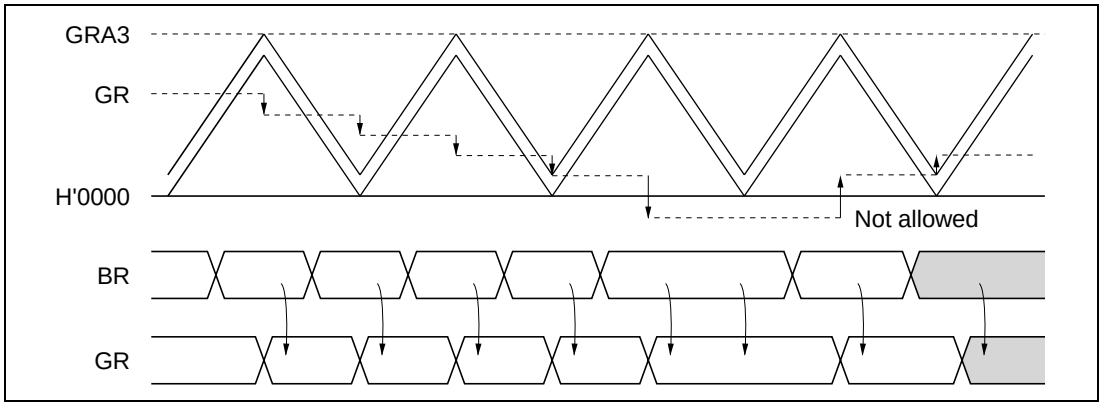


Figure 8.39 Changing a General Register Setting by Buffer Transfer (Example 1)

— 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 not transfer a buffer register value outside this range. Conversely, if the general register value is outside this range, do not transfer a value within this range. See figure 8.40.

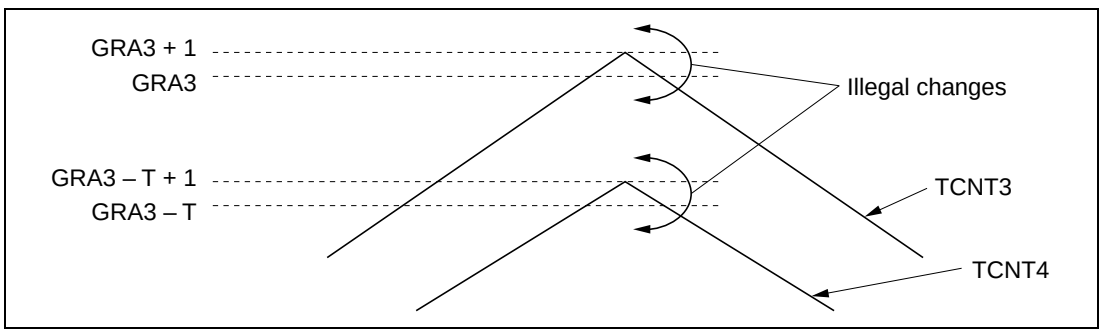


Figure 8.40 Changing a General Register Setting by Buffer Transfer (Caution 1)

— 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 transfer a buffer register value outside this range. Conversely, when a general register value is outside this range, do not transfer a value within this range. See figure 8.41.

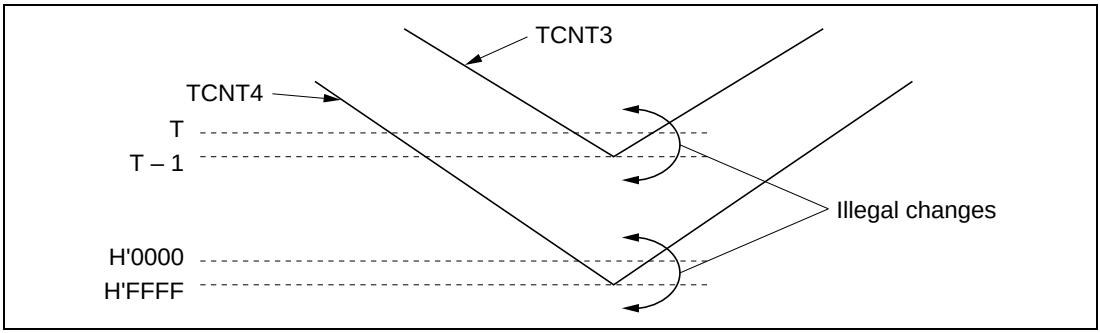


Figure 8.41 Changing a General Register Setting by Buffer Transfer (Caution 2)

— 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 register to a value outside the counting range. When a buffer register is set to a value outside the counting range, then later restored to a value within the counting range, the counting direction (up or down) must be the same both times. See figure 8.42.

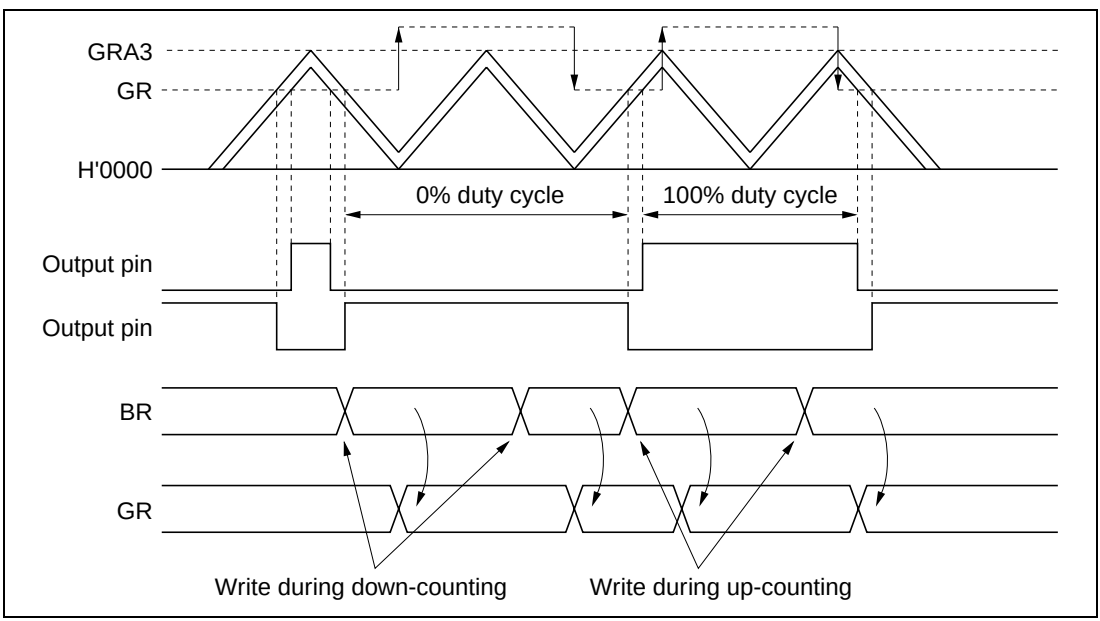


Figure 8.42 Changing a General Register Setting by Buffer Transfer (Example 2)

Settings can be made in this way by detecting GRA3 compare match or TCNT4 underflow before writing to the buffer register.

8.4.7 Phase Counting Mode

In phase counting mode the phase difference between two external clock inputs (at the TCLKA and TCLKB pins) is detected, and TCNT2 counts up or down accordingly.

In phase counting mode, the TCLKA and TCLKB pins automatically function as external clock input pins and TCNT2 becomes an up/down-counter, regardless of the settings of bits TPSC2 to TPSC0, CKEG1, and CKEG0 in TCR2. Settings of bits CCLR1, CCLR0 in TCR2, and settings in TIOR2, TIER2, TSR2, GRA2, and GRB2 are valid. The input capture and output compare 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 8.43 shows a sample procedure for setting up phase counting mode.

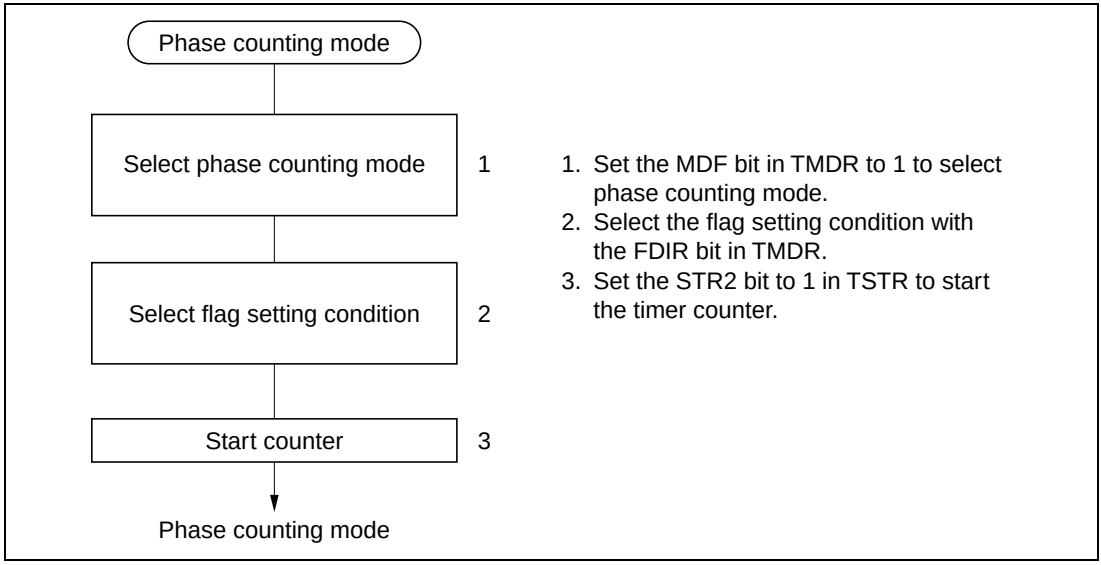


Figure 8.43 Setup Procedure for Phase Counting Mode (Example)

Example of Phase Counting Mode

Figure 8.44 shows an example of operations in phase counting mode. Table 8.9 lists the up-counting and down-counting conditions for TCNT2.

In phase counting mode both the rising and falling edges of TCLKA and TCLKB are counted. The phase difference between TCLKA and TCLKB must be at least 1.5 states, the phase overlap must also be at least 1.5 states, and the pulse width must be at least 2.5 states. See figure 8.45.

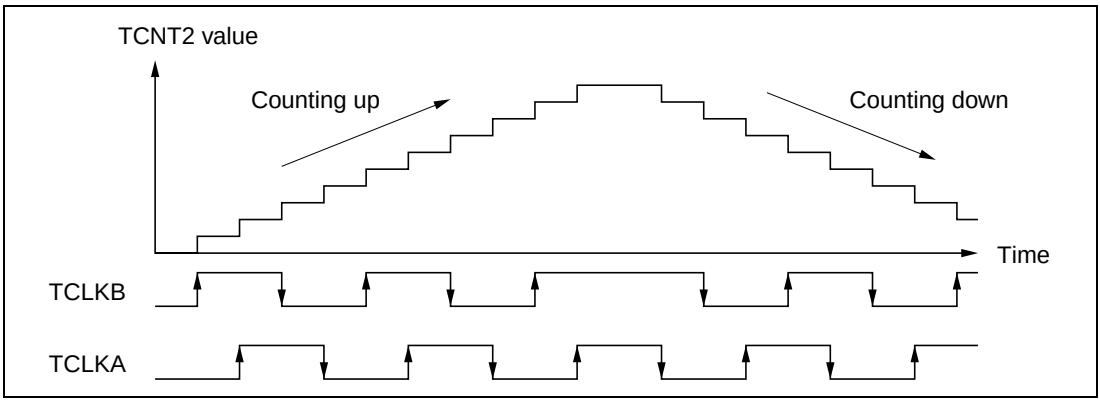


Figure 8.44 Operation in Phase Counting Mode (Example)

Table 8.9 Up/Down Counting Conditions

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

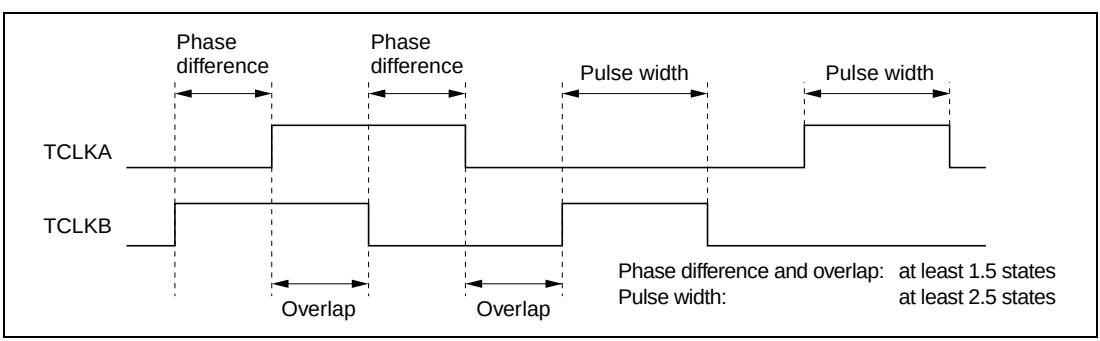


Figure 8.45 Phase Difference, Overlap, and Pulse Width in Phase Counting Mode

8.4.8 Buffering

Buffering operates differently depending on whether a general register is an output compare register or an input capture register, with further differences in reset-synchronized PWM mode and complementary PWM mode. Buffering is available only in channels 3 and 4. Buffering operations under the conditions mentioned above are described next.

- General register used for output compare

  The buffer register value is transferred to the general register at compare match. See figure 8.46.

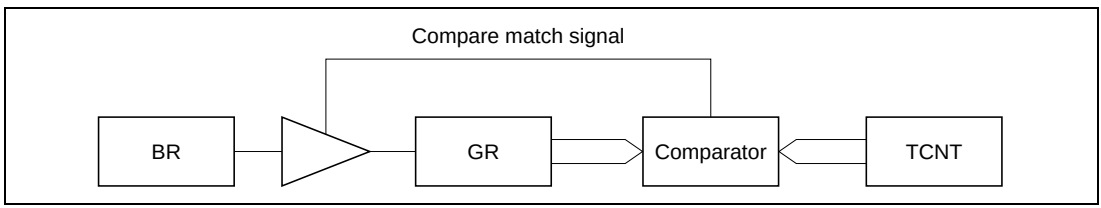


Figure 8.46 Compare Match Buffering

- General register used for input capture

  The TCNT value is transferred to the general register at input capture. The previous general register value is transferred to the buffer register.
  See figure 8.47.

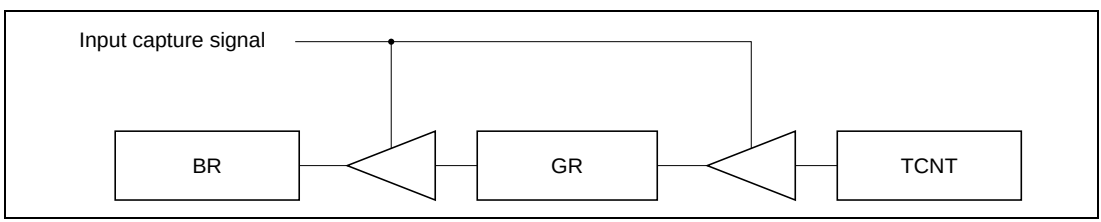


Figure 8.47 Input Capture Buffering

- Complementary PWM mode

  The buffer register value is transferred to the general register when TCNT3 and TCNT4 change counting direction. This occurs at the following two times:

  — When TCNT3 matches GRA3

  — When TCNT4 underflows

- Reset-synchronized PWM mode

  The buffer register value is transferred to the general register at compare match A3.

Sample Buffering Setup Procedure

Figure 8.48 shows a sample buffering setup procedure.

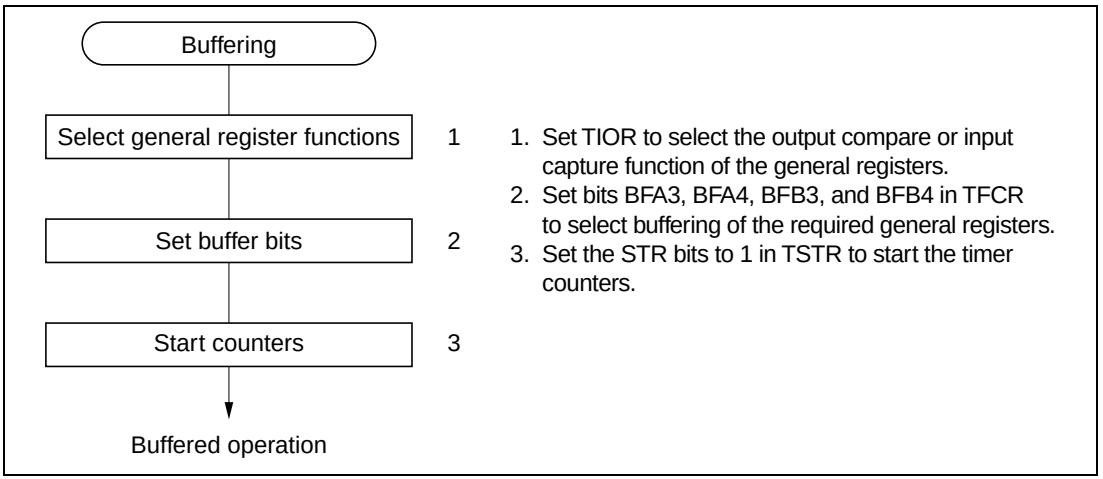


Figure 8.48 Buffering Setup Procedure (Example)

Examples of Buffering

Figure 8.49 shows an example in which GRA is set to function as an output compare register buffered by BRA, TCNT is set to operate as a periodic counter cleared by GRB compare match, and TIOCA and TIOCB are set to toggle at compare match A and B. Because of the buffer setting, when TIOCA toggles at compare match A, the BRA value is simultaneously transferred to GRA. This operation is repeated each time compare match A occurs. Figure 8.50 shows the transfer timing.

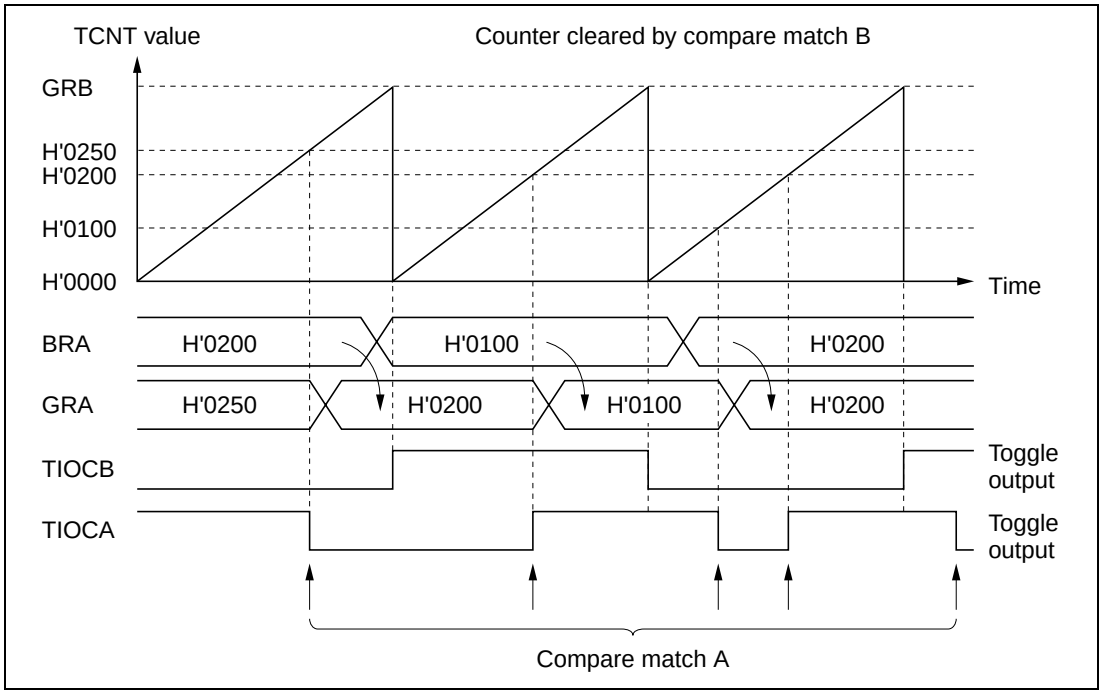


Figure 8.49 Register Buffering (Example 1: Buffering of Output Compare Register)

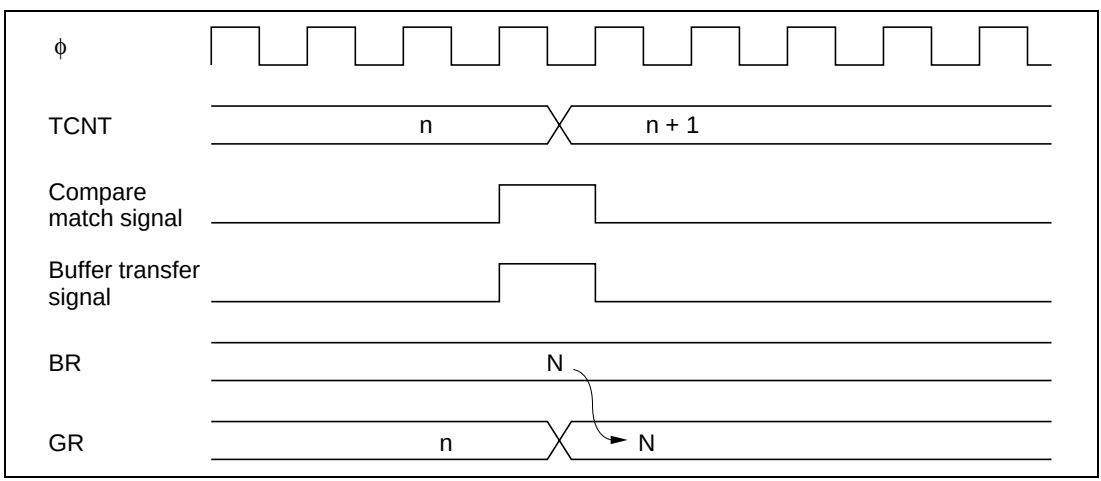


Figure 8.50 Compare Match and Buffer Transfer Timing (Example)

Figure 8.51 shows an example in which GRA is set to function as an input capture register buffered by BRA, and TCNT is cleared by input capture B. The falling edge is selected as the input capture edge at TIOCB. Both edges are selected as input capture edges at TIOCA. Because of the buffer setting, when the TCNT value is captured into GRA at input capture A, the previous GRA value is simultaneously transferred to BRA. Figure 8.52 shows the transfer timing.

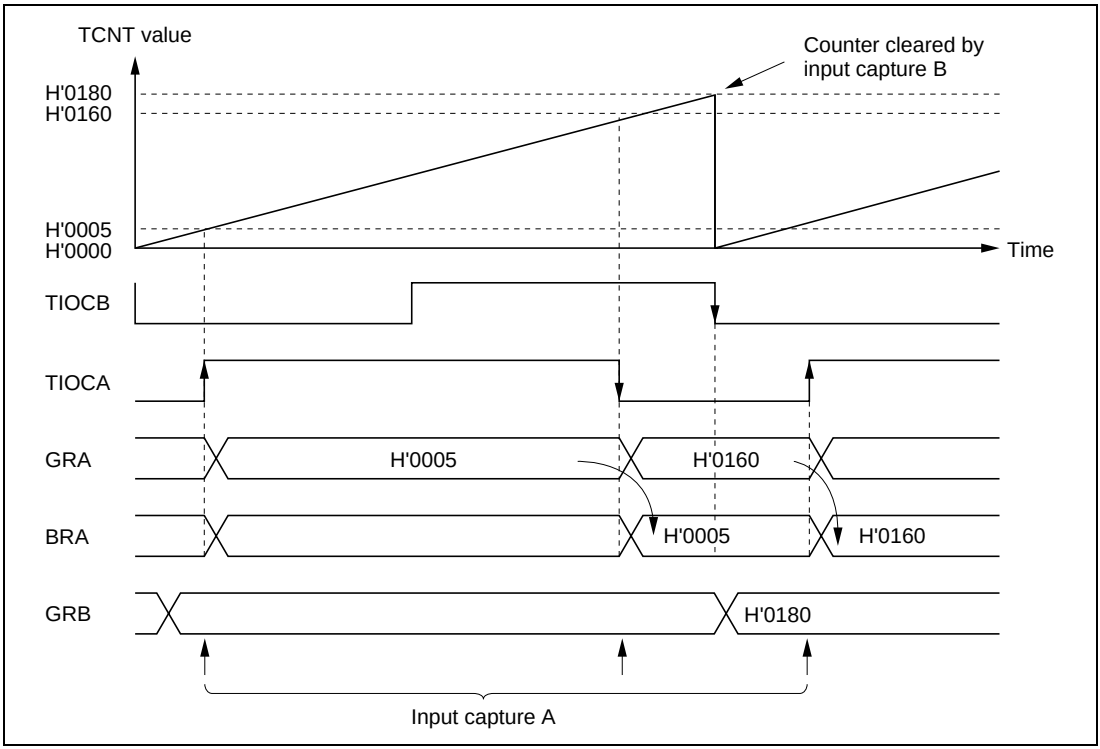


Figure 8.51 Register Buffering (Example 2: Buffering of Input Capture Register)

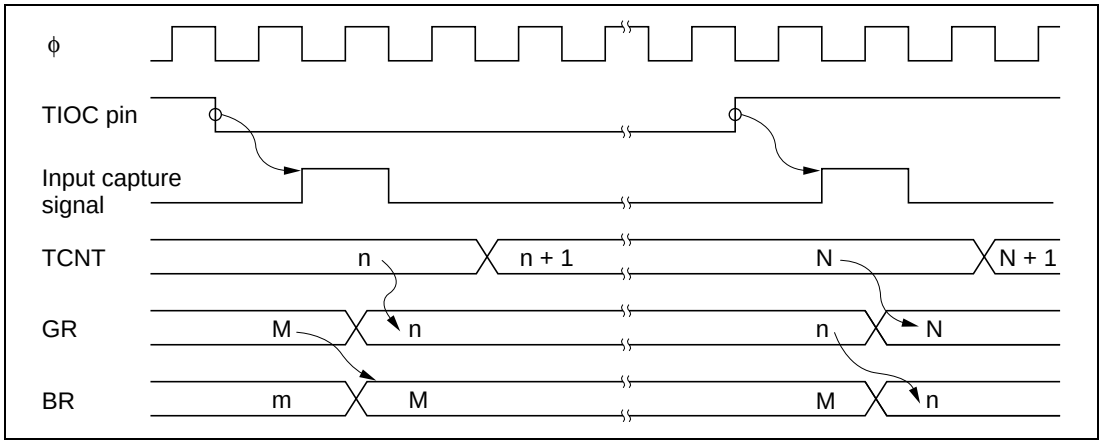


Figure 8.52 Input Capture and Buffer Transfer Timing (Example)

Figure 8.53 shows an example in which GRB3 is buffered by BRB3 in complementary PWM mode. Buffering is used to set GRB3 to a higher value than GRA3, generating a PWM waveform with 0% duty cycle. The BRB3 value is transferred to GRB3 when TCNT3 matches GRA3, and when TCNT4 underflows.

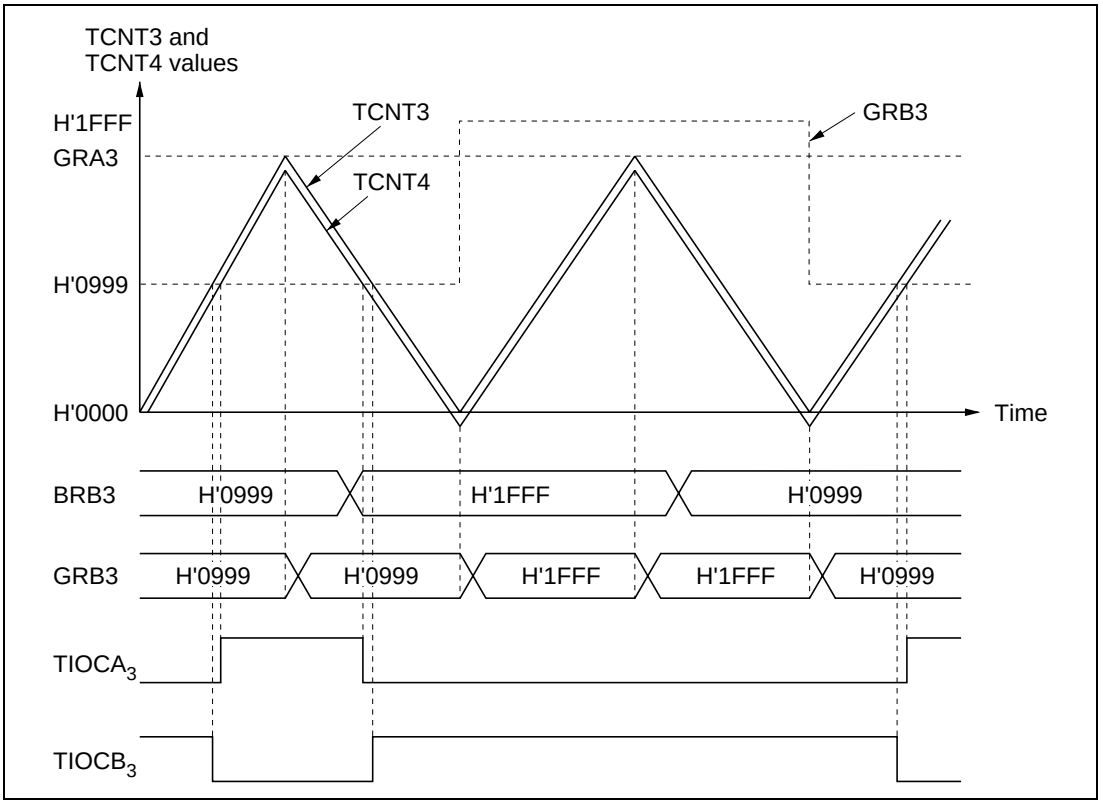


Figure 8.53 Register Buffering (Example 4: Buffering in Complementary PWM Mode)

8.4.9 ITU Output Timing

The ITU outputs from channels 3 and 4 can be disabled by bit settings in TOER or by an external trigger, or inverted by bit settings in TOCR.

Timing of Enabling and Disabling of ITU Output by TOER

In this example an ITU output is disabled by clearing a master enable bit to 0 in TOER. An arbitrary value can be output by appropriate settings of the data register (DR) and data direction register (DDR) of the corresponding input/output port. Figure 8.54 illustrates the timing of the enabling and disabling of ITU output by TOER.

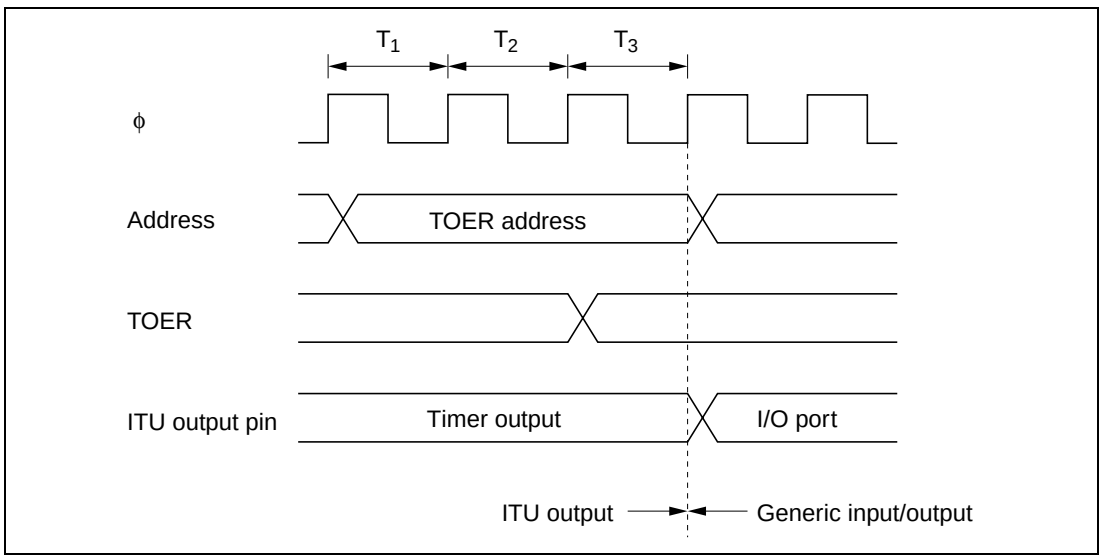


Figure 8.54 Timing of Disabling of ITU Output by Writing to TOER (Example)

Timing of Disabling of ITU Output by External Trigger

If the XTGD bit is cleared to 0 in TOCR in reset-synchronized PWM mode or complementary PWM mode, when an input capture A signal occurs in channel 1, the master enable bits are cleared to 0 in TOER, disabling ITU output. Figure 8.55 shows the timing.

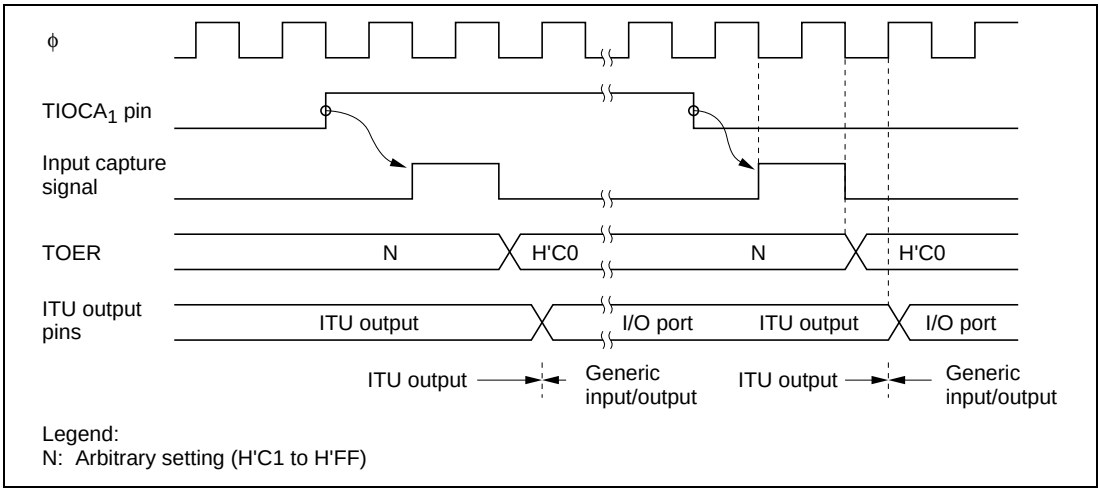


Figure 8.55 Timing of Disabling of ITU Output by External Trigger (Example)

Timing of Output Inversion by TOCR

The output levels in reset-synchronized PWM mode and complementary PWM mode can be inverted by inverting the output level select bits (OLS4 and OLS3) in TOCR. Figure 8.56 shows the timing.

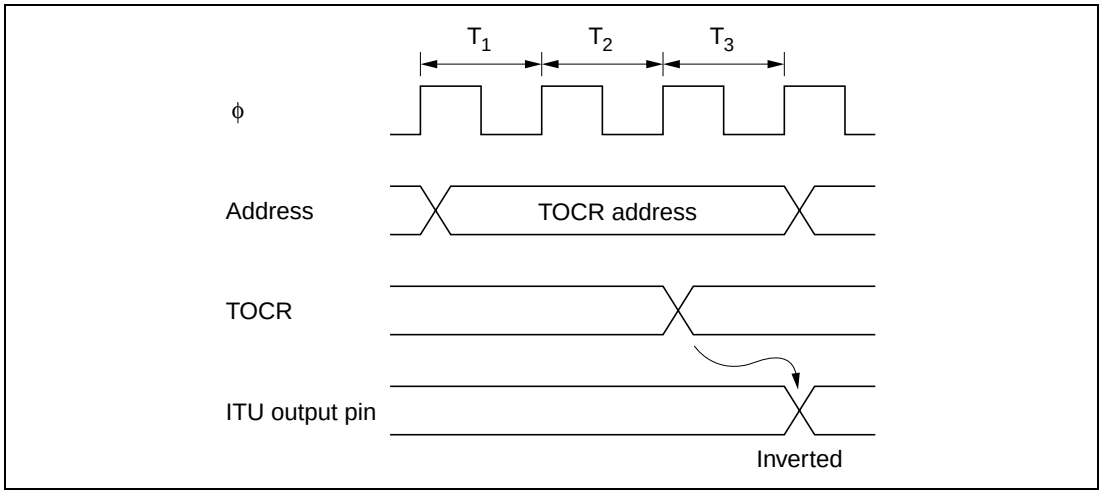


Figure 8.56 Timing of Inverting of ITU Output Level by Writing to TOCR (Example)

8.5 Interrupts

The ITU has two types of interrupts: input capture/compare match interrupts, and overflow interrupts.

8.5.1 Setting of Status Flags

Timing of Setting of IMFA and IMFB at Compare Match

IMFA and IMFB are set to 1 by a compare match signal generated when TCNT matches a general register (GR). The compare match signal is generated in the last state in which the values match (when TCNT is updated from the matching count to the next count). Therefore, when TCNT matches a general register, the compare match signal is not generated until the next timer clock input. Figure 8.57 shows the timing of the setting of IMFA and IMFB.

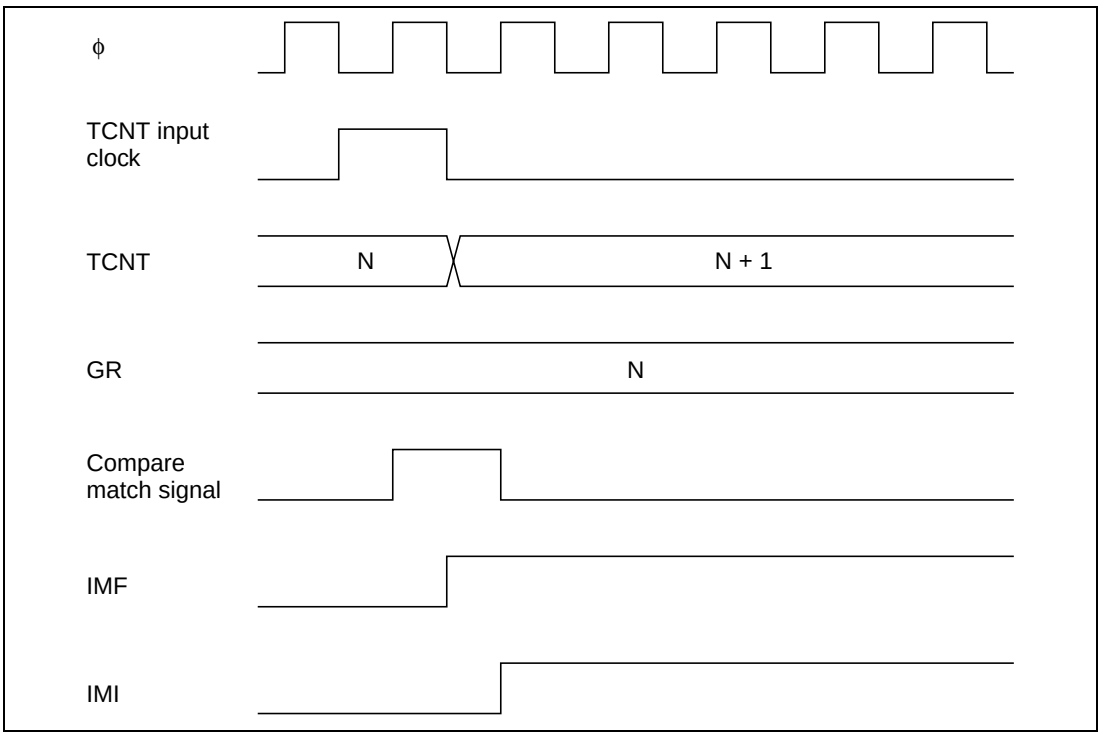


Figure 8.57 Timing of Setting of IMFA and IMFB by Compare Match

Timing of Setting of IMFA and IMFB by Input Capture

IMFA and IMFB are set to 1 by an input capture signal. The TCNT contents are simultaneously transferred to the corresponding general register. Figure 8.58 shows the timing.

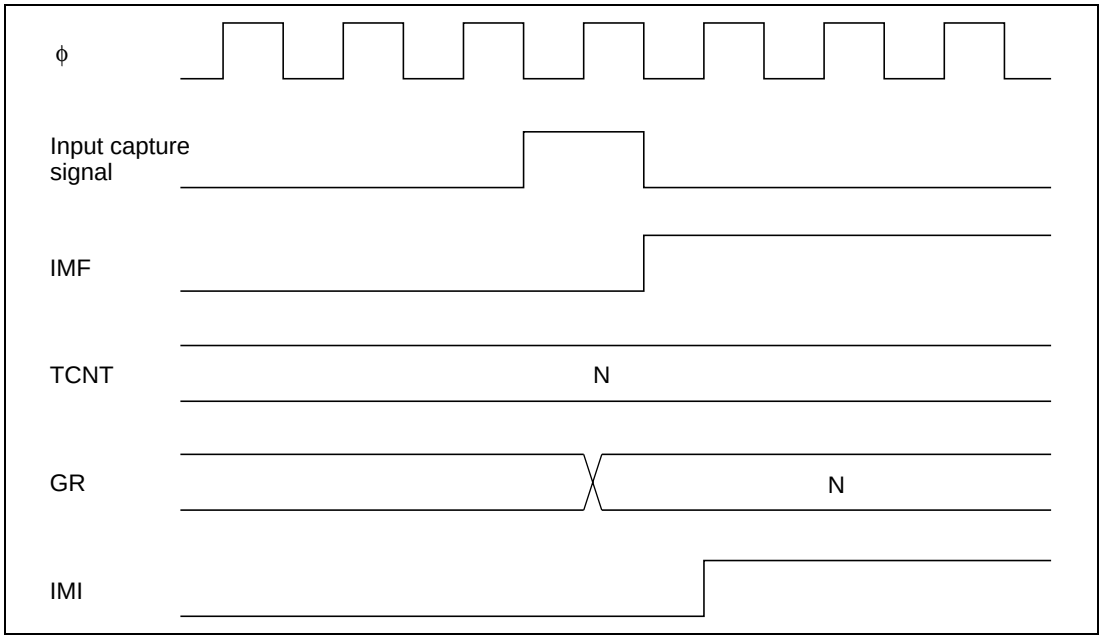


Figure 8.58 Timing of Setting of IMFA and IMFB by Input Capture

Timing of Setting of Overflow Flag (OVF)

OVF is set to 1 when TCNT overflows from H'FFFF to H'0000 or underflows from H'0000 to H'FFFF. Figure 8.59 shows the timing.

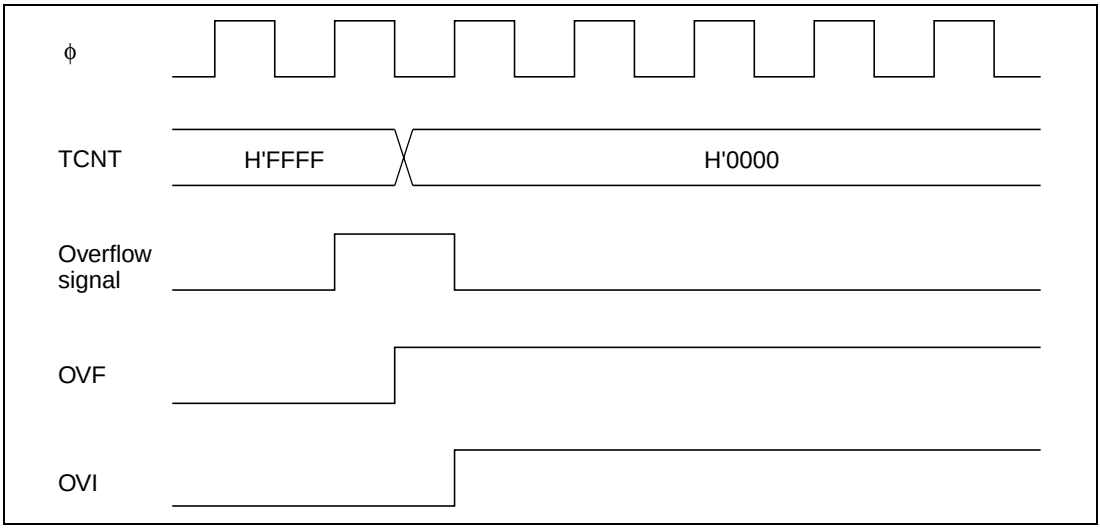


Figure 8.59 Timing of Setting of OVF

8.5.2 Clearing of Status Flags

If the CPU reads a status flag while it is set to 1, then writes 0 in the status flag, the status flag is cleared. Figure 8.60 shows the timing.

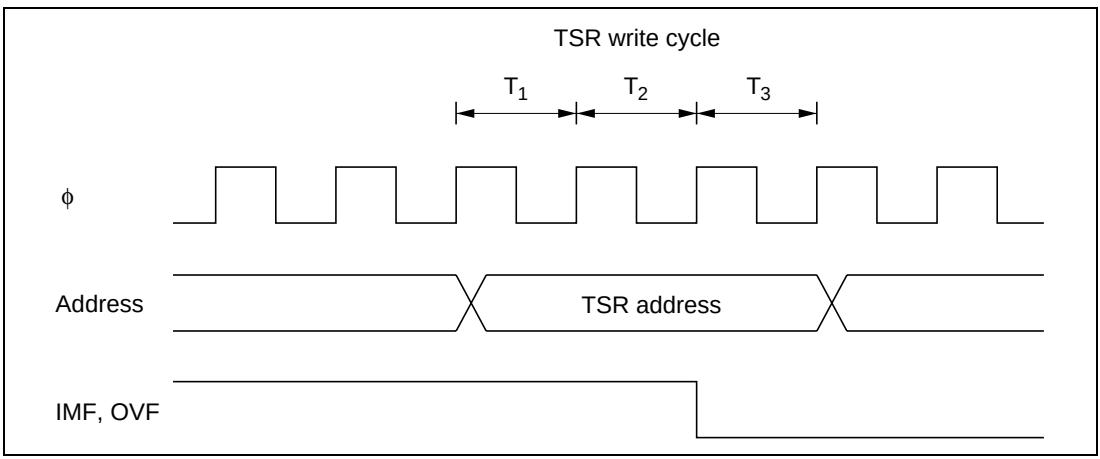


Figure 8.60 Timing of Clearing of Status Flags

8.5.3 Interrupt Sources

Each ITU channel can generate a compare match/input capture A interrupt, a compare match/input capture B interrupt, and an overflow interrupt. In total there are 15 interrupt sources, all independently vectored. An interrupt is requested when the interrupt request flag and interrupt enable bit are both set to 1.

The priority order of the channels can be modified in interrupt priority registers A and B (IPRA and IPRB). For details see section 5, Interrupt Controller.

Table 8.10 lists the interrupt sources.

Table 8.10 ITU Interrupt Sources

Channel	Interrupt Source	Description	Priority*
0	IMIA0	Compare match/input capture A0	High
	IMIB0	Compare match/input capture B0	
	OVI0	Overflow 0	
1	IMIA1	Compare match/input capture A1	
	IMIB1	Compare match/input capture B1	
	OVI1	Overflow 1	
2	IMIA2	Compare match/input capture A2	
	IMIB2	Compare match/input capture B2	
	OVI2	Overflow 2	
3	IMIA3	Compare match/input capture A3	
	IMIB3	Compare match/input capture B3	
	OVI3	Overflow 3	
4	IMIA4	Compare match/input capture A4	
	IMIB4	Compare match/input capture B4	
	OVI4	Overflow 4	Low

Note: * The priority immediately after a reset is indicated. Inter-channel priorities can be changed by settings in IPRA and IPRB.

8.6 Usage Notes

This section describes contention and other matters requiring special attention during ITU operations.

Contention between TCNT Write and Clear

If a counter clear signal occurs in the T_3 state of a TCNT write cycle, clearing of the counter takes priority and the write is not performed. See figure 8.61.

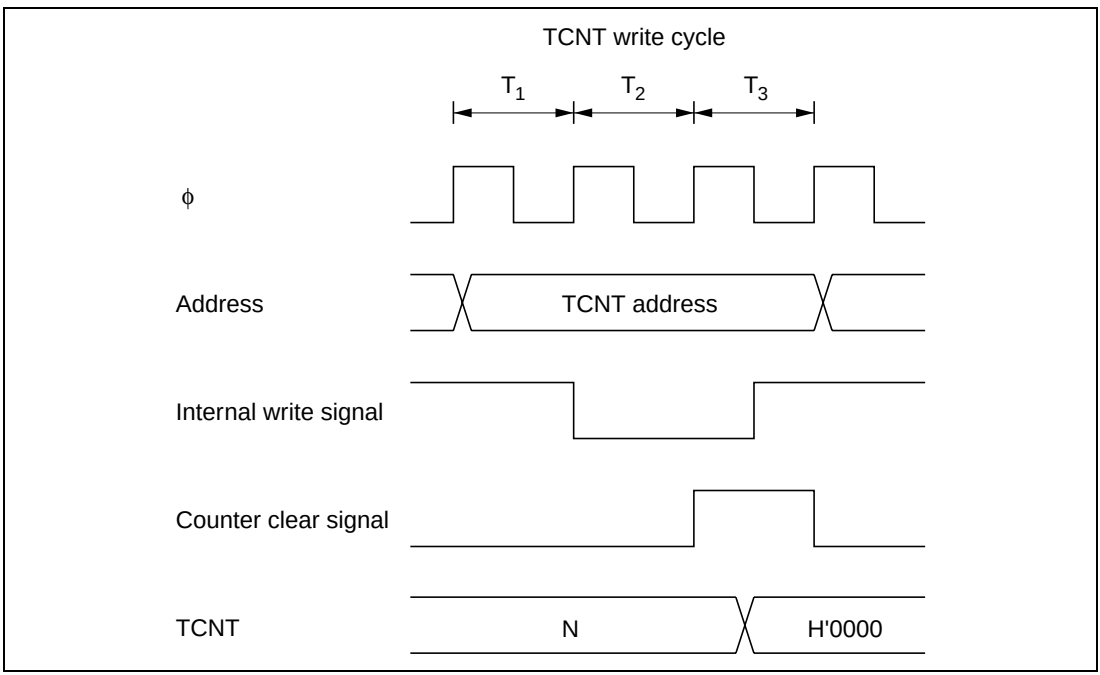


Figure 8.61 Contention between TCNT Write and Clear

Contention between TCNT Word Write and Increment

If an increment pulse occurs in the T_3 state of a TCNT word write cycle, writing takes priority and TCNT is not incremented. See figure 8.62.

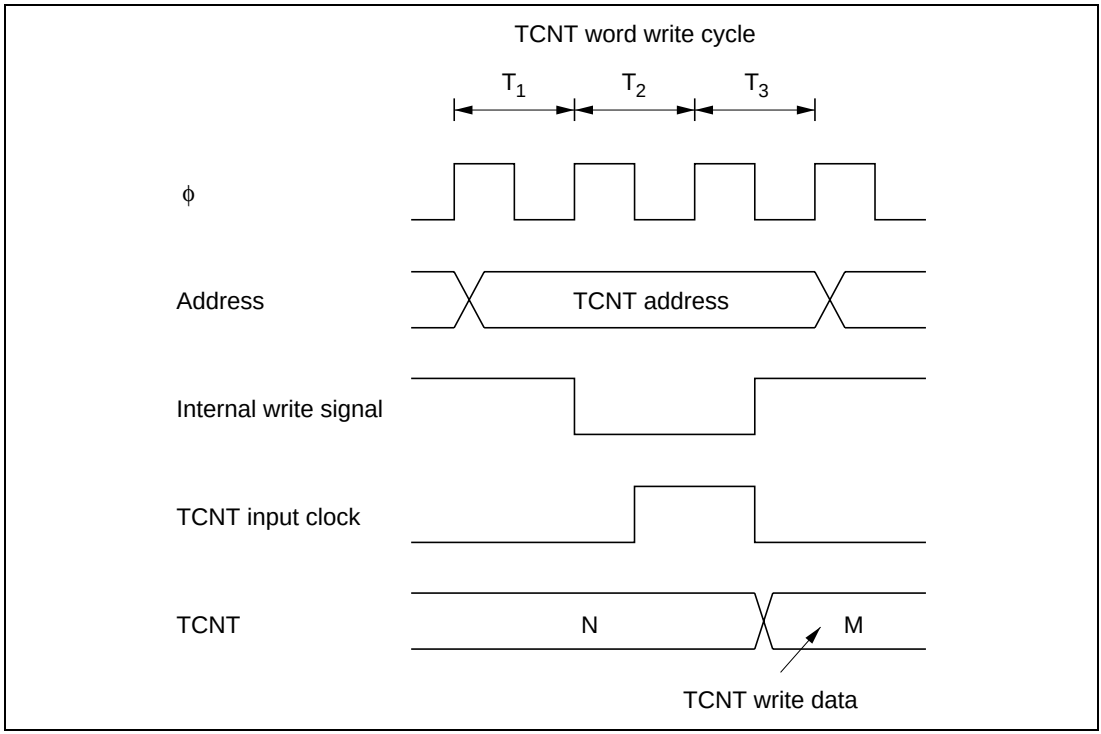


Figure 8.62 Contention between TCNT Word Write and Increment

Contention between TCNT Byte Write and Increment

If an increment pulse occurs in the T_2 or T_3 state of a TCNT byte write cycle, writing takes priority and TCNT is not incremented. The TCNT byte that was not written retains its previous value. See figure 8.63, which shows an increment pulse occurring in the T_2 state of a byte write to TCNTH.

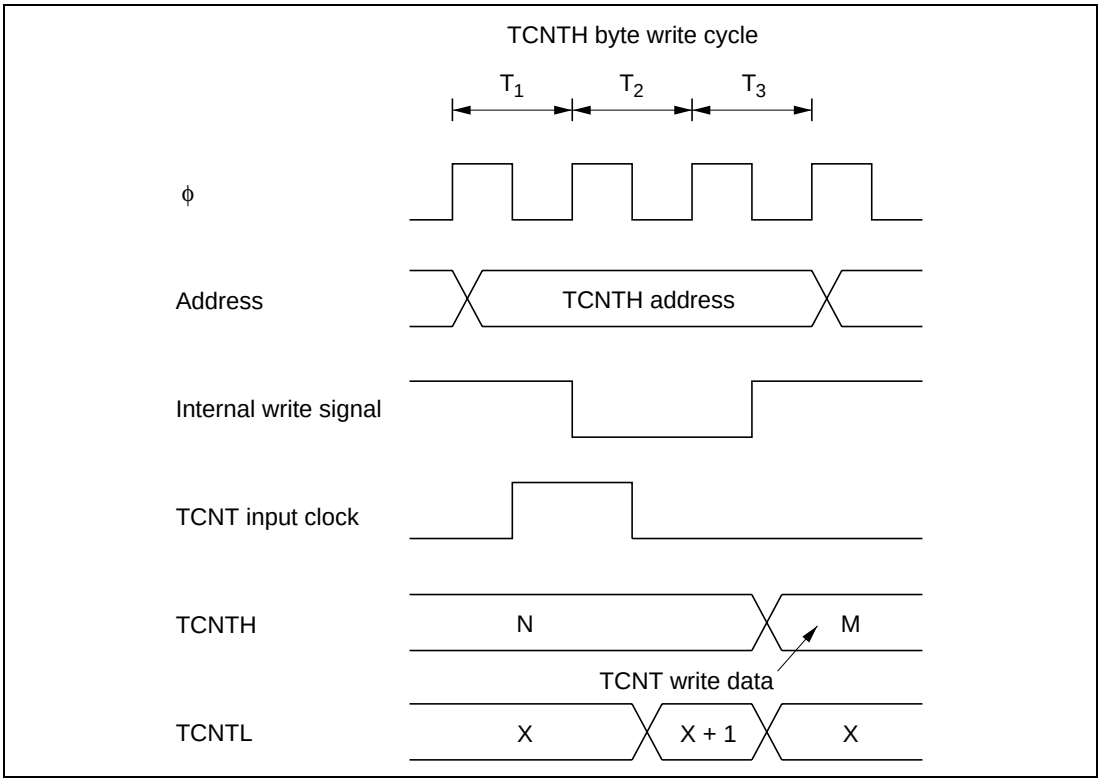


Figure 8.63 Contention between TCNT Byte Write and Increment

Contention between General Register Write and Compare Match

If a compare match occurs in the T_3 state of a general register write cycle, writing takes priority and the compare match signal is inhibited. See figure 8.64.

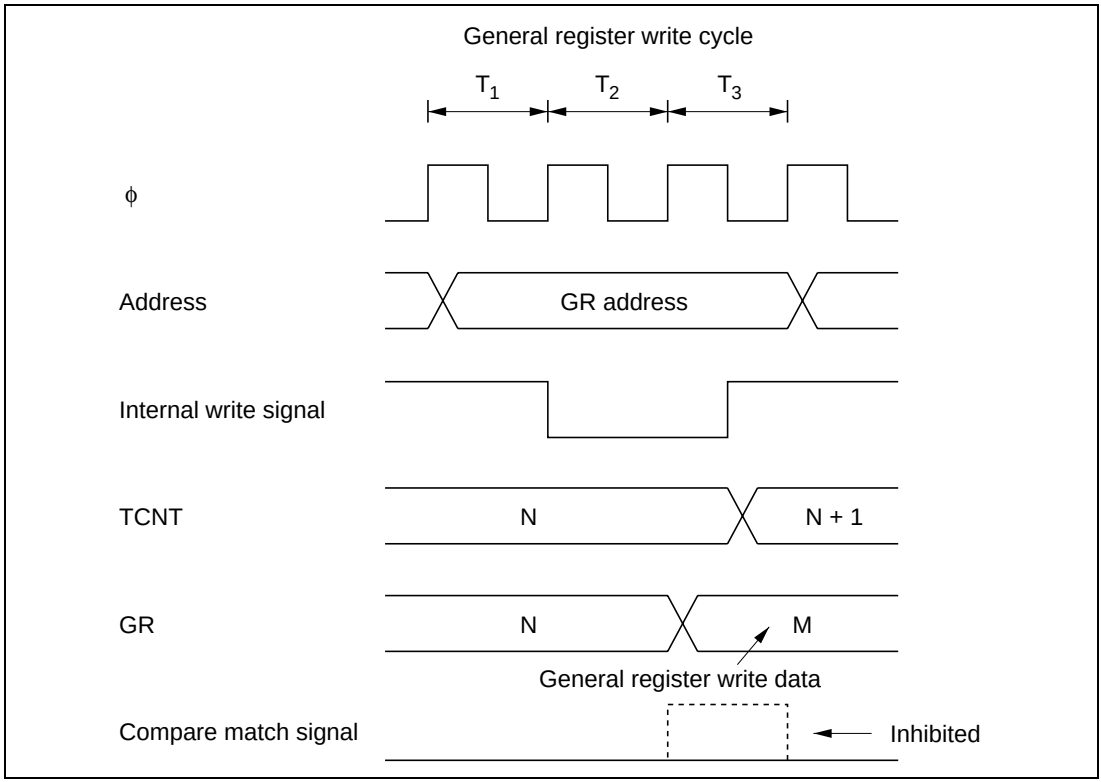


Figure 8.64 Contention between General Register Write and Compare Match

Contention between TCNT Write and Overflow or Underflow

If an overflow occurs in the T_3 state of a TCNT write cycle, writing takes priority and the counter is not incremented. OVF is set to 1. The same holds for underflow. See figure 8.65.

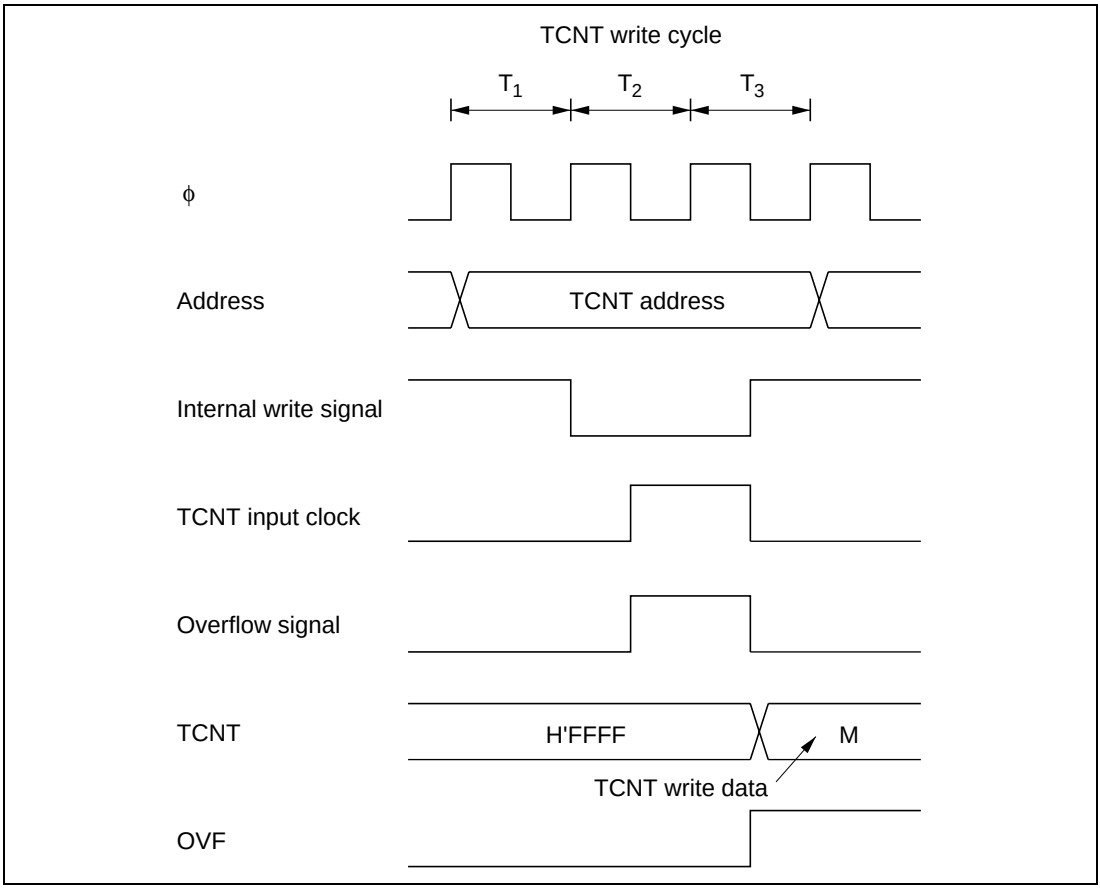


Figure 8.65 Contention between TCNT Write and Overflow

Contention between General Register Read and Input Capture

If an input capture signal occurs during the T_3 state of a general register read cycle, the value before input capture is read. See figure 8.66.

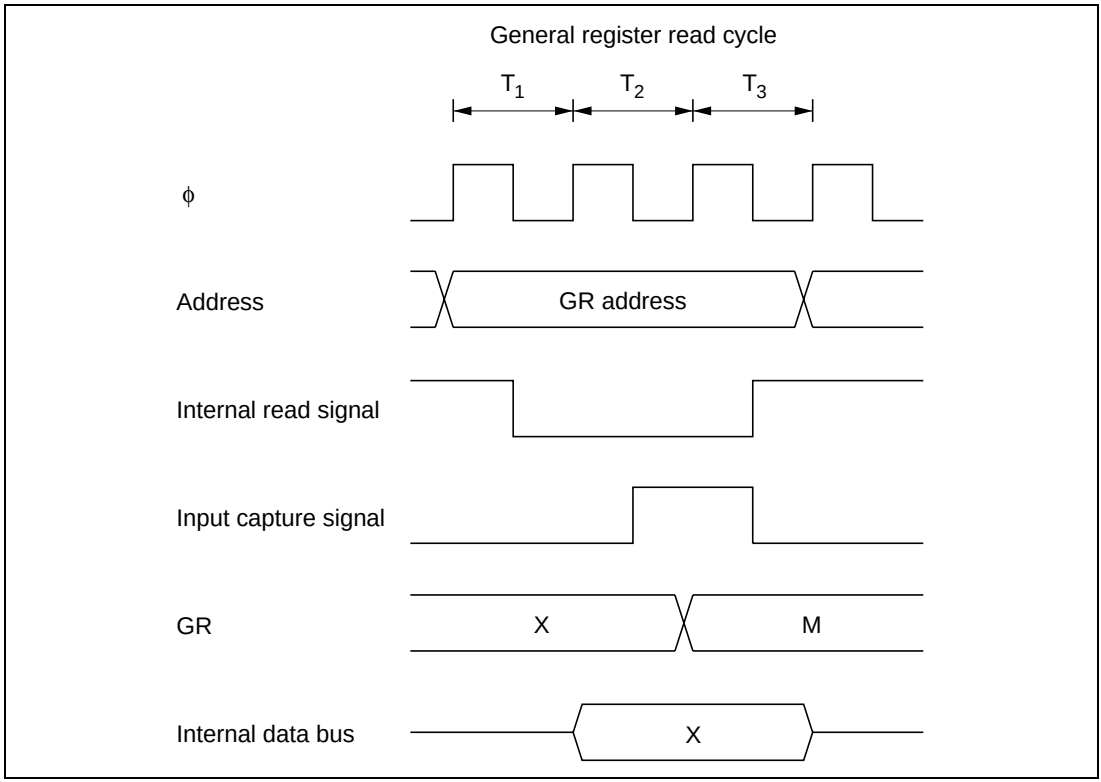


Figure 8.66 Contention between General Register Read and Input Capture

Contention between Counter Clearing by Input Capture and Counter Increment

If an input capture signal and counter increment signal occur simultaneously, the counter is cleared according to the input capture signal. The counter is not incremented by the increment signal. The value before the counter is cleared is transferred to the general register. See figure 8.67.

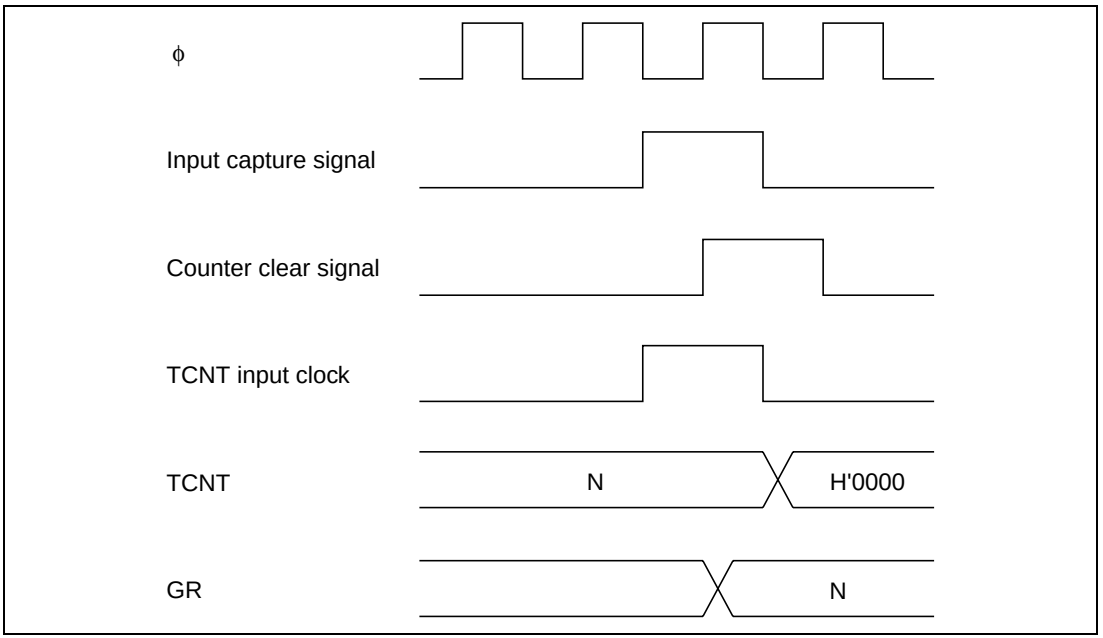


Figure 8.67 Contention between Counter Clearing by Input Capture and Counter Increment

Contention between General Register Write and Input Capture

If an input capture signal occurs in the T_3 state of a general register write cycle, input capture takes priority and the write to the general register is not performed. See figure 8.68.

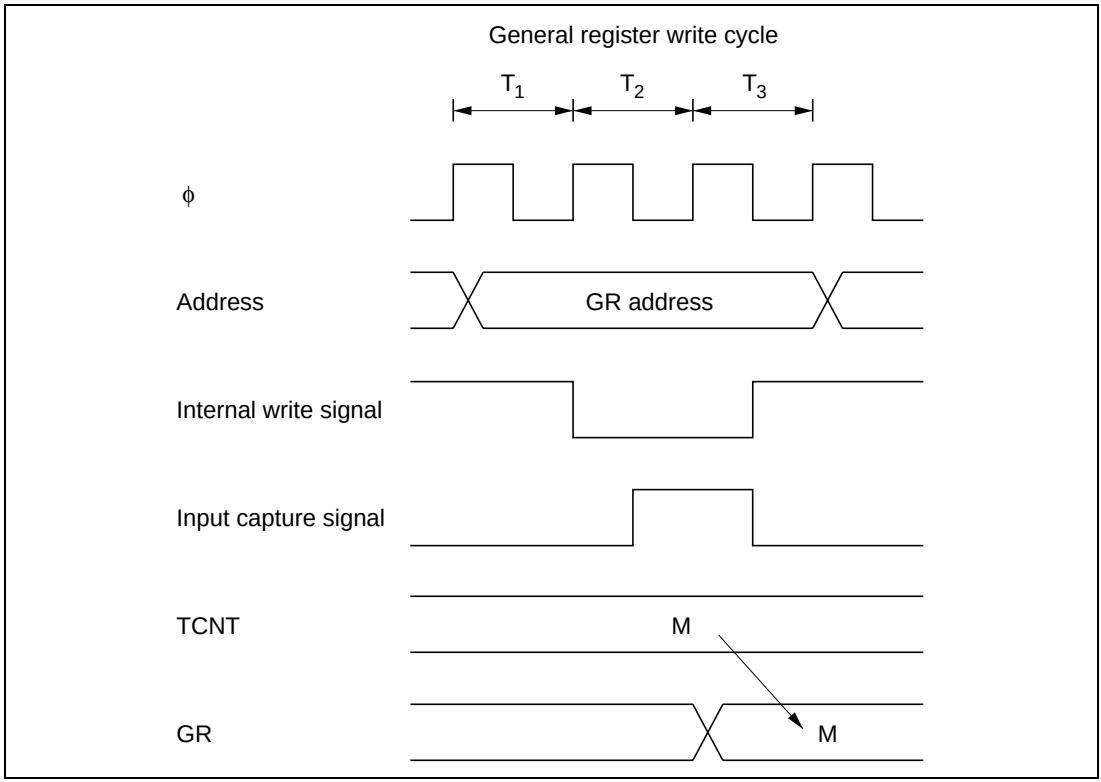


Figure 8.68 Contention between General Register Write and Input Capture

Note on Waveform Period Setting

When a counter is cleared by compare match, the counter is cleared in the last state at which the TCNT value matches the general register value, at the time when this value would normally be updated to the next count. The actual counter frequency is therefore given by the following formula:

$$f = \frac{\phi}{(N + 1)}$$

(f: counter frequency. ϕ: system clock frequency. N: value set in general register.)

Contention between Buffer Register Write and Input Capture

If a buffer register is used for input capture buffering and an input capture signal occurs in the T_3 state of a write cycle, input capture takes priority and the write to the buffer register is not performed. See figure 8.69.

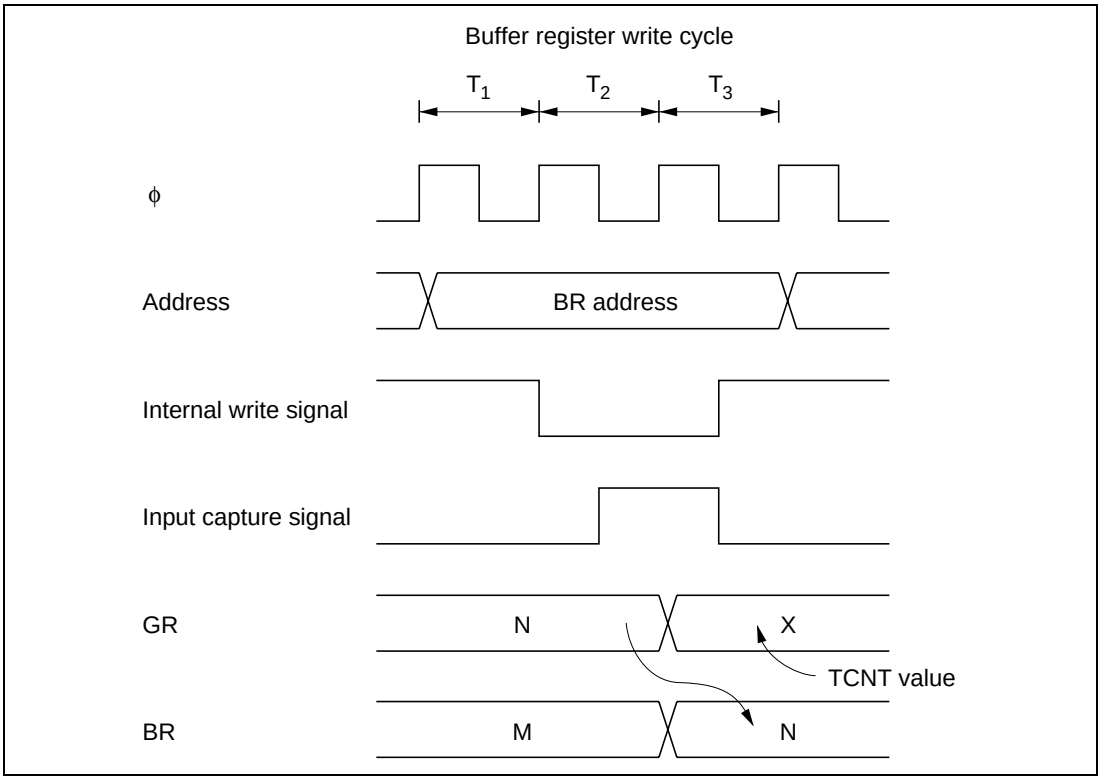


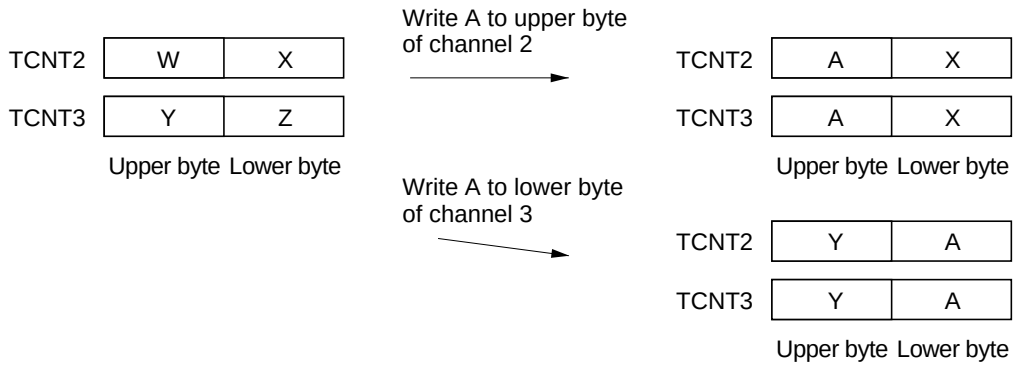
Figure 8.69 Contention between Buffer Register Write and Input Capture

Note on Synchronous Preset

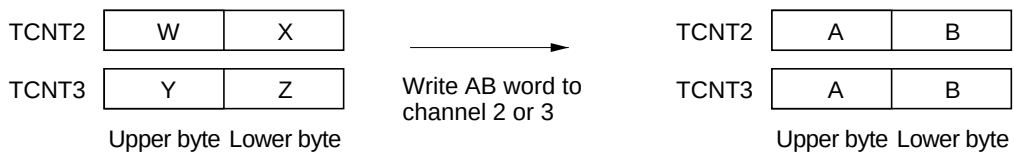
When channels are synchronized, if a TCNT value is modified by byte write access, all 16 bits of all synchronized counters assume the same value as the counter that was addressed.

(Example) When channels 2 and 3 are synchronized

- Byte write to channel 2 or byte write to channel 3



- Word write to channel 2 or word write to channel 3



Note on Setup of Reset-Synchronized PWM Mode and Complementary PWM Mode

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 complementary PWM mode. First switch to normal mode (by clearing bit CMD1 to 0), then select reset-synchronized PWM mode or complementary PWM mode.

ITU Operating Modes

Table 8.11 (a) ITU Operating Modes (Channel 0)

Operating Mode		Register Settings														
		TSNC	TMDR			TFCR			TOCR		TOER	TIOR0		TCR0		
		Synchro-nization	MDF	FDIR	PWM	Comple-mentary PWM	Reset-Synchro-nized PWM	Buffer-ing	XTGD	Output Level Select	Master Enable	IOA	IOB	Clear Select	Clock Select
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 unrestricted	◯	◯
Input capture A		◯	—	—	PWM0 = 0	—	—	—	—	—	—	IOA2 = 1 Other bits unrestricted	◯	◯	◯
Input capture B		◯	—	—	PWM0 = 0	—	—	—	—	—	—	◯	IOB2 = 1 Other bits unrestricted	◯	◯
Counter clearing	By compare match/input capture A	◯	—	—	◯	—	—	—	—	—	—	◯	◯	CCLR1 = 0 CCLR0 = 1	◯
	By compare match/input capture B	◯	—	—	◯	—	—	—	—	—	—	◯	◯	CCLR1 = 1 CCLR0 = 0	◯
	Syn-chronous clear	SYNC0 = 1	—	—	◯	—	—	—	—	—	—	◯	◯	CCLR1 = 1 CCLR0 = 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 match signal is inhibited.

Table 8.11 (b) ITU Operating Modes (Channel 1)

Operating Mode		Register Settings														
		TSNC	TMDR			TFCR			TOCR		TOER	TIOR1		TCR1		
		Synchro-nization	MDF	FDIR	PWM	Comple-mentary PWM	Reset-Synchro-nized PWM	Buffer-ing	XTGD	Output Level Select	Master Enable	IOA	IOB	Clear Select	Clock Select
Synchronous preset		SYNC1 = 1	—	—	○	—	—	—	—	—	—	○	○	○	○
PWM mode		○	—	—	PWM1 = 1	—	—	—	—	—	—	—	○*1	○	○
Output compare A		○	—	—	PWM1 = 0	—	—	—	—	—	—	IOA2 = 0 Other bits unrestricted	○	○	○
Output compare B		○	—	—	○	—	—	—	—	—	—	○	IOB2 = 0 Other bits unrestricted	○	○
Input capture A		○	—	—	PWM1 = 0	—	—	—	○*2	—	—	IOA2 = 1 Other bits unrestricted	○	○	○
Input capture B		○	—	—	PWM1 = 0	—	—	—	—	—	—	○	IOB2 = 1 Other bits unrestricted	○	○
Counter clearing	By compare match/input capture A	○	—	—	○	—	—	—	—	—	—	○	○	CCLR1 = 0 CCLR0 = 1	○
	By compare match/input capture B	○	—	—	○	—	—	—	—	—	—	○	○	CCLR1 = 1 CCLR0 = 0	○
	Syn-chronous clear	SYNC1 = 1	—	—	○	—	—	—	—	—	—	○	○	CCLR1 = 1 CCLR0 = 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 compare match signal is inhibited.
2. Valid only when channels 3 and 4 are operating in complementary PWM mode or reset-synchronized PWM mode.

Table 8.11 (c) ITU Operating Modes (Channel 2)

Operating Mode		Register Settings												
		TSNC	TMDR			TFCR			TOCR		TOER	TIOR2		TCR2	
		Synchro-nization	MDF	FDIR	PWM	Comple-mentary PWM	Reset-Synchro-nized PWM	Buffer-ing	XTGD	Output Level Select	Master Enable	IOA	IOB	Clear Select	Clock Select
Synchronous preset		SYNC2 = 1	○	—	○	—	—	—	—	—	—	○	○	○	○
PWM mode		○	○	—	PWM2 = 1	—	—	—	—	—	—	—	○*	○	○
Output compare A		○	○	—	PWM2 = 0	—	—	—	—	—	—	IOA2 = 0 Other bits unrestricted	○	○	○
Output compare B		○	○	—	○	—	—	—	—	—	—	○	IOB2 = 0 Other bits unrestricted	○	○
Input capture A		○	○	—	PWM2 = 0	—	—	—	—	—	—	IOA2 = 1 Other bits unrestricted	○	○	○
Input capture B		○	○	—	PWM2 = 0	—	—	—	—	—	—	○	IOB2 = 1 Other bits unrestricted	○	○
Counter clearing	By compare match/input capture A	○	○	—	○	—	—	—	—	—	—	○	○	CCLR1 = 0 CCLR0 = 1	○
	By compare match/input capture B	○	○	—	○	—	—	—	—	—	—	○	○	CCLR1 = 1 CCLR0 = 0	○
	Syn-chronous clear	SYNC2 = 1	○	—	○	—	—	—	—	—	—	○	○	CCLR1 = 1 CCLR0 = 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 compare match signal is inhibited.

Table 8.11 (d) ITU Operating Modes (Channel 3)

Operating Mode		Register Settings												
		TSNC	TMDR			TFCR		TOCR		TOER	TIOR3		TCR3	
		Synchro-nization	MDF	FDIR	PWM	Comple-mentary PWM	Reset-Synchro-nized PWM	Buffering	XTGD	Output Level Select	Master Enable	IOA	IOB	Clear Select	Clock Select
Synchronous preset		SYNC3 = 1	—	—	○	○*3	○	○	—	—	○*1	○	○	○	○
PWM mode		○	—	—	PWM3 = 1	CMD1 = 0	CMD1 = 0	○	—	—	○	—	○*2	○	○
Output compare A		○	—	—	PWM3 = 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		○	—	—	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	○	IOB2 = 1 Other bits unrestricted	○	○
Counter clearing	By compare match/input capture A	○	—	—	○	Illegal setting: CMD1 = 1 CMD0 = 0	○*4	○	—	—	○*1	○	○	CCLR1 = 0 CCLR0 = 1	○
	By compare match/input capture B	○	—	—	○	CMD1 = 0	CMD1 = 0	○	—	—	○*1	○	○	CCLR1 = 1 CCLR0 = 0	○
	Syn-chronous clear	SYNC3 = 1	—	—	○	Illegal setting: CMD1 = 1 CMD0 = 0	○	○	—	—	○*1	○	○	CCLR1 = 1 CCLR0 = 1	○
Complementary PWM mode		○*3	—	—	—	CMD1 = 1 CMD0 = 0	CMD1 = 1 CMD0 = 0	○	○*6	○	○	—	—	CCLR1 = 0 CCLR0 = 0	○*5
Reset-synchronized PWM mode		○	—	—	—	CMD1 = 1 CMD0 = 1	CMD1 = 1 CMD0 = 1	○	○*6	○	○	—	—	CCLR1 = 0 CCLR0 = 1	○
Buffering (BRA)		○	—	—	○	○	○	BFA3 = 1 Other bits unrestricted	—	—	○*1	○	○	○	○
Buffering (BRB)		○	—	—	○	○	○	BFB3 = 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.
2. The input capture function cannot be used in PWM mode. If compare match A and compare match B occur simultaneously, the compare match signal is inhibited.
3. Do not set both channels 3 and 4 for synchronous operation when complementary PWM mode is selected.
4. The counter cannot be cleared by input capture A when reset-synchronized PWM mode is selected.
5. In complementary PWM mode, select the same clock source for channels 3 and 4.
6. Use the input capture A function in channel 1.

Table 8.11 (e) ITU Operating Modes (Channel 4)

Operating Mode		Register Settings													
		TSNC	TMDR			TFCR			TOCR		TOER	TIOR4		TCR4	
		Synchro-nization	MDF	FDIR	PWM	Comple-mentary PWM	Reset-Synchro-nized PWM	Buffering	XTGD	Output Level Select	Master Enable	IOA	IOB	Clear Select	Clock Select
Synchronous preset		SYNC4 = 1	—	—	○	○*3	○	○	—	—	○*1	○	○	○	○
PWM mode		○	—	—	PWM4 = 1	CMD1 = 0	CMD1 = 0	○	—	—	○	—	○*2	○	○
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	○	○	CCLR1 = 0 CCLR0 = 1	○
	By compare match/input capture B	○	—	—	○	Illegal setting: CMD1 = 1 CMD0 = 0	○*4	○	—	—	○*1	○	○	CCLR1 = 1 CCLR0 = 0	○
	Syn-chronous clear	SYNC4 = 1	—	—	○	Illegal setting: CMD1 = 1 CMD0 = 0	○*4	○	—	—	○*1	○	○	CCLR1 = 1 CCLR0 = 1	○
Complementary PWM mode		○*3	—	—	—	CMD1 = 1 CMD0 = 0	CMD1 = 1 CMD0 = 0	○	○	○	○	—	—	CCLR1 = 0 CCLR0 = 0	○*5
Reset-synchronized PWM mode		○	—	—	—	CMD1 = 1 CMD0 = 1	CMD1 = 1 CMD0 = 1	○	○	○	○	—	—	○*6	○*6
Buffering (BRA)		○	—	—	○	○	○	BFA4 = 1 Other bits unrestricted	—	—	○*1	○	○	○	○
Buffering (BRB)		○	—	—	○	○	○	BFB4 = 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.
2. The input capture function cannot be used in PWM mode. If compare match A and compare match B occur simultaneously, the compare match signal is inhibited.
3. Do not set both channels 3 and 4 for synchronous operation when complementary PWM mode is selected.
4. When reset-synchronized PWM mode is selected, TCNT4 operates independently and the counter clearing function is available. W aveform output is not affected.
5. In complementary PWM mode, select the same clock source for channels 3 and 4.
6. TCR4 settings are valid in reset-synchronized PWM mode, but TCNT4 operates independently , without affecting waveform output.

Section 9 Programmable Timing Pattern Controller

9.1 Overview

The H8/3039 Group has a built-in programmable timing pattern controller (TPC)* that provides pulse outputs by using the 16-bit integrated timer-pulse unit (ITU) as a time base. The TPC pulse outputs are divided into 4-bit groups (group 3 to group 0) that can operate simultaneously and independently.

9.1.1 Features

TPC features are listed below.

- 15-bit output data
  Maximum 15-bit data can be output. TPC output can be enabled on a bit-by-bit basis.
- Four output groups and one 3-bit output.
  Output trigger signals can be selected in 4-bit groups to provide up to three different 4-bit outputs and one 3-bit output.
- Selectable output trigger signals
  Output trigger signals can be selected for each group from the compare-match signals of four ITU channels.
- Non-overlap mode
  A non-overlap margin can be provided between pulse outputs.

Note: * Note that since this LSI does not have a TP_{14} pin, it is a 15-bit programmable timing pattern controller (TPC).

9.1.2 Block Diagram

Figure 9.1 shows a block diagram of the TPC.

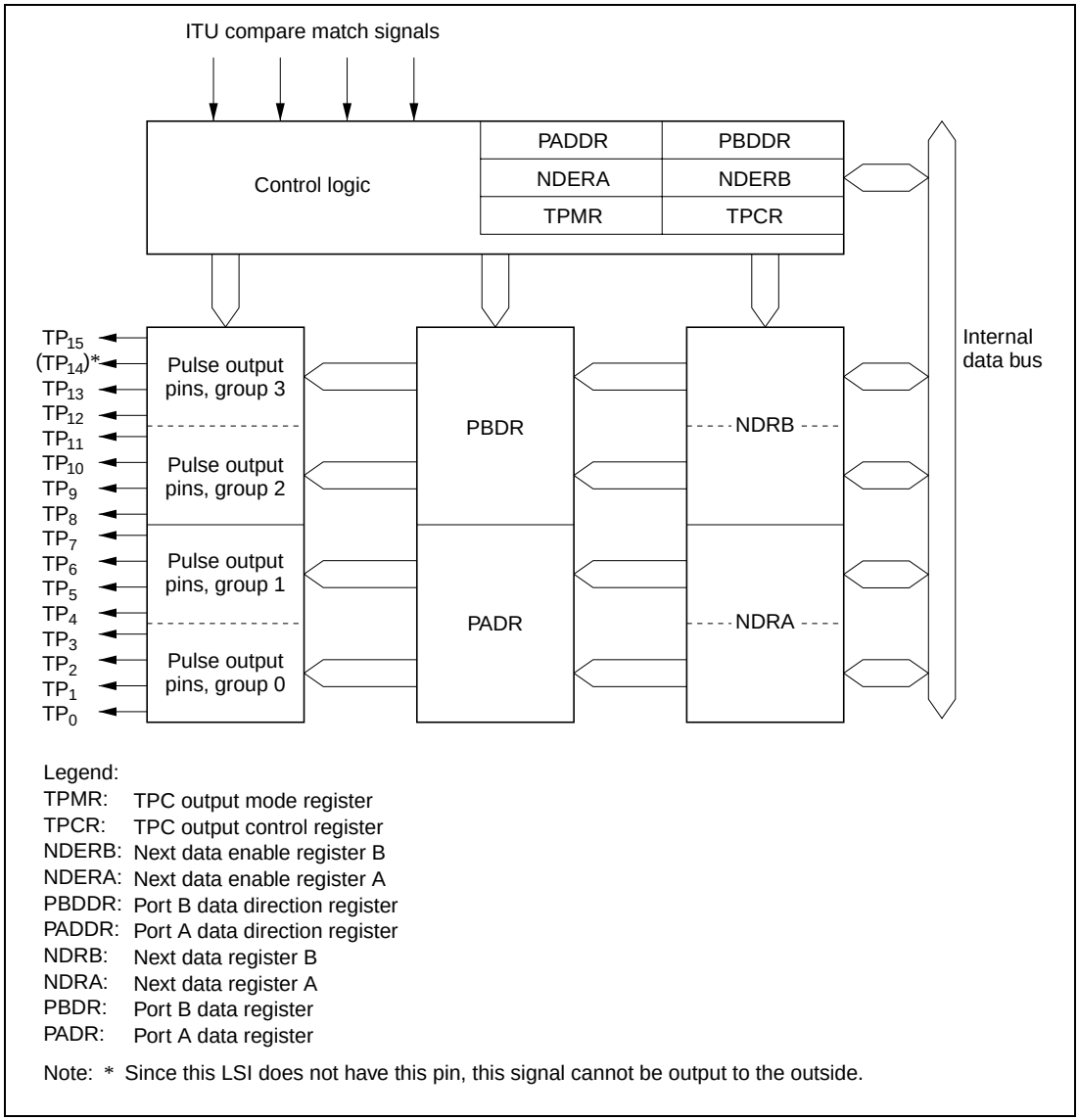


Figure 9.1 TPC Block Diagram

9.1.3 TPC Pins

Table 9.1 summarizes the TPC output pins.

Table 9.1 TPC Pins

Name	Symbol	I/O	Function
TPC output 0	TP_0	Output	Group 0 pulse output
TPC output 1	TP_1	Output	
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	

Note: * Since this LSI does not have this pin, this signal cannot be output to the outside.

9.1.4 Registers

Table 9.2 summarizes the TPC registers.

Table 9.2 TPC Registers

Address*[1]	Name	Abbreviation	R/W	Initial Value
H'FFD1	Port A data direction register	PADDR	W	H'00
H'FFD3	Port A data register	PADR	R/(W)*[2]	H'00
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 TPC output groups 0 and 1 by settings in TPCR. When the output triggers are different, the NDRA address is H'FFA7 for group 0 and H'FFA5 for group 1. Similarly, the address of NDRB is H'FFA4 when the same output trigger is selected for TPC output groups 2 and 3 by settings in TPCR. When the output triggers are different, the NDRB address is H'FFA6 for group 2 and H'FFA4 for group 3.

9.2 Register Descriptions

9.2.1 Port A Data Direction Register (PADDR)

PADDR is an 8-bit write-only register that selects input or output for each pin in port A.

Bit	7	6	5	4	3	2	1	0
	PA_7DDR	PA_6DDR	PA_5DDR	PA_4DDR	PA_3DDR	PA_2DDR	PA_1DDR	PA_0DDR
Initial value	0	0	0	0	0	0	0	0
Read/Write	W	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 output must be set to 1. For further information about PADDR, see section 7.10, Port A.

9.2.2 Port A Data Register (PADR)

PADR is an 8-bit readable/writable register that stores TPC output data for groups 0 and 1, when these TPC output groups are used.

Bit	7	6	5	4	3	2	1	0
	PA_7	PA_6	PA_5	PA_4	PA_3	PA_2	PA_1	PA_0
Initial value	0	0	0	0	0	0	0	0
Read/Write	R/(W)*	R/(W)*	R/(W)*	R/(W)*	R/(W)*	R/(W)*	R/(W)*	R/(W)*

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 7.10, Port A.

9.2.3 Port B Data Direction Register (PBDDR)

PBDDR is an 8-bit write-only register that selects input or output for each pin in port B.

Bit	7	6	5	4	3	2	1	0
	PB_7DDR	—	PB_5DDR	PB_4DDR	PB_3DDR	PB_2DDR	PB_1DDR	PB_0DDR
Initial value	0	0	0	0	0	0	0	0
Read/Write	W	W	W	W	W	W	W	W

Reserved bit

Port B data direction 7, 5 to 0
These bits select input or output for port B pins

Port B is multiplexed with pins TP_{15}, TP_{13} to TP_8. Bits corresponding to pins used for TPC output must be set to 1. For further information about PBDDR, see section 7.11, Port B.

9.2.4 Port B Data Register (PBDR)

PBDR is an 8-bit readable/writable register that stores TPC output data for groups 2 and 3, when these TPC output groups are used.

Bit	7	6	5	4	3	2	1	0
	PB_7	—	PB_5	PB_4	PB_3	PB_2	PB_1	PB_0
Initial value	0	0	0	0	0	0	0	0
Read/Write	R/(W)*	R/(W)*	R/(W)*	R/(W)*	R/(W)*	R/(W)*	R/(W)*	R/(W)*

Reserved bit

Port B data 7, 5 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 7.11, Port B.

9.2.5 Next Data Register A (NDRA)

NDRA is an 8-bit readable/writable register that stores the next output data for TPC output groups 1 and 0 (pins TP_7 to TP_0). During TPC output, when an ITU compare match event specified in TPCR occurs, NDRA contents are transferred to the corresponding bits in PADR. The address of NDRA differs depending on whether TPC output groups 0 and 1 have the same output trigger or different output triggers.

NDRA is initialized to H'00 by a reset and in hardware standby mode. It is not initialized in 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 NDRA address is H'FFA5. The upper 4 bits belong to group 1 and the lower 4 bits to group 0. Address H'FFA7 consists entirely of reserved bits that cannot be modified and always read 1.

Address H'FFA5

Bit	7	6	5	4	3	2	1	0
	NDR7	NDR6	NDR5	NDR4	NDR3	NDR2	NDR1	NDR0
Initial value	0	0	0	0	0	0	0	0
Read/Write	R/W	R/W	R/W	R/W	R/W	R/W	R/W	R/W

Next data 7 to 4
These bits store the next output data for TPC output group 1

Next data 3 to 0
These bits store the next output data for TPC output group 0

Address H'FFA7

Bit	7	6	5	4	3	2	1	0
	—	—	—	—	—	—	—	—
Initial value	1	1	1	1	1	1	1	1
Read/Write	—	—	—	—	—	—	—	—

Reserved bits

Different Triggers for TPC Output Groups 0 and 1

If TPC output groups 0 and 1 are triggered by different compare match events, the address of the upper 4 bits of NDRA (group 1) is H'FFA5 and the address of the lower 4 bits (group 0) is H'FFA7. Bits 3 to 0 of address H'FFA5 and bits 7 to 4 of address H'FFA7 are reserved bits that cannot be modified and always read 1.

Address H'FFA5

Bit	7	6	5	4	3	2	1	0
	NDR7	NDR6	NDR5	NDR4	—	—	—	—
Initial value	0	0	0	0	1	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	0
	—	—	—	—	NDR3	NDR2	NDR1	NDR0
Initial value	1	1	1	1	0	0	0	0
Read/Write	—	—	—	—	R/W	R/W	R/W	R/W

Reserved bits

Next data 3 to 0
These bits store the next output data for TPC output group 0

9.2.6 Next Data Register B (NDRB)

NDRB is an 8-bit readable/writable register that stores the next output data for TPC output groups 3 and 2 (pins TP_{15} to TP_8)*. During TPC output, when an ITU compare match event specified in TPCR occurs, NDRB contents are transferred to the corresponding bits in PBDR. The address of NDRB differs depending on whether TPC output groups 2 and 3 have the same output trigger or different output triggers.

NDRB is initialized to H'00 by a reset and in hardware standby mode. It is not initialized in software standby mode.

Note: * Since this LSI does not have a TP_{14} pin, the TP_{14} signal cannot be output to the outside.

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 NDRB address is H'FFA4. The upper 4 bits belong to group 3 and the lower 4 bits to group 2. Address H'FFA6 consists entirely of reserved bits that cannot be modified and always read 1.

Address H'FFA4

Bit	7	6	5	4	3	2	1	0
	NDR15	NDR14	NDR13	NDR12	NDR11	NDR10	NDR9	NDR8
Initial value	0	0	0	0	0	0	0	0
Read/Write	R/W	R/W	R/W	R/W	R/W	R/W	R/W	R/W

Next data 15 to 12
These bits store the next output data for TPC output group 3

Next data 11 to 8
These bits store the next output data for TPC output group 2

Address H'FFA6

Bit	7	6	5	4	3	2	1	0
	—	—	—	—	—	—	—	—
Initial value	1	1	1	1	1	1	1	1
Read/Write	—	—	—	—	—	—	—	—

Reserved bits

Different Triggers for TPC Output Groups 2 and 3

If TPC output groups 2 and 3 are triggered by different compare match events, the address of the upper 4 bits of NDRB (group 3)* is H'FFA4 and the address of the lower 4 bits (group 2) is H'FFA6. Bits 3 to 0 of address H'FFA4 and bits 7 to 4 of address H'FFA6 are reserved bits that cannot be modified and always read 1.

Note: * Since this LSI does not have a TP_{14} pin, the TP_{14} signal cannot be output off-chip.

Address H'FFA4

Bit	7	6	5	4	3	2	1	0
	NDR15	NDR14	NDR13	NDR12	—	—	—	—
Initial value	0	0	0	0	1	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	0
	—	—	—	—	NDR11	NDR10	NDR9	NDR8
Initial value	1	1	1	1	0	0	0	0
Read/Write	—	—	—	—	R/W	R/W	R/W	R/W

Reserved bits

Next data 11 to 8
These bits store the next output data for TPC output group 2

9.2.7 Next Data Enable Register A (NDERA)

NDERA is an 8-bit readable/writable register that enables or disables TPC output groups 1 and 0 (TP_7 to TP_0) on a bit-by-bit basis.

Bit	7	6	5	4	3	2	1	0
	NDER7	NDER6	NDER5	NDER4	NDER3	NDER2	NDER1	NDER0
Initial value	0	0	0	0	0	0	0	0
Read/Write	R/W	R/W	R/W	R/W	R/W	R/W	R/W	R/W

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 event selected in the TPC output control register (TPCR) occurs, the NDRA value is automatically transferred to the corresponding PADR bit, updating the output value. If TPC output is disabled, the bit value is 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 initialized in software standby mode.

Bits 7 to 0—Next Data Enable 7 to 0 (NDER7 to NDER0): These bits enable or disable TPC 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 (NDR_7 to NDR0 are not transferred to PA_7 to PA_0)	(Initial value)
1	TPC outputs TP_7 to TP_0 are enabled (NDR_7 to NDR_0 are transferred to PA_7 to PA_0)	

9.2.8 Next Data Enable Register B (NDERB)

NDERB is an 8-bit readable/writable register that enables or disables TPC output groups 3 and 2 (TP_{15} to TP_8)* on a bit-by-bit basis.

Bit	7	6	5	4	3	2	1	0
	NDER15	NDER14	NDER13	NDER12	NDER11	NDER10	NDER9	NDER8
Initial value	0	0	0	0	0	0	0	0
Read/Write	R/W	R/W	R/W	R/W	R/W	R/W	R/W	R/W

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 selected in the TPC output control register (TPCR) occurs, the NDRB value is automatically transferred to the corresponding PBDR bit, updating the output value. If TPC output is disabled, the bit value is not 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 initialized in software standby mode.

Bits 7 to 0—Next Data Enable 15 to 8 (NDER15 to NDER8): These bits enable or disable TPC output groups 3 and 2 (TP_{15} to TP_8)* on a bit-by-bit basis.

Bits 7 to 0 NDER15 to NDER8	Description	
0	TPC outputs TP_{15} to TP_8 are disabled (NDR15 to NDR8 are not transferred to PB_7 to PB_0)	(Initial value)
1	TPC outputs TP_{15} to TP_8 are enabled (NDR15 to NDR8 are transferred to PB_7 to PB_0)	

Note: * Since this LSI does not have a TP_{14} pin, the TP_{14} signal cannot be output to the outside.

9.2.9 TPC Output Control Register (TPCR)

TPCR is an 8-bit readable/writable register that selects output trigger signals for TPC outputs on a group-by-group basis.

Bit	7	6	5	4	3	2	1	0
	G3CMS1	G3CMS0	G2CMS1	G2CMS0	G1CMS1	G1CMS0	G0CMS1	G0CMS0
Initial value	1	1	1	1	1	1	1	1
Read/Write	R/W	R/W	R/W	R/W	R/W	R/W	R/W	R/W

Group 3 compare match select 1 and 0
These bits select the compare match event that triggers TPC output group 3 (TP_{15} to TP_{12})*

Group 2 compare match select 1 and 0
These bits select the compare match event that triggers TPC output group 2 (TP_{11} to TP_8)

Group 1 compare match select 1 and 0
These bits select the compare match event that triggers TPC output group 1 (TP_7 to TP_4)

Group 0 compare match select 1 and 0
These bits select the compare match event that triggers TPC output group 0 (TP_3 to TP_0)

Note: * Since this LSI does not have a TP_{14} pin, the TP_{14} signal cannot be output to the outside.

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

Bits 7 and 6—Group 3 Compare Match Select 1 and 0 (G3CMS1, G3CMS0): These bits select the compare match event that triggers TPC output group 3 (TP_{15} to TP_{12})*.

Bit 7 G3CMS1	Bit6 G3CMS0	Description
0	0	TPC output group 3 (TP_{15} to TP_{12})* is triggered by compare match in ITU channel 0
	1	TPC output group 3 (TP_{15} to TP_{12})* is triggered by compare match in ITU channel 1
1	0	TPC output group 3 (TP_{15} to TP_{12})* is triggered by compare match in ITU channel 2
	1	TPC output group 3 (TP_{15} to TP_{12})* is triggered by compare match in ITU channel 3 (Initial value)

Note: * Since this LSI does not have a TP_{14} pin, the TP_{14} signal cannot be output off-chip.

Bits 5 and 4—Group 2 Compare Match Select 1 and 0 (G2CMS1, G2CMS0): These bits select the compare match event that triggers TPC output group 2 (TP_{11} to TP_{8}).

Bit 5 G2CMS1	Bit4 G2CMS0	Description
0	0	TPC output group 2 (TP_{11} to TP_{8}) is triggered by compare match in ITU channel 0
	1	TPC output group 2 (TP_{11} to TP_{8}) is triggered by compare match in ITU channel 1
1	0	TPC output group 2 (TP_{11} to TP_{8}) is triggered by compare match in ITU channel 2
	1	TPC output group 2 (TP_{11} to TP_{8}) is triggered by compare match in ITU channel 3 (Initial value)

Bits 3 and 2—Group 1 Compare Match Select 1 and 0 (G1CMS1, G1CMS0): These bits select the compare match event that triggers TPC output group 1 (TP_7 to TP_4).

Bit 3 G1CMS1	Bit2 G1CMS0	Description
0	0	TPC output group 1 (TP_7 to TP_4) is triggered by compare match in ITU channel 0
	1	TPC output group 1 (TP_7 to TP_4) is triggered by compare match in ITU channel 1
1	0	TPC output group 1 (TP_7 to TP_4) is triggered by compare match in ITU channel 2
	1	TPC output group 1 (TP_7 to TP_4) is triggered by compare match in ITU channel 3 (Initial value)

Bits 1 and 0—Group 0 Compare Match Select 1 and 0 (G0CMS1, G0CMS0): These bits select the compare match event that triggers TPC output group 0 (TP_3 to TP_0).

Bit1 G0CMS1	Bit0 G0CMS0	Description
0	0	TPC output group 0 (TP_3 to TP_0) is triggered by compare match in ITU channel 0
	1	TPC output group 0 (TP_3 to TP_0) is triggered by compare match in ITU channel 1
1	0	TPC output group 0 (TP_3 to TP_0) is triggered by compare match in ITU channel 2
	1	TPC output group 0 (TP_3 to TP_0) is triggered by compare match in ITU channel 3 (Initial value)

9.2.10 TPC Output Mode Register (TPMR)

TPMR is an 8-bit readable/writable register that selects normal or non-overlapping TPC output for each group.

Bit	7	6	5	4	3	2	1	0
	—	—	—	—	G3NOV	G2NOV	G1NOV	G0NOV
Initial value	1	1	1	1	0	0	0	0
Read/Write	—	—	—	—	R/W	R/W	R/W	R/W

Reserved bits

Group 3 non-overlap
Selects non-overlapping TPC output for group 3 (TP_{15} to TP_{12})*

Group 2 non-overlap
Selects non-overlapping TPC output for group 2 (TP_{11} to TP_{8})

Group 1 non-overlap
Selects non-overlapping TPC output for group 1 (TP_{7} to TP_{4})

Group 0 non-overlap
Selects non-overlapping TPC output for group 0 (TP_{3} to TP_{0})

Note: * Since this LSI does not have a TP_{14} pin, the TP_{14} signal cannot be output to the outside.

The output trigger period of a non-overlapping TPC output waveform is set in general register B (GRB) in the ITU channel selected for output triggering. The non-overlap margin is set in general register A (GRA). The output values change at compare match A and B. For details see section 9.3.4, Non-Overlapping TPC Output.

TPMR is initialized to H'F0 by a reset and in hardware standby mode. It is not initialized in software standby mode.

Bits 7 to 4—Reserved: These bits cannot be modified and are always read as 1.

Bit 3—Group 3 Non-Overlap (G3NOV): Selects normal or non-overlapping TPC output for group 3 (TP_{15} to TP_{12})*.

Note: * Since this LSI does not have a TP_{14} pin, the TP_{14} signal cannot be output off-chip.

Bit 3 G3NOV	Description
0	Normal TPC output in group 3 (output values change at compare match A in the selected ITU channel) (Initial value)
1	Non-overlapping TPC output in group 3 (independent 1 and 0 output at compare match A and B in the selected ITU channel)

Bit 2—Group 2 Non-Overlap (G2NOV): Selects normal or non-overlapping TPC output for group 2 (TP_{11} to TP_{8}).

Bit 2 G2NOV	Description
0	Normal TPC output in group 2 (output values change at compare match A in the selected ITU channel) (Initial value)
1	Non-overlapping TPC output in group 2 (independent 1 and 0 output at compare match A and B in the selected ITU channel)

Bit 1—Group 1 Non-Overlap (G1NOV): Selects normal or non-overlapping TPC output for group 1 (TP_{7} to TP_{4}).

Bit 1 G1NOV	Description
0	Normal TPC output in group 1 (output values change at compare match A in the selected ITU channel) (Initial value)
1	Non-overlapping TPC output in group 1 (independent 1 and 0 output at compare match A and B in the selected ITU channel)

Bit 0—Group 0 Non-Overlap (G0NOV): Selects normal or non-overlapping TPC output for group 0 (TP_3 to TP_0).

Bit 0 G0NOV	Description
0	Normal TPC output in group 0 (output values change at compare match A in the selected ITU channel) (Initial value)
1	Non-overlapping TPC output in group 0 (independent 1 and 0 output at compare match A and B in the selected ITU channel)

9.3 Operation

9.3.1 Overview

When corresponding bits in PADDR or PBDDR and NDERA or NDERB are set to 1, TPC output is enabled. The TPC output initially consists of the corresponding PADR or PBDR contents. When a compare-match event selected in TPCR occurs, the corresponding NDRA or NDRB bit contents are transferred to PADR or PBDR to update the output values.

Figure 9.2 illustrates the TPC output operation. Table 9.3 summarizes the TPC operating conditions.

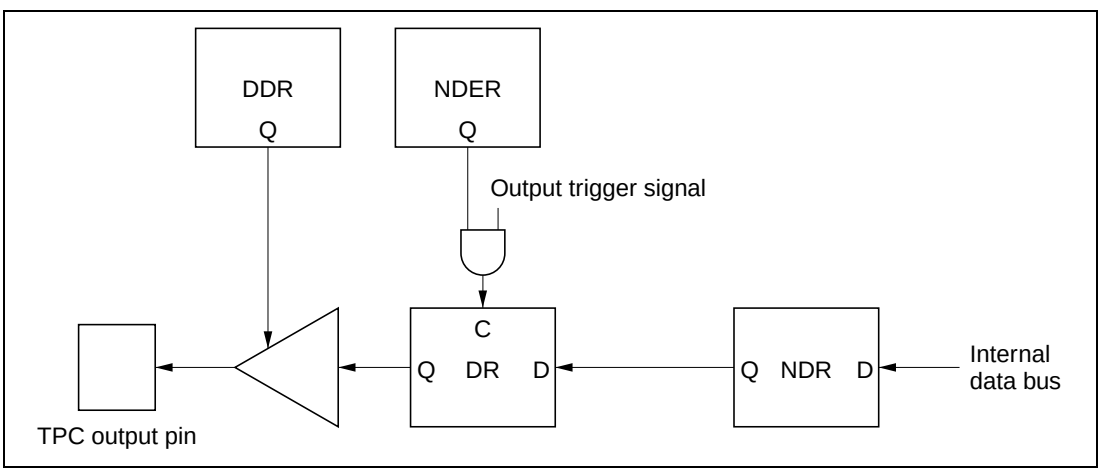


Figure 9.2 TPC Output Operation

Table 9.3 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 compare 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 NDRA and NDRB before the next compare match. For information on non-overlapping operation, see section 9.3.4, Non-Overlapping TPC Output.

9.3.2 Output Timing

If TPC output is enabled, NDRA/NDRB contents are transferred to PADR/PBDR and output when the selected compare match event occurs. Figure 9.3 shows the timing of these operations for the case of normal output in groups 0 and 1, triggered by compare match A.

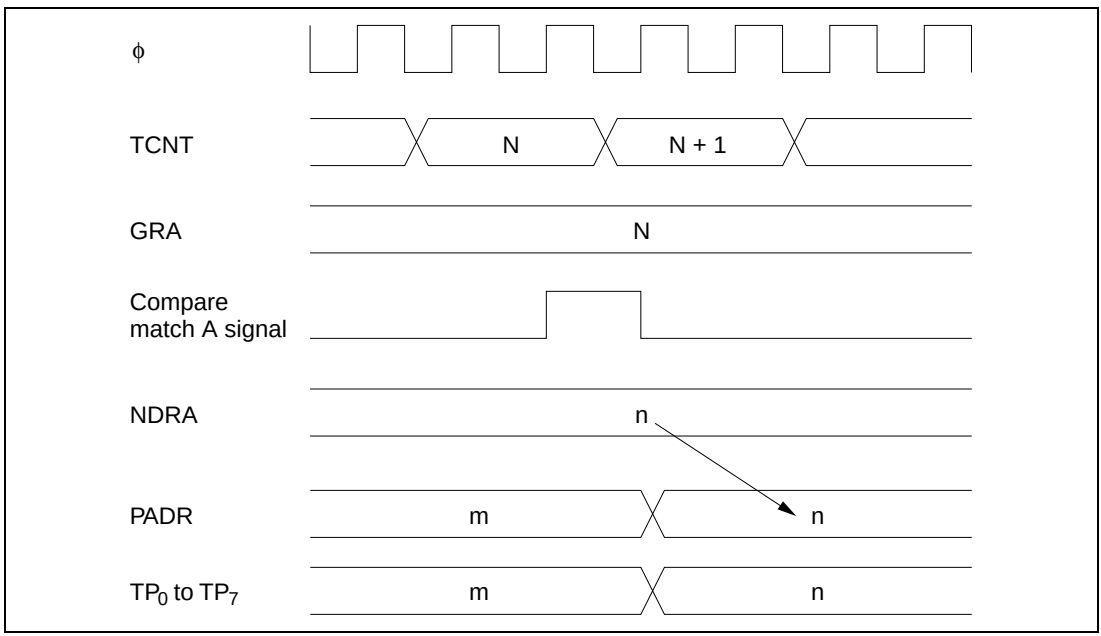


Figure 9.3 Timing of Transfer of Next Data Register Contents and Output (Example)

9.3.3 Normal TPC Output

Sample Setup Procedure for Normal TPC Output

Figure 9.4 shows a sample procedure for setting up normal TPC output.

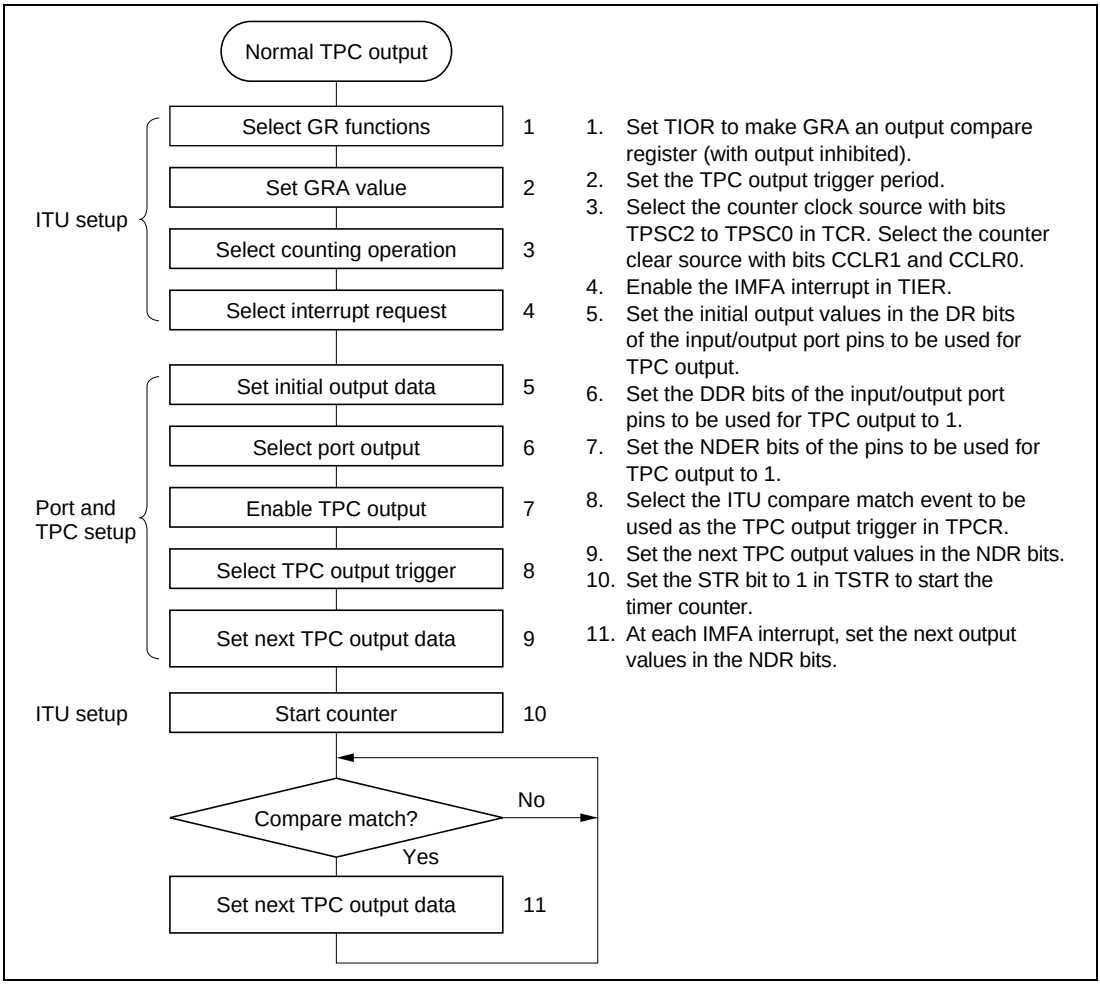
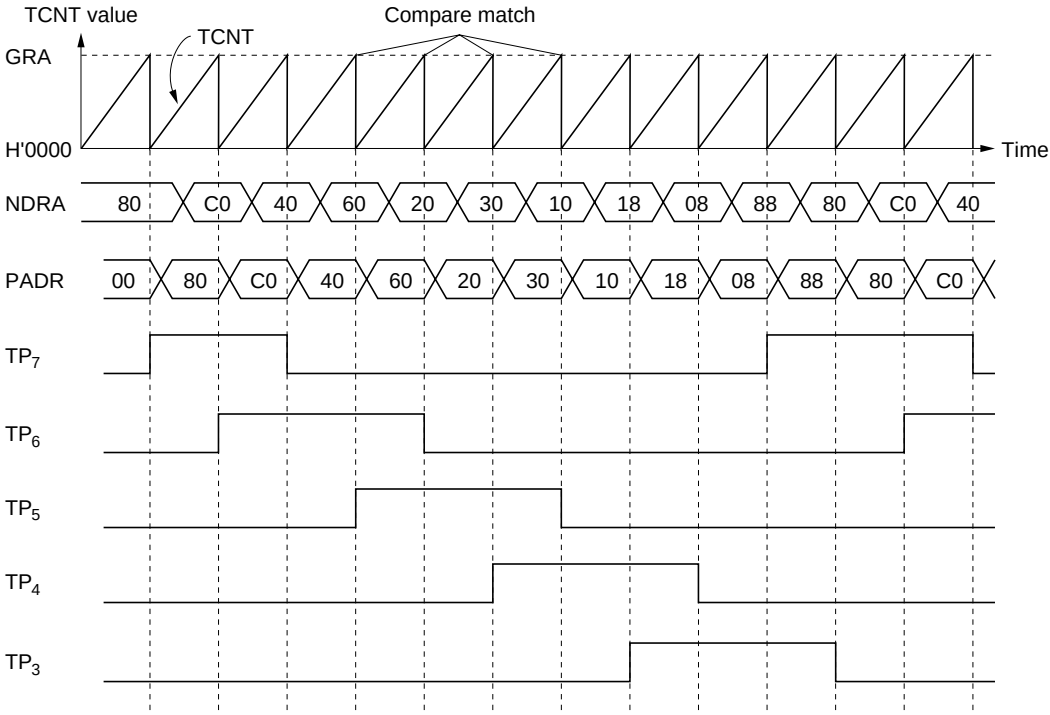


Figure 9.4 Setup Procedure for Normal TPC Output (Example)

Example of Normal TPC Output (Example of Five-Phase Pulse Output)

Figure 9.5 shows an example in which the TPC is used for cyclic five-phase pulse output.



- The ITU channel to be used as the output trigger channel is set up so that GRA is an output compare 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 PADDR and NDERA, and bits G1CMS1, G1CMS0, G0CMS1, and G0CMS0 are set in 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 NDRA.
- The timer counter in this ITU channel is started. When compare match A occurs, the NDRA contents are transferred to PADR and output. The compare match/input capture A (IMFA) interrupt service routine writes the next output data (H'C0) in NDRA.
- Five-phase overlapping pulse output (one or two phases active at a time) can be obtained by writing H'40, H'60, H'20, H'30, H'10, H'18, H'08, H'88... at successive IMFA interrupts.

Figure 9.5 Normal TPC Output Example (Five-Phase Pulse Output)

9.3.4 Non-Overlapping TPC Output

Sample Setup Procedure for Non-Overlapping TPC Output

Figure 9.6 shows a sample procedure for setting up non-overlapping TPC output.

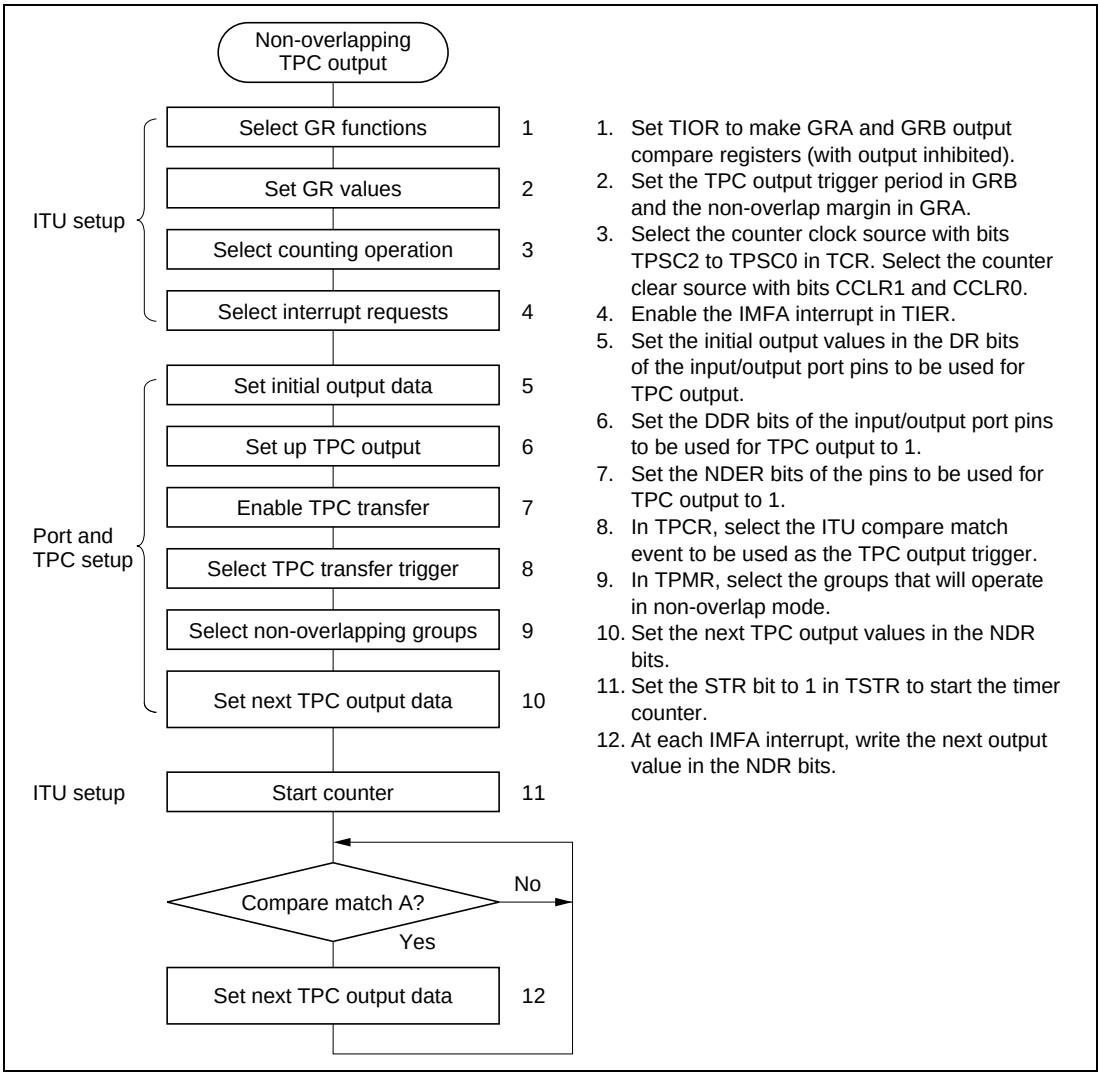
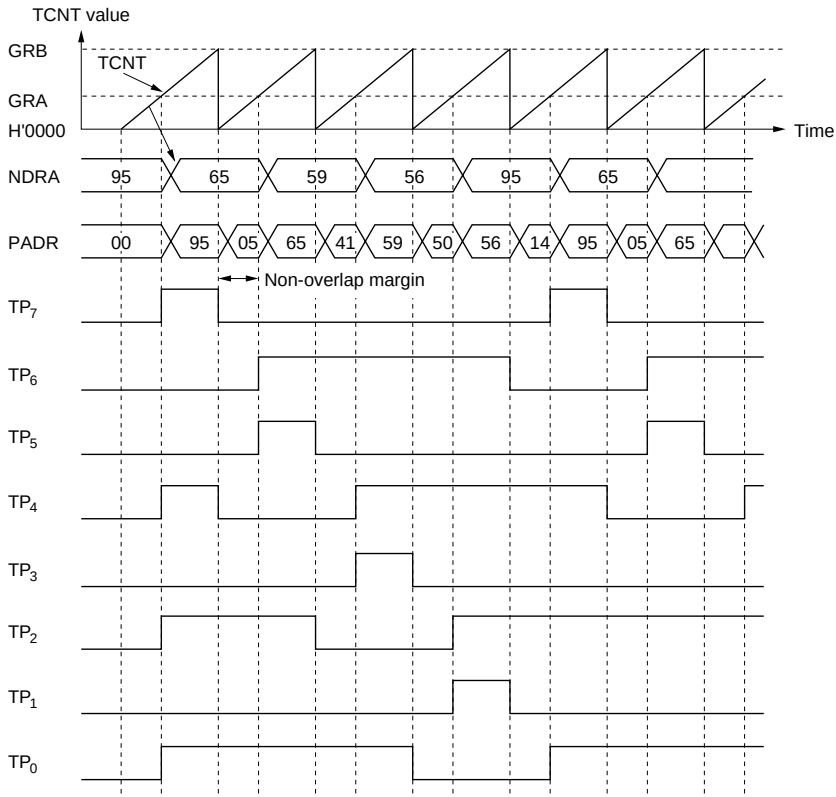


Figure 9.6 Setup Procedure for Non-Overlapping TPC Output (Example)

Example of Non-Overlapping TPC Output (Example of Four-Phase Complementary Non-Overlapping Output)

Figure 9.7 shows an example of the use of TPC output for four-phase complementary non-overlapping pulse output.



- The ITU channel to be used as the output trigger 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 PADDR and NDERA, and bits G1CMS1, G1CMS0, G0CMS1, and G0CMS0 are set in TPCR to select compare match in the ITU channel set up in step 1 as the output trigger. Bits G1NOV and G0NOV are set to 1 in TPMR to select non-overlapping output. Output data H'95 is written in NDRA.
- 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 NDRA.
- Four-phase complementary non-overlapping pulse output can be obtained by writing H'59, H'56, H'95... at successive IMFA interrupts.

Figure 9.7 Non-Overlapping TPC Output Example (Four-Phase Complementary Non-Overlapping Pulse Output)

9.3.5 TPC Output Triggering by Input Capture

TPC output can be triggered by ITU input capture as well as by compare match. If GRA functions as an input capture register in the ITU channel selected in TPCR, TPC output will be triggered by the input capture signal. Figure 9.8 shows the timing.

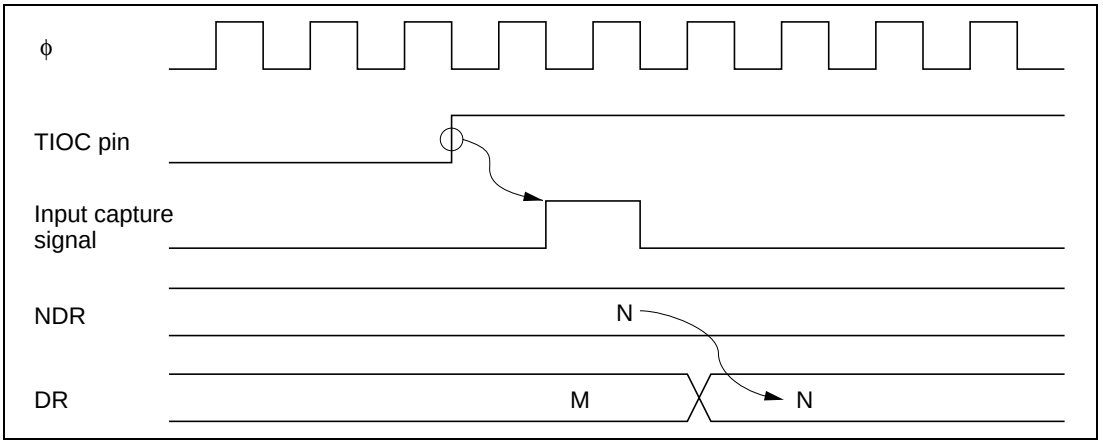


Figure 9.8 TPC Output Triggering by Input Capture (Example)

9.4 Usage Notes

9.4.1 Operation of TPC Output Pins

TP_0 to TP_{15}* are multiplexed with ITU pin functions. When ITU output is enabled, the corresponding pins cannot be used for TPC output. The data transfer from NDR bits to DR bits takes place, however, regardless of the usage of the pin.

Pin functions should be changed only under conditions in which the output trigger event will not occur.

Note: * Since this LSI does not have a TP_{14} pin, the TP_{14} signal cannot be output to the outside.

9.4.2 Note on Non-Overlapping Output

During non-overlapping operation, the transfer of NDR bit values to DR bits takes place as 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 transferred if their value is 1.

Figure 9.9 illustrates the non-overlapping TPC output operation.

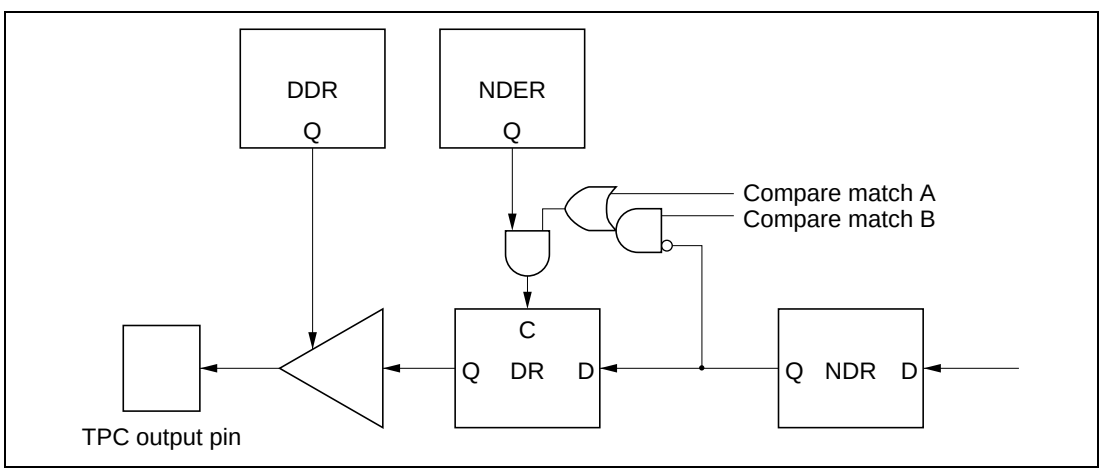


Figure 9.9 Non-Overlapping TPC Output

Therefore, 0 data can be transferred ahead of 1 data by making compare match B occur before compare match A. NDR contents should not be altered during the interval from compare match B to compare match A (the non-overlap margin).

This can be accomplished by having the IMFA interrupt service routine write the next data in NDR, or by having the IMFA interrupt activate the DMAC. The next data must be written before the next compare match B occurs.

Figure 9.10 shows the timing relationships.

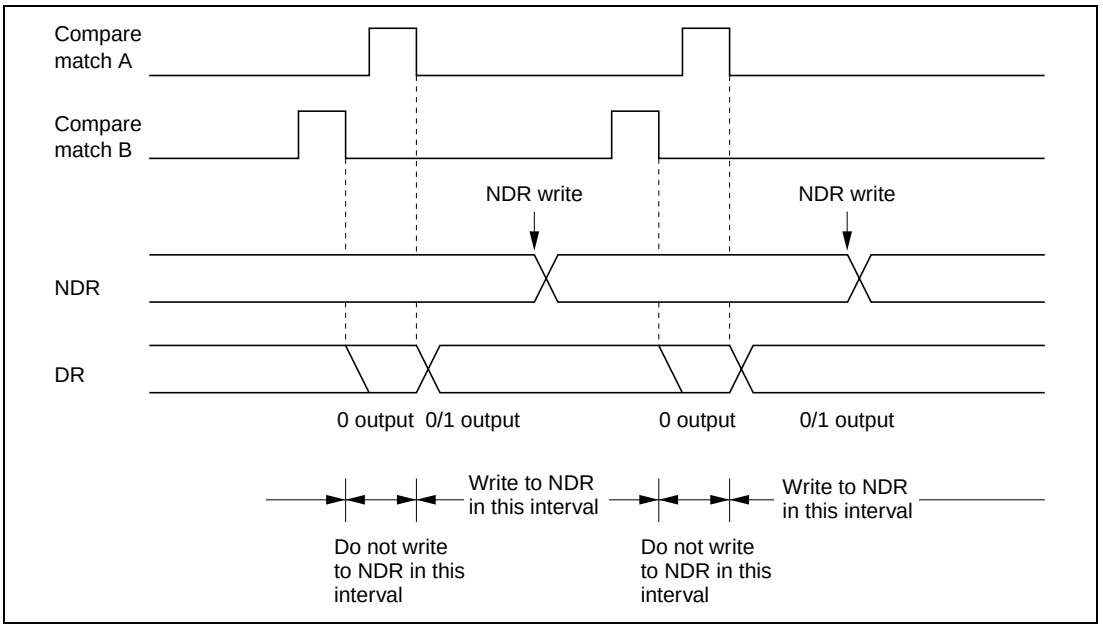


Figure 9.10 Non-Overlapping Operation and NDR Write Timing

Section 10 Watchdog Timer

10.1 Overview

The H8/3039 Group has an on-chip watchdog timer (WDT). The WDT has two selectable functions: it can operate as a watchdog timer to supervise system operation, or it can operate as an interval timer. As a watchdog timer, it generates a reset signal for the H8/3039 Group chip if a system crash allows the timer counter (TCNT) to overflow before being rewritten. In interval timer operation, an interval timer interrupt is requested at each TCNT overflow.

10.1.1 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 interrupt is generated in interval timer operation.
- Watchdog timer reset signal resets the entire H8/3039 Group chip internally, and can also be output externally.*
  The reset signal generated by timer counter overflow during watchdog timer operation resets the entire H8/3039 Group internally. An external reset signal can be output from the $\overline{\text{RESO}}$ pin to reset other system devices simultaneously.

Note: * The $\overline{\text{RESO}}$ pin of the mask ROM version is the dedicated FWE input pin of the F-ZTAT version. Therefore, the F-ZTAT version cannot output the reset signal to the outside.

10.1.2 Block Diagram

Figure 10.1 shows a block diagram of the WDT.

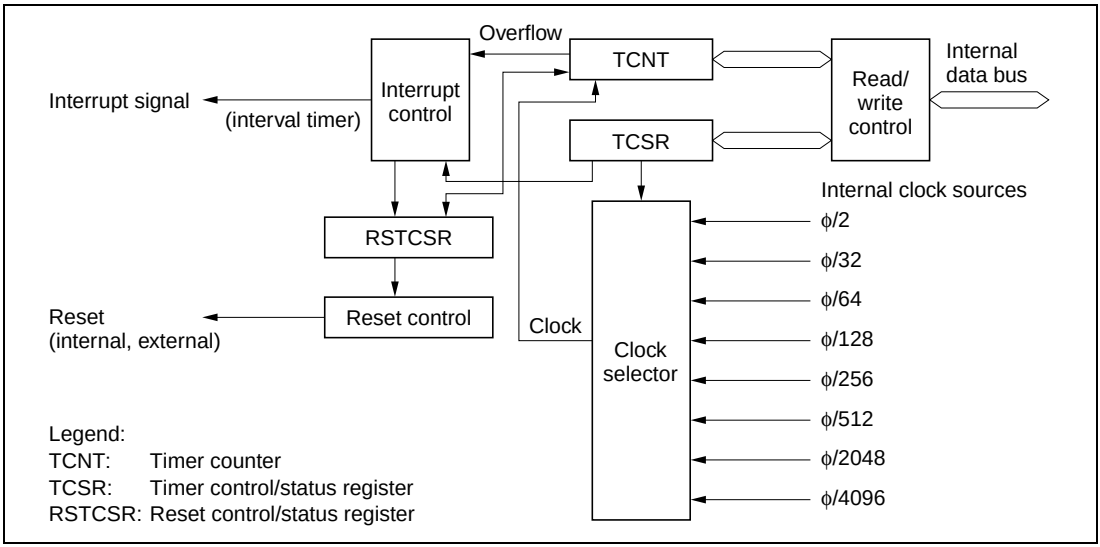


Figure 10.1 WDT Block Diagram

10.1.3 Pin Configuration

Table 10.1 describes the WDT output pin.*

Note: * Shows the mask ROM version pin. The F-ZTAT does not have any pins used by the WDT. For F-ZTAT version, see section 15.9, Notes on Flash Memory Programming/Erasing.

Table 10.1 WDT Pin

Name	Abbreviation	I/O	Function
Reset output	$\overline{\text{RESO}}$	Output*	External output of the watchdog timer reset signal

Note: * Open-drain output. Externally pull-up to Vcc whether or not the reset output is used

10.1.4 Register Configuration

Table 10.2 summarizes the WDT registers.

Table 10.2 WDT Registers

Address*[1]					
Write*[2]	**Read**	**Name**	**Abbreviation**	**R/W**	**Initial Value**
H'FFA8	H'FFA8	Timer control/status register	TCSR	R/(W)*[3]	H'18
	H'FFA9	Timer counter	TCNT	R/W	H'00
H'FFAA	H'FFAB	Reset control/status register	RSTCSR	R/(W)*[3]	H'3F

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.

10.2 Register Descriptions

10.2.1 Timer Counter (TCNT)

TCNT is an 8-bit readable and writable* up-counter.

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

When the TME bit is set to 1 in TCSR, TCNT starts counting pulses generated from an internal clock source selected by bits CKS2 to CKS0 in TCSR. When the count overflows (changes from H'FF to H'00), the OVF bit is set to 1 in TCSR. TCNT is initialized to H'00 by a reset and when the TME bit is cleared to 0.

Note: * TCNT is write-protected by a password. For details see section 10.2.4, Notes on Register Access.

10.2.2 Timer Control/Status Register (TCSR)

TCSR is an 8-bit readable and writable* register. Its functions include selecting the timer mode and clock source.

Note: * TCSR is write-protected by a password. For details see section 10.2.4, Notes on Register Access.

Bit	7	6	5	4	3	2	1	0
	OVF	WT/$\overline{IT}$	TME	—	—	CKS2	CKS1	CKS0
Initial value	0	0	0	1	1	0	0	0
Read/Write	R/(W)*	R/W	R/W	—	—	R/W	R/W	R/W

Clock select
These bits select the TCNT clock source

Reserved bits

Timer enable
Selects whether TCNT runs or halts

Timer mode select
Selects the mode

Overflow flag
Status flag indicating overflow

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 to 0 by a reset. In software standby mode bits 2 to 0 are not initialized, but retain their previous values.

Bit 7—Overflow Flag (OVF): This status flag indicates that the timer counter has overflowed 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	(Initial value)
1	[Setting condition]	
	Set when TCNT changes from H'FF to H'00	

Bit 6—Timer Mode Select (WT/$\overline{IT}$): Selects whether to use the WDT as a watchdog timer or interval timer. If used as an interval timer, the WDT generates an interval timer interrupt request when TCNT overflows. If used as a watchdog timer, the WDT generates a reset signal when TCNT overflows.

Bit 6 WT/$\overline{IT}$	Description	
0	Interval timer: requests interval timer interrupts	(Initial value)
1	Watchdog timer: generates a reset signal	

Bit 5—Timer Enable (TME): Selects whether TCNT runs or is halted.

Bit 5 TME	Description	
0	TCNT is initialized to H'00 and halted	(Initial value)
1	TCNT is counting	

Bits 4 and 3—Reserved: These bits cannot be modified and are always read as 1.

Bits 2 to 0—Clock Select 2 to 0 (CKS2/1/0): These bits select one of eight internal clock 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	(Initial value)
		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	

10.2.3 Reset Control/Status Register (RSTCSR)

RSTCSR is an 8-bit readable and writable* register that indicates when a reset signal has been generated by watchdog timer overflow, and controls external output of the reset signal.

Note: * RSTCSR is write-protected by a password. For details see section 10.2.4, Notes on Register Access.

Bit	7	6	5	4	3	2	1	0
	WRST	RSTOE	—	—	—	—	—	—
Initial value	0	0	1	1	1	1	1	1
Read/Write	R/(W)*1	R/W	—	—	—	—	—	—

Reserved bits

Reset output enable*2
Enables or disables external output of the reset signal

Watchdog timer reset
Indicates that a reset signal has been generated

Notes: 1. Only 0 can be written in bit 7 to clear the flag.
2. With the mask ROM version, enable and disable can be set. With the F-ZTAT version, do not set enable.

Bits 7 and 6 are initialized by input of a reset signal at the $\overline{\text{RES}}$ pin. They are not initialized by reset signals generated by watchdog timer overflow.

Bit 7—Watchdog Timer Reset (WRST): During watchdog timer operation, this bit indicates that TCNT has overflowed and generated a reset signal. This reset signal resets the entire chip internally. If bit RSTOE is set to 1, this reset signal is also output (low) at the $\overline{\text{RESO}}$ pin*[1] to initialize external system devices.

Bit 7 WRST	Description	
0	[Clearing conditions]	(Initial value)
	(1) Cleared to 0 by reset signal input at $\overline{\text{RES}}$ pin	
	(2) Cleared by reading WRST when WRST = 1, then writing 0 in WERST	
1	[Setting condition]	
	Set when TCNT overflow generates a reset signal during watchdog timer operation	

Bit 6—Reset Output Enable (RSTOE): Enables or disables external output at the $\overline{\text{RESO}}$ pin*[1] of the reset signal generated if TCNT overflows during watchdog timer operation.

Bit 6 RSTOE	Description	
0	Reset signal is not output externally	(Initial value)
1	Reset signal is output externally*[2]	

Notes: 1. Mask ROM version. Dedicated FWE input pin for F-ZTAT version.
2. Mask ROM version. Do not set to 1 with the F-ZTAT version.

Bits 5 to 0—Reserved: These bits cannot be modified and are always read as 1.

10.2.4 Notes on Register Access

The watchdog timer's TCNT, TCSR, and RSTCSR registers differ from other registers in being more difficult to write. The procedures for writing and reading these registers are given below.

Writing to TCNT and TCSR

These registers must be written by a word transfer instruction. They cannot be written by byte instructions. Figure 10.2 shows the format of data written to TCNT and TCSR. TCNT and TCSR both have the same write address. The write data must be contained in the lower byte of the written word. The upper byte must contain H'5A (password for TCNT) or H'A5 (password for TCSR). This transfers the write data from the lower byte to TCNT or TCSR.

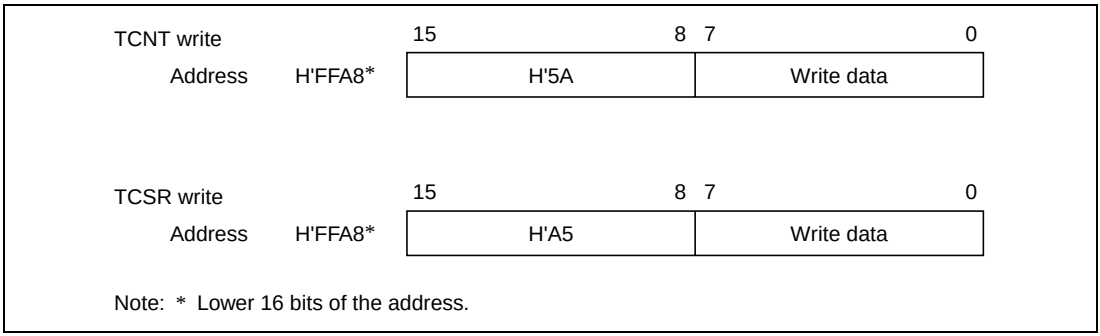


Figure 10.2 Format of Data Written to TCNT and TCSR

Writing to RSTCSR

RSTCSR must be written by a word transfer instruction. It cannot be written by byte transfer instructions. Figure 10.3 shows the format of data written to RSTCSR. To write 0 in the WRST bit, the write data must have H'A5 in the upper byte and H'00 in the lower byte. The H'00 in the lower byte clears the WRST bit in RSTCSR to 0. To write to the RSTOE bit, the upper byte must contain H'5A and the lower byte must contain the write data. Writing this word transfers a write data value into the RSTOE bit.

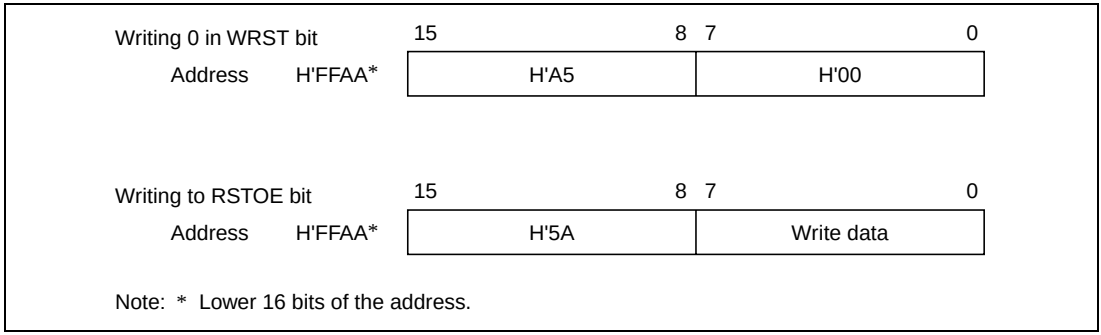


Figure 10.3 Format of Data Written to RSTCSR

Reading TCNT, TCSR, and RSTCSR

These registers are read like other registers. Byte access instructions can be used. The read addresses are H'FFA8 for TCSR, H'FFA9 for TCNT, and H'FFAB for RSTCSR, as listed in table 10.3.

Table 10.3 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.

10.3 Operation

Operations when the WDT is used as a watchdog timer and as an interval timer are described below.

10.3.1 Watchdog Timer Operation

Figure 10.4 illustrates watchdog timer operation. To use the WDT as a watchdog timer, set the WT/$\overline{\text{IT}}$ and TME bits to 1 in TCSR. Software must prevent TCNT overflow by rewriting the TCNT value (normally by writing H'00) before overflow occurs. If TCNT fails to be rewritten and overflows due to a system crash etc., the H8/3039 Group is internally reset for a duration of 518 states.

The watchdog reset signal can be externally output from the $\overline{\text{RESO}}$ pin* to reset external system devices. The reset signal is output externally for 132 states. External output can be enabled or disabled by the RSTOE bit in RSTCSR.

A watchdog reset has the same vector as a reset generated by input at the $\overline{\text{RES}}$ pin. Software can 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.

Note: * Mask ROM version.
Since the RES pin is a dedicated FWE input pin with the F-ZTAT version, the reset signal cannot be output to the outside.

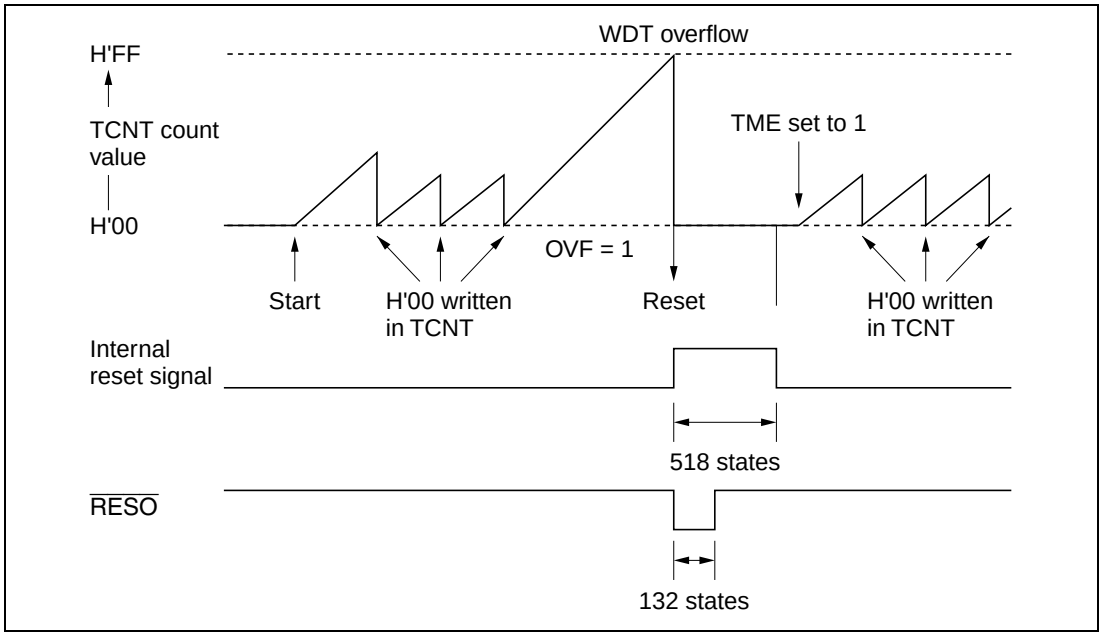


Figure 10.4 Watchdog Timer Operation (Mask ROM Version)

10.3.2 Interval Timer Operation

Figure 10.5 illustrates interval timer operation. To use the WDT as an interval timer, clear bit WT/$\overline{IT}$ to 0 and set bit TME to 1 in TCSR. An interval timer interrupt request is generated at each TCNT overflow. This function can be used to generate interval timer interrupts at regular intervals.

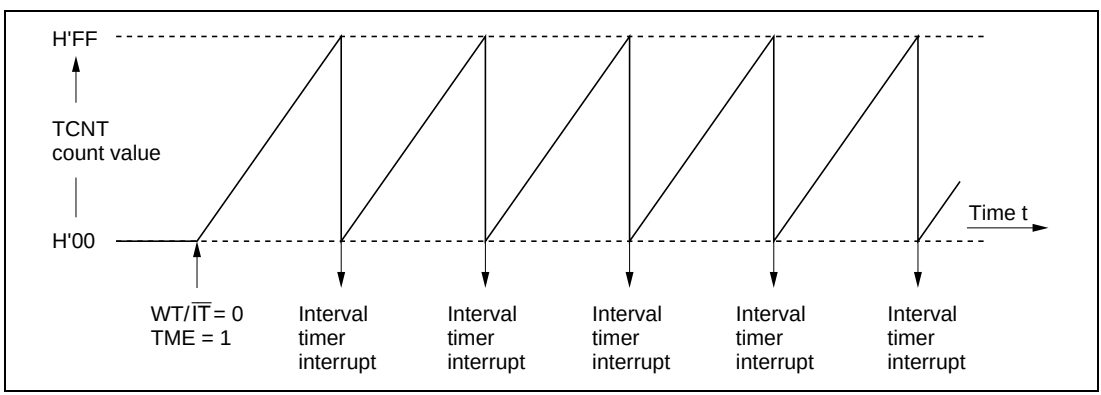


Figure 10.5 Interval Timer Operation

10.3.3 Timing of Setting of Overflow Flag (OVF)

Figure 10.6 shows the timing of setting of the OVF flag in TCSR. The OVF flag is set to 1 when TCNT overflows. At the same time, a reset signal is generated in watchdog timer operation, or an interval timer interrupt is generated in interval timer operation.

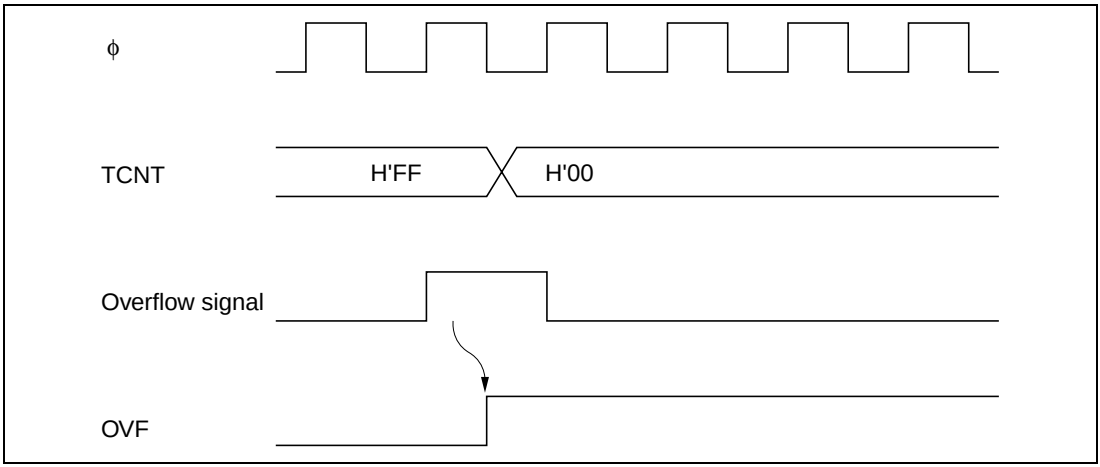


Figure 10.6 Timing of Setting of OVF

10.3.4 Timing of Setting of Watchdog Timer Reset Bit (WRST)

The WRST bit in RSTCSR is valid when bits WT/$\overline{IT}$ and TME are both set to 1 in TCSR.

Figure 10.7 shows the timing of setting of WRST and the internal reset timing. The WRST bit is set to 1 when TCNT overflows and OVF is set to 1. At the same time an internal reset signal is generated for the entire H8/3039 Group chip. This internal reset signal clears OVF to 0, but the WRST bit remains set to 1. The reset routine must therefore clear the WRST bit.

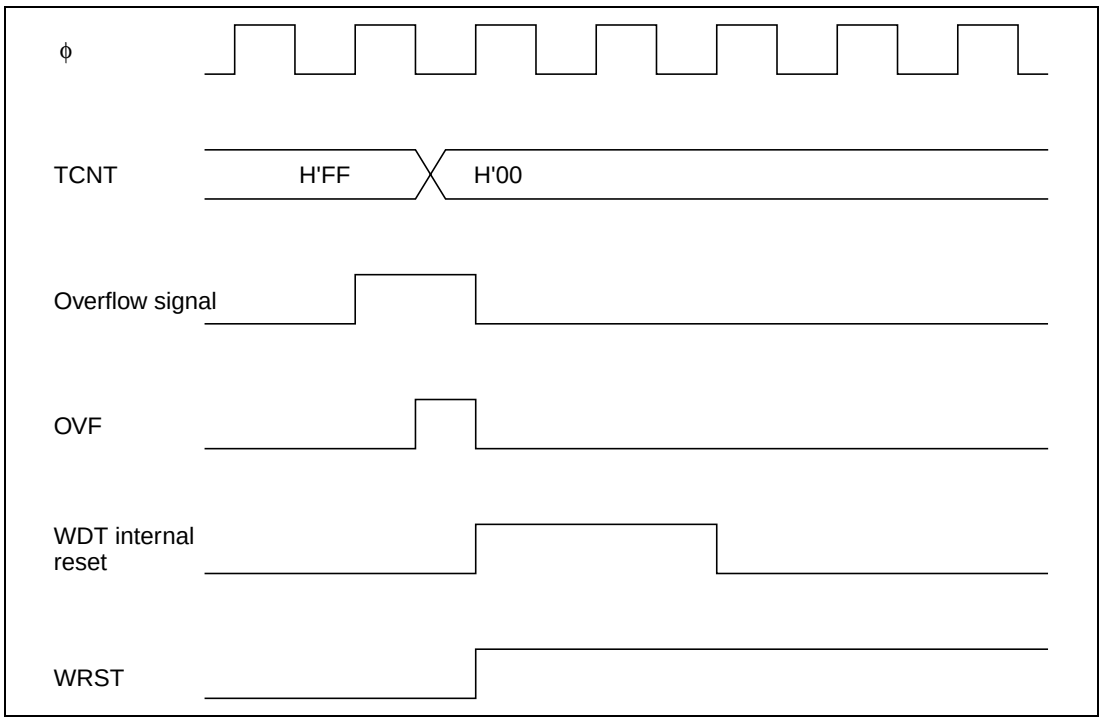


Figure 10.7 Timing of Setting of WRST Bit and Internal Reset

10.4 Interrupts

During interval timer operation, an overflow generates an interval timer interrupt (WOVI). The interval timer interrupt is requested whenever the OVF bit is set to 1 in TCSR.

10.5 Usage Notes

Contention between TCNT Write and Increment

If a timer counter clock pulse is generated during the T3 state of a write cycle to TCNT, the write takes priority and the timer count is not incremented. See figure 10.8.

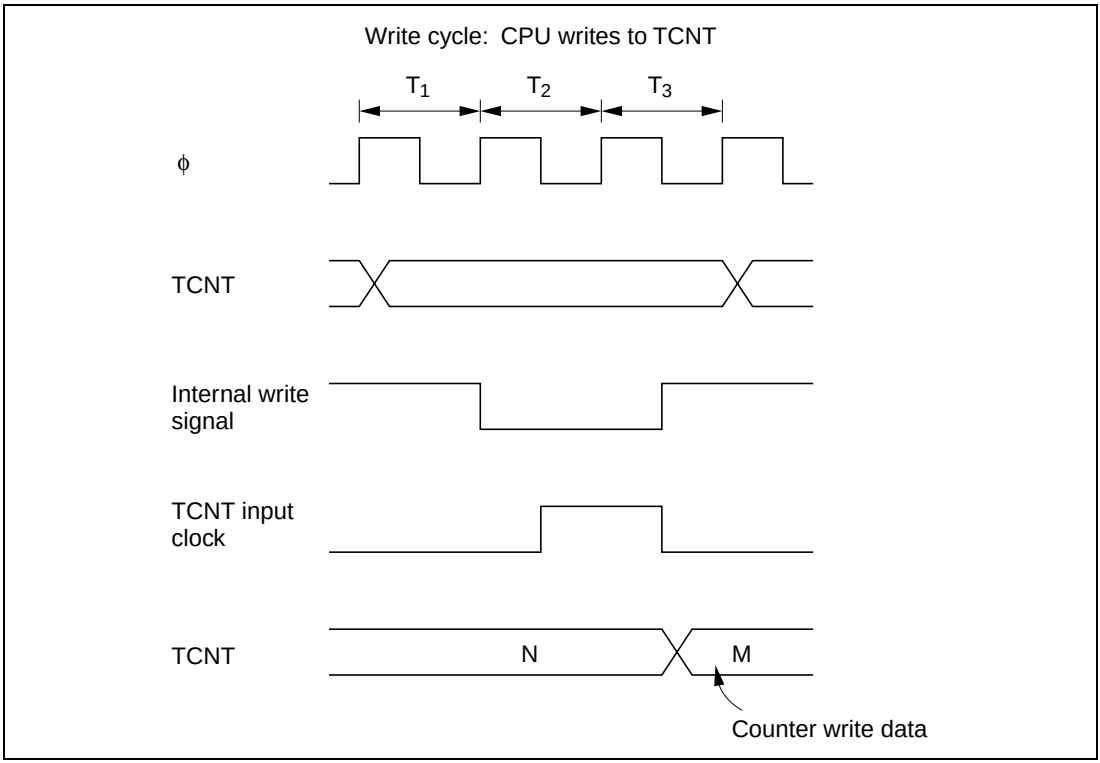


Figure 10.8 Contention between TCNT Write and Increment

Changing CKS2 to CKS0 Values

Halt TCNT by clearing the TME bit to 0 in TCSR before changing the values of bits CKS2 to CKS0.

Section 11 Serial Communication Interface

11.1 Overview

The H8/3039 Group has a serial communication interface (SCI) with two independent channels. The two channels are functionally identical. The SCI can communicate in asynchronous or synchronous mode. It also has a multiprocessor communication function for serial communication among two or more processors.

When the SCI is not used, it can be halted to conserve power. Each SCI channel can be halted independently. For details see section 17.6, Module Standby Function.

Channel 0 (SCI0) also has a smart card interface function conforming to the ISO/IEC7816-3 (Identification Card) standard. This function supports serial communication with a smart card. For details, see section 12, Smart Card Interface.

11.1.1 Features

SCI features are listed below.

- Selection of asynchronous or synchronous mode for serial communication

a. Asynchronous mode

Serial data communication is synchronized one character at a time. The SCI can communicate with a universal asynchronous receiver/transmitter (UART), asynchronous communication interface adapter (ACIA), or other chip that employs standard asynchronous serial communication. It can also communicate with two or more other processors using the multiprocessor communication function. There are twelve selectable serial data communication formats.

— Data length: 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 occurs

b. Synchronous mode

Serial data communication is synchronized with a clock signal. The SCI can communicate with other chips having a synchronous communication function. There is one serial data communication format.

— Data length: 8 bits

— Receive error detection: overrun errors

- Full-duplex communication

The transmitting and receiving sections are independent, so the SCI can transmit and receive simultaneously. The transmitting and receiving sections are both double-buffered, so serial 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, or external clock from the SCK pin.
- Four types of interrupts
  Transmit-data-empty, transmit-end, receive-data-full, and receive-error interrupts are requested independently.

11.1.2 Block Diagram

Figure 11.1 shows a block diagram of the SCI.

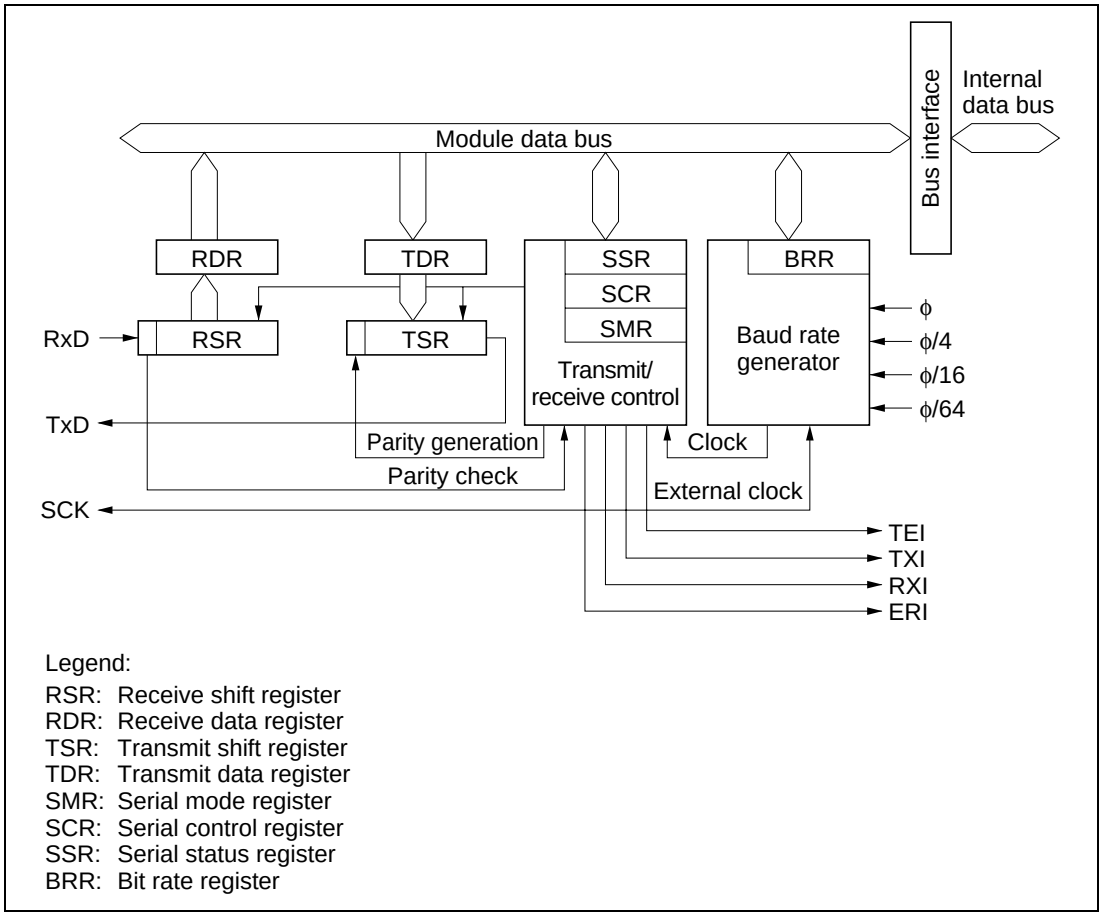


Figure 11.1 SCI Block Diagram

11.1.3 Input/Output Pins

The SCI has the serial pins for each channel as listed in table 11.1.

Table 11.1 SCI Pins

Channel	Name	Abbreviation	I/O	Function
0	Serial clock pin	SCK_0	Input/output	SCI_0 clock input/output
	Receive data pin	RxD_0	Input	SCI_0 receive data input
	Transmit data pin	TxD_0	Output	SCI_0 transmit data output
1	Serial clock pin	SCK_1	Input/output	SCI_1 clock input/output
	Receive data pin	RxD_1	Input	SCI_1 receive data input
	Transmit data pin	TxD_1	Output	SCI_1 transmit data output

11.1.4 Register Configuration

The SCI has the internal registers as listed in table 11.2. These registers select asynchronous or synchronous mode, specify the data format and bit rate, and control the transmitter and receiver sections.

Table 11.2 Registers

Channel	Address*[1]	Name	Abbreviation	R/W	Initial Value
0	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]	H'84
	H'FFB5	Receive data register	RDR	R	H'00
1	H'FFB8	Serial mode register	SMR	R/W	H'00
	H'FFB9	Bit rate register	BRR	R/W	H'FF
	H'FFBA	Serial control register	SCR	R/W	H'00
	H'FFBB	Transmit data register	TDR	R/W	H'FF
	H'FFBC	Serial status register	SSR	R/(W)*[2]	H'84
	H'FFBD	Receive data register	RDR	R	H'00

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

11.2 Register Descriptions

11.2.1 Receive Shift Register (RSR)

RSR is an 8-bit register that receives serial data.

Bit	7	6	5	4	3	2	1	0
Read/Write	—	—	—	—	—	—	—	—

The SCI loads serial data input at the RxD pin into RSR in the order received, LSB (bit 0) first, thereby converting the data to parallel data. When 1 byte has been received, it is automatically transferred to RDR. The CPU cannot read or write RSR directly.

11.2.2 Receive Data Register (RDR)

RDR is an 8-bit register that stores received serial data.

Bit	7	6	5	4	3	2	1	0
Initial value	0	0	0	0	0	0	0	0
Read/Write	R	R	R	R	R	R	R	R

When the SCI finishes receiving 1 byte of serial data, it transfers the received data from RSR into RDR for storage. RSR is then ready to receive the next data. This double buffering allows data to be received continuously.

RDR is a read-only register. Its contents cannot be modified by the CPU. RDR is initialized to H'00 by a reset and in standby mode.

11.2.3 Transmit Shift Register (TSR)

TSR is an 8-bit register used to transmit serial data.

Bit	7	6	5	4	3	2	1	0
Read/Write	—	—	—	—	—	—	—	—

The SCI loads transmit data from TDR into TSR, then transmits the data serially from the TxD pin, LSB (bit 0) first. After transmitting one data byte, the SCI automatically loads the next transmit data from TDR into TSR and starts transmitting it. If the TDRE flag is set to 1 in SSR, however, the SCI does not load the TDR contents into TSR. The CPU cannot read or write TSR directly.

11.2.4 Transmit Data Register (TDR)

TDR is an 8-bit register that stores data for serial transmission.

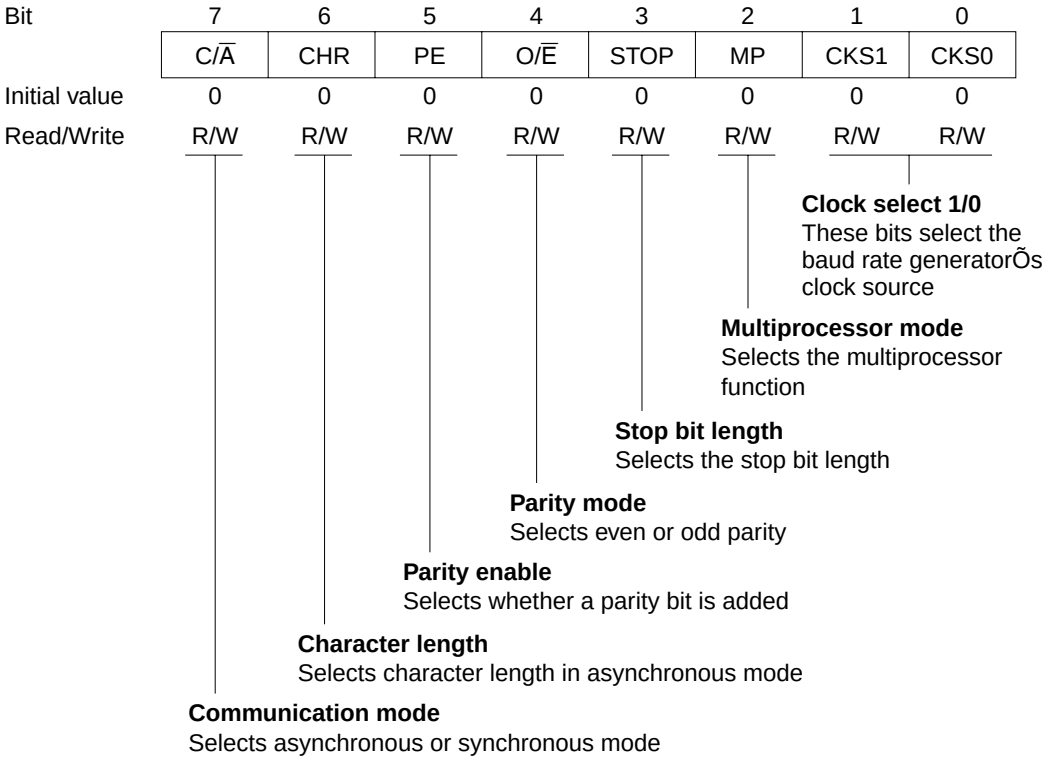
Bit	7	6	5	4	3	2	1	0
Initial value	1	1	1	1	1	1	1	1
Read/Write	R/W	R/W	R/W	R/W	R/W	R/W	R/W	R/W

When the SCI detects that TSR is empty, it moves transmit data written in TDR from TDR into TSR and starts serial transmission. Continuous serial transmission is possible by writing the next 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 standby mode.

11.2.5 Serial Mode Register (SMR)

SMR is an 8-bit register that specifies the SCI serial communication format and selects the clock source for the baud rate generator.



The CPU can always read and write SMR. SMR is initialized to H'00 by a reset and in standby mode.

Bit 7—Communication Mode (C/$\overline{A}$): Selects whether the SCI operates in asynchronous or synchronous mode.

Bit 7 C/$\overline{A}$	Description	
0	Asynchronous mode	(Initial value)
1	Synchronous mode	

Bit 6—Character Length (CHR): Selects 7-bit or 8-bit data length in asynchronous mode. In synchronous mode the data length is 8 bits regardless of the CHR setting.

Bit 6 CHR	Description	
0	8-bit data	(Initial value)
1	7-bit data*	

Note: * When 7-bit data is selected, the MSB (bit 7) in TDR is not transmitted.

Bit 5—Parity Enable (PE): In asynchronous mode, this bit enables or disables the addition of a parity bit to transmit data, and the checking of the parity bit in receive data. In synchronous mode the parity bit is neither added nor checked, regardless of the PE setting.

Bit 5 PE	Description	
0	Parity bit not added or checked	(Initial value)
1	Parity bit added and checked*	

Note: * When PE is set to 1, an even or odd parity bit is added to transmit data according to the even or odd parity mode selected by the O/$\overline{E}$ bit, and the parity bit in receive data is checked to see that it matches the even or odd mode selected by the O/$\overline{E}$ bit.

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 parity bit. The O/$\overline{E}$ setting is ignored in synchronous mode, or when parity adding and checking is disabled in asynchronous mode.

Bit 4 O/$\overline{E}$	Description	
0	Even parity*[1]	(Initial value)
1	Odd parity*[2]	

Notes: 1. When even parity is selected, the parity bit added to transmit data makes an even number of 1s in the transmitted character and parity bit combined. Receive data must 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 odd number of 1s in the transmitted character and parity bit combined. Receive data must have an odd number of 1s in the received character and parity bit combined.

Bit 3—Stop Bit Length (STOP): Selects one or two stop bits in asynchronous mode. This setting is used only in asynchronous mode. In synchronous mode no stop bit is added, so the STOP bit setting is ignored.

Bit 3 STOP	Description	
0	One stop bit*[1]	(Initial value)
1	Two stop bits*[2]	

Notes: 1. One stop bit (with value 1) is added at the end of each transmitted character.
2. Two stop bits (with value 1) are added at the end of each transmitted character.

In receiving, only the first stop bit is checked, regardless of the STOP bit setting. If the second stop bit is 1 it is treated as a stop bit. If the second stop bit is 0 it is treated as the start bit of the next incoming character.

Bit 2—Multiprocessor Mode (MP): Selects a multiprocessor format. When a multiprocessor format is selected, parity settings made by the PE and O/$\overline{E}$ bits are ignored. The MP bit setting is valid only in asynchronous mode. It is ignored in synchronous mode.

For further information on the multiprocessor communication function, see section 11.3.3, Multiprocessor Communication.

Bit 2 MP	Description	
0	Multiprocessor function disabled	(Initial value)
1	Multiprocessor format selected	

Bits 1 and 0—Clock Select 1 and 0 (CKS1/0): These bits select the clock source of the on-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, see section 11.2.8, Bit Rate Register (BRR).

Bit 1 CKS1	Bit 0 CKS0	Description	
0	0	ϕ clock selected	(Initial value)
0	1	ϕ/4 clock selected	
1	0	ϕ/16 clock selected	
1	1	ϕ/64 clock selected	

11.2.6 Serial Control Register (SCR)

SCR enables the SCI transmitter and receiver, enables or disables serial clock output in asynchronous mode, enables or disables interrupts, and selects the transmit/receive clock source.

Bit	7	6	5	4	3	2	1	0
	TIE	RIE	TE	RE	MPIE	TEIE	CKE1	CKE0
Initial value	0	0	0	0	0	0	0	0
Read/Write	R/W	R/W	R/W	R/W	R/W	R/W	R/W	R/W

Clock enable 1/0
These bits select the SCI clock source

Transmit end interrupt enable
Enables or disables transmit-end interrupts (TEI)

Multiprocessor interrupt enable
Enables or disables multiprocessor interrupts

Receive enable
Enables or disables the receiver

Transmit enable
Enables or disables the transmitter

Receive interrupt enable
Enables or disables receive-data-full interrupts (RXI) and receive-error interrupts (ERI)

Transmit interrupt enable
Enables or disables transmit-data-empty interrupts (TXI)

The CPU can always read and write SCR. SCR is initialized to H'00 by a reset and in standby mode.

Bit 7—Transmit Interrupt Enable (TIE): Enables or disables the transmit-data-empty interrupt (TXI) requested when the TDRE flag in SSR is set to 1 due to transfer of serial transmit data from TDR to TSR.

Bit 7 TIE	Description	
0	Transmit-data-empty interrupt request (TXI) is disabled*	(Initial value)
1	Transmit-data-empty interrupt request (TXI) is enabled	

Note: * TXI interrupt requests can be cleared by reading the value 1 from the TDRE flag, then 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 interrupt (RXI) requested when the RDRF flag is set to 1 in SSR due to transfer of serial receive data from RSR to RDR; also enables or disables the receive-error interrupt (ERI).

Bit 6 RIE	Description	
0	Receive-end (RXI) and receive-error (ERI) interrupt requests are disabled*	(Initial value)
1	Receive-end (RXI) and receive-error (ERI) interrupt requests are enabled	

Note: * RXI and ERI interrupt requests can be cleared by reading the value 1 from the RDRF, FER, PER, or ORER flag, then clearing it to 0; or by clearing the RIE bit to 0.

Bit 5—Transmit Enable (TE): Enables or disables the start of SCI serial transmitting operations.

Bit 5 TE	Description	
0	Transmitting disabled*[1]	(Initial value)
1	Transmitting enabled*[2]	

Notes: 1. The TDRE flag is fixed at 1 in SSR.
2. In the enabled state, serial transmitting starts when the TDRE flag in SSR is cleared to 0 after writing of transmit data into TDR. Select the transmit format in SMR before setting the TE bit to 1.

Bit 4—Receive Enable (RE): Enables or disables the start of SCI serial receiving operations.

Bit 4 RE	Description	
0	Receiving disabled*[1]	(Initial value)
1	Receiving enabled*[2]	

Notes: 1. Clearing the RE bit to 0 does not affect the RDRF, FER, PER, and ORER flags. These flags retain their previous values.
2. In the enabled state, serial receiving starts when a start bit is detected in asynchronous mode, or serial clock input is detected in synchronous mode. Select the receive format in SMR before setting the RE bit to 1.

Bit 3—Multiprocessor Interrupt Enable (MPIE): Enables or disables multiprocessor interrupts. The MPIE setting is valid only in asynchronous mode, and only if the MP bit is set to 1 in SMR. 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) [Clearing conditions] • The MPIE bit is cleared to 0 • MPB = 1 in received data	(Initial value)
1	Multiprocessor interrupts are enabled* Receive-data-full interrupts (RXI), receive-error interrupts (ERI), and setting of the RDRF, FER, and ORER status flags in SSR are disabled until data with the multiprocessor bit set to 1 is received.	

Note: * The SCI does not transfer receive data from RSR to RDR, does not detect receive errors, and does not set the RDRF, FER, and ORER flags in SSR. When it receives data in which MPB = 1, the SCI sets the MPB bit to 1 in SSR, automatically clears the MPIE bit to 0, and enables RXI and ERI interrupts (if the RIE bit is set to 1 in SCR) and setting of the FER and ORER flags.

Bit 2—Transmit-End Interrupt Enable (TEIE): Enables or disables the transmit-end interrupt (TEI) requested if TDR does not contain new transmit data when the MSB is transmitted.

Bit 2 TEIE	Description	
0	Transmit-end interrupt requests (TEI) are disabled*	(Initial value)
1	Transmit-end interrupt requests (TEI) are enabled*	

Note: * TEI interrupt requests can be cleared by reading the value 1 from the TDRE flag in SSR, then clearing the TDRE flag to 0, thereby also clearing the TEND flag to 0; or by clearing the TEIE bit to 0.

Bits 1 and 0—Clock Enable 1 and 0 (CKE1/0): These bits select the SCI clock source and enable or disable clock output from the SCK pin. Depending on the settings of CKE1 and CKE0, the SCK pin can be used for generic input/output, serial clock output, or serial clock input.

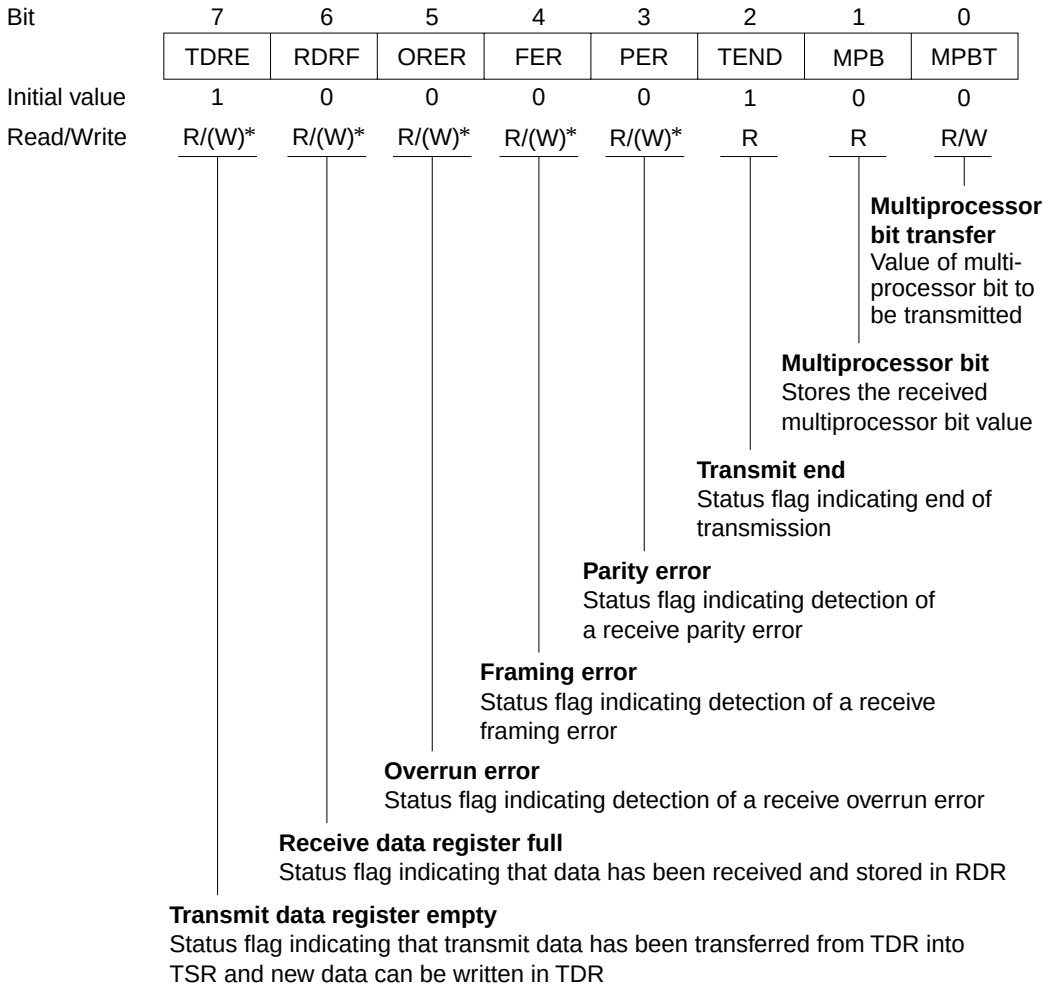
The CKE0 setting is valid only in asynchronous mode, and only when the SCI is internally clocked (CKE1 = 0). The CKE0 setting is ignored in synchronous mode, or when an external clock source is selected (CKE1 = 1). After setting the CKE1 and CKE0 bits, select the SCI operating mode in SMR. For further details on selection of the SCI clock source, see table 11.9.

Bit 1 CKE1	Bit 0 CKE0	Description	
0	0	Asynchronous mode	Internal clock, SCK pin available for generic input/output*[1]
		Synchronous mode	Internal clock, SCK pin used for serial clock output*[1]
0	1	Asynchronous mode	Internal clock, SCK pin used for clock output*[2]
		Synchronous mode	Internal clock, SCK pin used for serial clock output
1	0	Asynchronous mode	External clock, SCK pin used for clock input*[3]
		Synchronous mode	External clock, SCK pin used for serial clock input
1	1	Asynchronous mode	External clock, SCK pin used for clock input*[3]
		Synchronous mode	External clock, SCK pin used for serial clock input

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.

11.2.7 Serial Status Register (SSR)

SSR is an 8-bit register containing multiprocessor bit values, and status flags that indicate the SCI operating status.



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, ORER, PER, and FER flags. These flags can be cleared to 0 only if they have first been read while set to 1. The TEND and MPB flags are read-only bits that cannot be written.

SSR is initialized to H'84 by a reset and in standby mode.

Bit 7—Transmit Data Register Empty (TDRE): Indicates that the SCI has loaded transmit data from TDR into TSR and the next serial transmit data can be written in TDR.

Bit 7 TDRE	Description	
0	TDR contains valid transmit data [Clearing condition] Software reads TDRE while it is set to 1, then writes 0	
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 in TDR	(Initial value)

Bit 6—Receive Data Register Full (RDRF): Indicates that RDR contains new receive data.

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	(Initial value)
1	RDR contains new receive data [Setting condition] When serial data is received normally and transferred from RSR to RDR	

Note: The RDR contents and RDRF flag are not affected by detection of receive errors or by clearing of the RE bit to 0 in SCR. They retain their previous values. If the RDRF flag is still set to 1 when reception of the next data ends, an overrun error occurs and receive data is lost.

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

Bit 5 ORER	Description	
0	Receiving is in progress or has ended normally [Clearing conditions] • The chip is reset or enters standby mode • Software reads ORER while it is set to 1, then writes 0	(Initial value)*[1]
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 its previous value.
2. RDR continues to hold the receive data before the overrun error, so subsequent receive data is lost. Serial receiving cannot continue while the ORER flag is set to 1. In synchronous mode, serial transmitting is also disabled.

Bit 4—Framing Error (FER): Indicates that data reception ended abnormally due to a framing error in asynchronous mode.

Bit 4 FER	Description	
0	Receiving is in progress or has ended normally [Clearing conditions] • The chip is reset or enters standby mode • Software reads FER while it is set to 1, then writes 0	(Initial value)*[1]
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 its previous value.
2. When the stop bit length is 2 bits, only the first bit is checked. The second stop bit is not checked. When a framing error occurs the SCI transfers the receive data into RDR but does not set the RDRF flag. Serial receiving cannot continue while the FER flag is set to 1. In synchronous mode, serial transmitting is also disabled.

Bit 3—Parity Error (PER): Indicates that data reception ended abnormally due to a parity error in asynchronous mode.

Bit 3 PER	Description
0	Receiving is in progress or has ended normally*[1] (Initial value) [Clearing condition] The chip is reset or enters standby mode. Software reads PER while it is set to 1, then writes 0
1	A receive parity error occurred*[2] [Setting condition] The number of 1s in receive data, including the parity bit, does not match the 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 its previous value.

2. When a parity error occurs the SCI transfers the receive data into RDR but does not set the RDRF flag. Serial receiving cannot continue while the PER flag is set to 1. In synchronous mode, serial transmitting is also disabled.

Bit 2—Transmit End (TEND): Indicates that when the last bit of a serial character was transmitted TDR did not contain new transmit data, so transmission has ended. The TEND flag is a read-only bit and cannot be written.

Bit 2 TEND	Description
0	Transmission is in progress [Clearing condition] Software reads TDRE while it is set to 1, then writes 0 in the TDRE flag
1	End of transmission (Initial value) [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

Bit 1—Multiprocessor Bit (MPB): Stores the value of the multiprocessor bit in receive data when a multiprocessor format is used in asynchronous mode. MPB is a read-only bit and cannot be written.

Bit 1 MPB	Description	
0	Multiprocessor bit value in receive data is 0*	(Initial value)
1	Multiprocessor bit value in receive data is 1	

Note: * If the RE bit is cleared to 0 when a multiprocessor format is selected, MPB retains its previous value.

Bit 0—Multiprocessor Bit Transfer (MPBT): Stores the value of the multiprocessor bit added to transmit data when a multiprocessor format is selected for transmitting in asynchronous mode. The MPBT setting is ignored in synchronous mode, when a multiprocessor format is not selected, or when the SCI is not transmitting.

Bit 0 MPBT	Description	
0	Multiprocessor bit value in transmit data is 0	(Initial value)
1	Multiprocessor bit value in transmit data is 1	

11.2.8 Bit Rate Register (BRR)

BRR is an 8-bit register that, together with the CKS1 and CKS0 bits in SMR that select the baud rate generator clock source, determines the serial communication bit rate.

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

The CPU can always read and write BRR. BRR is initialized to H'FF by a reset and in standby mode.

The baud rate generator is controlled separately for the individual channels, so different values may be set for each.

Table 11.3 shows examples of BRR settings in asynchronous mode. Table 11.4 shows examples of BRR settings in synchronous mode.

Table 11.3 Examples of Bit Rates and BRR Settings in Asynchronous Mode

	φ (MHz)											
	2			2.097152			2.4576			3		
Bit Rate (bits/s)	n	N	Error (%)	n	N	Error (%)	n	N	Error (%)	n	N	Error (%)
110	1	141	0.03	1	148	–0.04	1	174	–0.26	1	212	0.03
150	1	103	0.16	1	108	0.21	1	127	0	1	155	0.16
300	0	207	0.16	0	217	0.21	0	255	0	1	77	0.16
600	0	103	0.16	0	108	0.21	0	127	0	0	155	0.16
1200	0	51	0.16	0	54	–0.70	0	63	0	0	77	0.16
2400	0	25	0.16	0	26	1.14	0	31	0	0	38	0.16
4800	0	12	0.16	0	13	–2.48	0	15	0	0	19	–2.34
9600	0	6	–6.99	0	6	–2.48	0	7	0	0	9	–2.34
19200	0	2	8.51	0	2	13.78	0	3	0	0	4	–2.34
31250	0	1	0	0	1	4.86	0	1	22.88	0	2	0
38400	0	1	–18.62	0	1	–14.67	0	1	0	—	—	—

	φ (MHz)											
	3.6864			4			4.9152			5		
Bit Rate (bits/s)	n	N	Error (%)	n	N	Error (%)	n	N	Error (%)	n	N	Error (%)
110	2	64	0.70	2	70	0.03	2	86	0.31	2	88	–0.25
150	1	191	0	1	207	0.16	1	255	0	2	64	0.16
300	1	95	0	1	103	0.16	1	127	0	1	129	0.16
600	0	191	0	0	207	0.16	0	255	0	1	64	0.16
1200	0	95	0	0	103	0.16	0	127	0	0	129	0.16
2400	0	47	0	0	51	0.16	0	63	0	0	64	0.16
4800	0	23	0	0	25	0.16	0	31	0	0	32	–1.36
9600	0	11	0	0	12	0.16	0	15	0	0	15	1.73
19200	0	5	0	0	6	–6.99	0	7	0	0	7	1.73
31250	—	—	—	0	3	0	0	4	–1.70	0	4	0
38400	0	2	0	0	2	8.51	0	3	0	0	3	1.73

	φ (MHz)											
	6			6.144			7.3728			8		
Bit Rate (bits/s)	n	N	Error (%)	n	N	Error (%)	n	N	Error (%)	n	N	Error (%)
110	2	106	–0.44	2	108	0.08	2	130	–0.07	2	141	0.03
150	1	77	0.16	2	79	0	2	95	0	2	103	0.16
300	1	155	0.16	1	159	0	1	191	0	1	207	0.16
600	1	77	0.16	1	79	0	1	95	0	1	103	0.16
1200	0	155	0.16	0	159	0	0	191	0	0	207	0.16
2400	0	77	0.16	0	79	0	0	95	0	0	103	0.16
4800	0	38	0.16	0	39	0	0	47	0	0	51	0.16
9600	0	19	–2.34	0	19	0	0	23	0	0	25	0.16
19200	0	9	–2.34	0	9	0	0	11	0	0	12	0.16
31250	0	5	0	0	5	2.40	0	6	5.33	0	7	0
38400	0	4	–2.34	0	4	0	0	5	0	0	6	–6.99

	φ (MHz)											
	9.8304			10			12			12.288		
Bit Rate (bits/s)	n	N	Error (%)	n	N	Error (%)	n	N	Error (%)	n	N	Error (%)
110	2	174	–0.26	2	177	–0.25	2	212	0.03	2	217	0.08
150	2	127	0	2	129	0.16	2	155	0.16	2	159	0
300	1	255	0	2	64	0.16	2	77	0.16	2	79	0
600	1	127	0	1	129	0.16	1	155	0.16	1	159	0
1200	0	255	0	1	64	0.16	1	77	0.16	1	79	0
2400	0	127	0	0	129	0.16	0	155	0.16	0	159	0
4800	0	63	0	0	64	0.16	0	77	0.16	0	79	0
9600	0	31	0	0	32	–1.36	0	38	0.16	0	39	0
19200	0	15	0	0	15	1.73	0	19	–2.34	0	19	0
31250	0	9	–1.70	0	9	0	0	11	0	0	11	2.40
38400	0	7	0	0	7	1.73	0	9	–2.34	0	9	0

	φ (MHz)											
	14			14.7456			16			18		
Bit Rate (bits/s)	n	N	Error (%)	n	N	Error (%)	n	N	Error (%)	n	N	Error (%)
110	2	248	–0.17	3	64	0.70	3	70	0.03	3	79	–0.12
150	2	181	0.16	2	191	0	2	207	0.16	2	233	0.16
300	2	90	0.16	2	95	0	2	103	0.16	2	116	0.16
600	1	181	0.16	1	191	0	1	207	0.16	1	233	0.16
1200	1	90	0.16	1	95	0	1	103	0.16	1	116	0.16
2400	0	181	0.16	0	191	0	0	207	0.16	0	233	0.16
4800	0	90	0.16	0	95	0	0	103	0.16	0	116	0.16
9600	0	45	–0.93	0	47	0	0	51	0.16	0	58	–0.69
19200	0	22	–0.93	0	23	0	0	25	0.16	0	28	1.02
31250	0	11	0	0	14	–1.70	0	15	0	0	17	0.00
38400	0	10	3.57	0	11	0	0	12	0.16	0	14	–2.34

Table 11.4 Examples of Bit Rates and BRR Settings in Synchronous Mode

Bit Rate (bits/s)	φ (MHz)											
	2		4		8		10		16		18	
	n	N	n	N	n	N	n	N	n	N	n	N
110	3	70	—	—	—	—	—	—	—	—	—	—
250	2	124	2	249	3	124	—	—	3	249	—	—
500	1	249	2	124	2	249	—	—	3	124	3	140
1 k	1	124	1	249	2	124	—	—	2	249	3	69
2.5 k	0	199	1	99	1	199	1	249	2	99	2	112
5 k	0	99	0	199	1	99	1	124	1	199	1	224
10 k	0	49	0	99	0	199	0	249	1	99	1	112
25 k	0	19	0	39	0	79	0	99	0	159	0	179
50 k	0	9	0	19	0	39	0	49	0	79	0	89
100 k	0	4	0	9	0	19	0	24	0	39	0	44
250 k	0	1	0	3	0	7	0	9	0	15	0	17
500 k	0	0*	0	1	0	3	0	4	0	7	0	8
1 M			0	0*	0	1	—	—	0	3	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 transmission/reception not possible

Note: Settings with an error of 1% or less are recommended.

The BRR setting is calculated as follows:

Asynchronous mode:

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

Synchronous mode:

$$N = \frac{\phi}{8 \times 2^{2n-1} \times B} \times 10^6 - 1$$

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.

$$\text{Error (\%)} = \{\frac{\phi \times 10^6}{(N + 1) \times B \times 64 \times 2^{2n-1}} - 1\} \times 100$$

Table 11.5 indicates the maximum bit rates in asynchronous mode for various system clock frequencies. Tables 11.6 and 11.7 indicate the maximum bit rates with external clock input.

Table 11.5 Maximum Bit Rates for Various Frequencies (Asynchronous Mode)

		Settings	
ϕ (MHz)	**Maximum Bit Rate (bits/s)**	**n**	**N**
2	62500	0	0
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

Table 11.6 Maximum Bit Rates with External Clock Input (Asynchronous Mode)

ϕ (MHz)	External Input Clock (MHz)	Maximum Bit Rate (bits/s)
2	0.5000	31250
2.097152	0.5243	32768
2.4576	0.6144	38400
3	0.7500	46875
3.6864	0.9216	57600
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

Table 11.7 Maximum Bit Rates with External Clock Input (Synchronous Mode)

φ (MHz)	External Input Clock (MHz)	Maximum Bit Rate (bits/s)
2	0.3333	333333.3
4	0.6667	666666.7
6	1.0000	1000000.0
8	1.3333	1333333.3
10	1.6667	1666666.7
12	2.0000	2000000.0
14	2.3333	2333333.3
16	2.6667	2666666.7
18	3.0000	3000000.0

11.3 Operation

11.3.1 Overview

The SCI has an asynchronous mode in which characters are synchronized individually, and a synchronous mode in which communication is synchronized with clock pulses. Serial communication is possible in either mode. Asynchronous or synchronous mode and the communication format are selected in SMR, as shown in table 11.8. The SCI clock source is selected by the C/$\overline{A}$ bit in SMR and the CKE1 and CKE0 bits in SCR, as shown in table 11.9.

Asynchronous Mode:

- Data length is selectable: 7 or 8 bits.
- Parity and multiprocessor bits are selectable, and so is the stop bit length (1 or 2 bits). These selections determine the communication format and character length.
- In receiving, it is possible to detect framing errors, parity errors, overrun errors, and the break state.
- An internal or external clock can be selected as the SCI clock source.
  - When an internal clock is selected, the SCI operates using the on-chip baud rate generator, 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 frequency 16 times 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 rate generator, and outputs a serial clock signal to external devices.
  - When an external clock is selected, the SCI operates on the input serial clock. The on-chip baud rate generator is not used.

Table 11.8 SMR Settings and Serial Communication Formats

SMR Settings						SCI Communication Format			
Bit 7 C/$\overline{A}$	Bit 6 CHR	Bit 2 MP	Bit 5 PE	Bit 3 STOP	Mode	Data Length	Multi-processor Bit	Parity Bit	Stop Bit Length
0	0	0	0	0	Asynchronous mode	8-bit data	Absent	Absent	1 bit
0	0	0	0	1					2 bits
0	0	0	1	0				Present	1 bit
0	0	0	1	1					2 bits
0	1	0	0	0		7-bit data		Absent	1 bit
0	1	0	0	1					2 bits
0	1	0	1	0				Present	1 bit
0	1	0	1	1					2 bits
0	0	1	—	0	Asynchronous mode (multi-processor format)	8-bit data	Present	Absent	1 bit
0	0	1	—	1					2 bits
0	1	1	—	0		7-bit data			1 bit
0	1	1	—	1					2 bits
1	—	—	—	—	Synchronous mode	8-bit data	Absent		None

Table 11.9 SMR and SCR Settings and SCI Clock Source Selection

SMR	SCR Settings			SCI Communication Format	
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 pin
0	0	1			Outputs a clock with frequency matching the bit rate
0	1	0		External	Inputs a clock with frequency 16 times the bit rate
0	1	1			
1	0	0	Synchronous mode	Internal	Outputs the serial clock
1	0	1			
1	1	0		External	Inputs the serial clock
1	1	1			

11.3.2 Operation in Asynchronous Mode

In asynchronous mode each transmitted or received character begins with a start bit and ends with a stop bit. Serial communication is synchronized one character at a time.

The transmitting and receiving sections of the SCI are independent, so full-duplex communication is possible. The transmitter and receiver are both double buffered, so data can be written and read while transmitting and receiving are in progress, enabling continuous transmitting and receiving.

Figure 11.2 shows the general format of asynchronous serial communication. In asynchronous serial communication the communication line is normally held in the mark (high) state. The SCI monitors the line and starts serial communication when the line goes to the space (low) state, indicating a start bit. One serial character consists of a start bit (low), data (LSB first), parity bit (high or low), and stop bit (high), in that order.

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

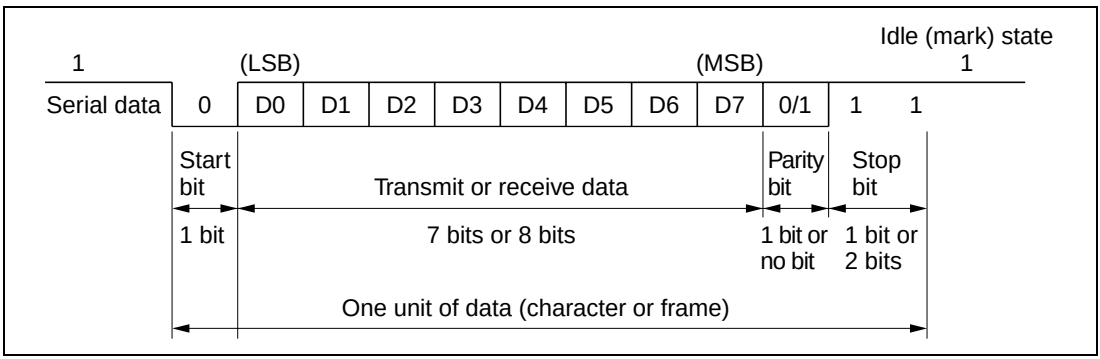


Figure 11.2 Data Format in Asynchronous Communication (Example: 8-Bit Data with Parity and 2 Stop Bits)

Communication Formats

Table 11.10 shows the 12 communication formats that can be selected in asynchronous mode. The format is selected by settings in SMR.

Table 11.10 Serial Communication Formats (Asynchronous Mode)

SMR Settings				Serial Communication Format and Frame Length											
CHR	PE	MP	STOP	1	2	3	4	5	6	7	8	9	10	11	12
0	0	0	0	S	8-bit data								STOP		
0	0	0	1	S	8-bit data								STOP	STOP	
0	1	0	0	S	8-bit data								P	STOP	
0	1	0	1	S	8-bit data								P	STOP	STOP
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	STOP	
0	—	1	0	S	8 bit data								MPB	STOP	
0	—	1	1	S	8 bit data								MPB	STOP	STOP
1	—	1	0	S	7-bit data							MPB	STOP		
1	—	1	1	S	7-bit data							MPB	STOP	STOP	

Legend:
S: Start bit
STOP: Stop bit
P: Parity bit
MPB: Multiprocessor bit

Clock

An internal clock generated by the on-chip baud rate generator or an external clock input from the SCK pin can be selected as the SCI transmit/receive clock. The clock source is selected by the C/$\overline{A}$ bit in SMR and bits CKE1 and CKE0 in SCR. See table 11.9.

When an external clock is input at the SCK pin, it must have a frequency equal to 16 times the desired bit rate.

When the SCI operates on an internal clock, it can output a clock signal at the SCK pin. The frequency of this output clock is equal to the bit rate. The phase is aligned as in figure 11.3 so that the rising edge of the clock occurs at the center of each transmit data bit.

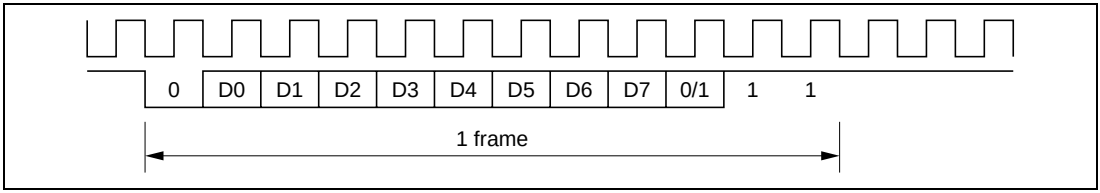


Figure 11.3 Phase Relationship between Output Clock and Serial Data (Asynchronous Mode)

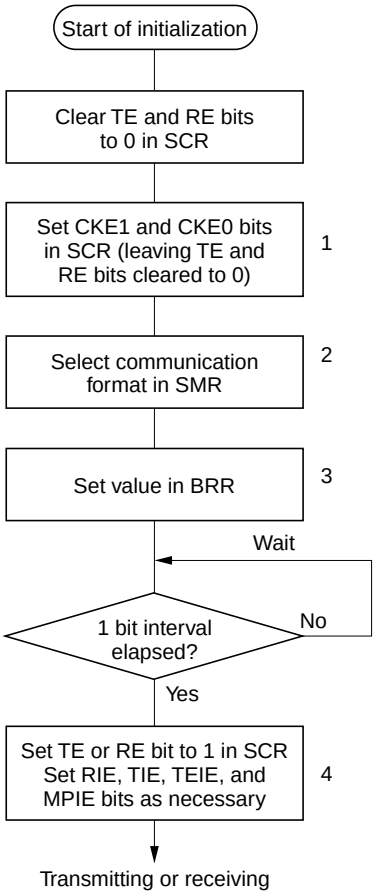
Transmitting and Receiving Data

SCI Initialization (Asynchronous Mode): Before transmitting or receiving, clear the TE and 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 0 before following the procedure given below. Clearing TE to 0 sets the TDRE flag to 1 and initializes TSR. Clearing RE to 0, however, does not initialize the RDRF, PER, FER, and ORER flags and RDR, which retain their previous contents.

When an external clock is used, the clock should not be stopped during initialization or subsequent operation. SCI operation becomes unreliable if the clock is stopped.

Figure 11.4 shows a sample flowchart for initializing the SCI.



1. Select the clock source in SCR. Clear the RIE, TIE, TEIE, MPIE, TE, and RE bits to 0. If clock output is selected in asynchronous mode, clock output starts immediately after the setting is made in SCR.
2. Select the communication format in SMR.
3. Write the value corresponding to the bit rate in BRR. This step is not necessary when an external clock is used.
4. Wait for at least the interval required to transmit or receive 1 bit, then set the TE or RE bit to 1 in SCR. Set the RIE, TIE, TEIE, and MPIE bits as necessary. Setting the TE or RE bit enables the SCI to use the TxD or RxD pin.

Figure 11.4 Sample Flowchart for SCI Initialization

Transmitting Serial Data (Asynchronous Mode): Figure 11.5 shows a sample flowchart for transmitting serial data and indicates the procedure to follow.

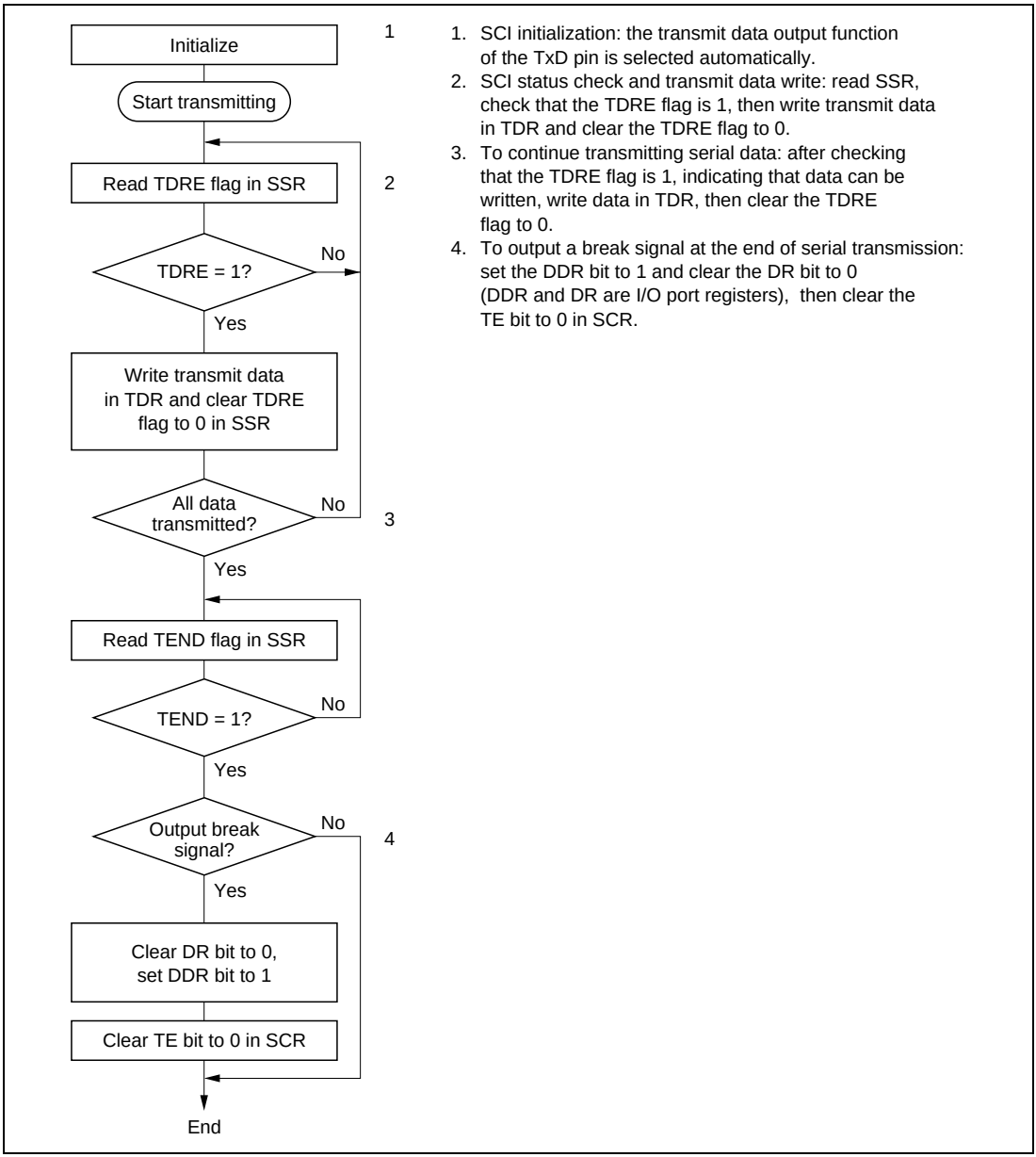


Figure 11.5 Sample Flowchart for Transmitting Serial Data

In transmitting serial data, the SCI operates as follows.

- The SCI monitors the TDRE flag in SSR. When the TDRE flag is cleared to 0 the SCI recognizes that TDR contains new data, and loads this data from TDR into TSR.
- After loading the data from TDR into TSR, the SCI sets the TDRE flag to 1 and starts transmitting. If the TIE bit is set to 1 in SCR, the SCI requests a transmit-data-empty interrupt (TXI) at this time.

  Serial transmit data is transmitted in the following order from the TxD pin:

  — Start bit: One 0 bit is output.

  — Transmit data: 7 or 8 bits are output, LSB first.

  — Parity bit or multiprocessor bit: One parity bit (even or odd parity) or one multiprocessor bit is output. Formats in which neither a parity bit nor a multiprocessor bit is output can also be selected.

  — Stop bit: One or two 1 bits (stop bits) are output.

  — Mark state: Output of 1 continues until the start bit of the next transmit data.
- The SCI checks the TDRE flag when it outputs the stop bit. If the TDRE flag is 0, the SCI loads new data from TDR into TSR, outputs the stop bit, then begins serial transmission of the next frame. If the TDRE flag is 1, the SCI sets the TEND flag to 1 in SSR, outputs the stop bit, then continues output of 1 in the mark state. If the TEIE bit is set to 1 in SCR, a transmit-end interrupt (TEI) is requested at this time.

Figure 11.6 shows an example of SCI transmit operation in asynchronous mode.

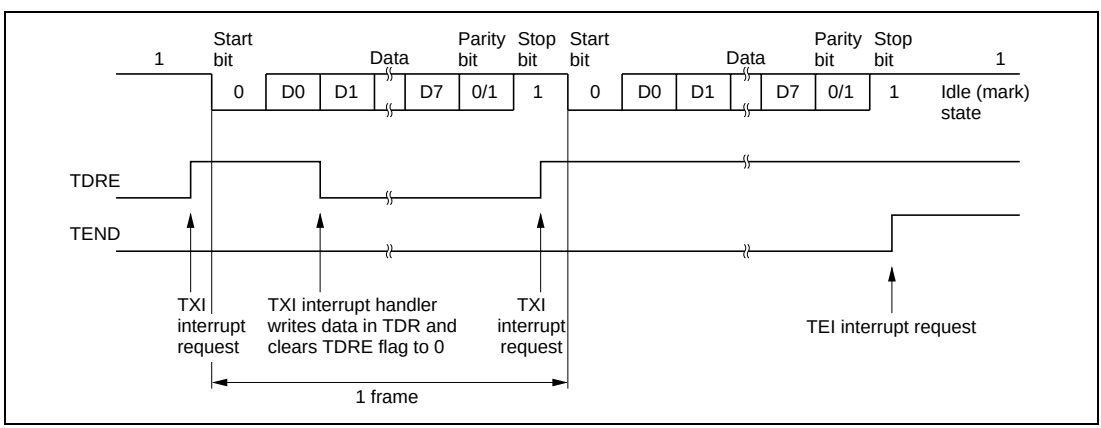
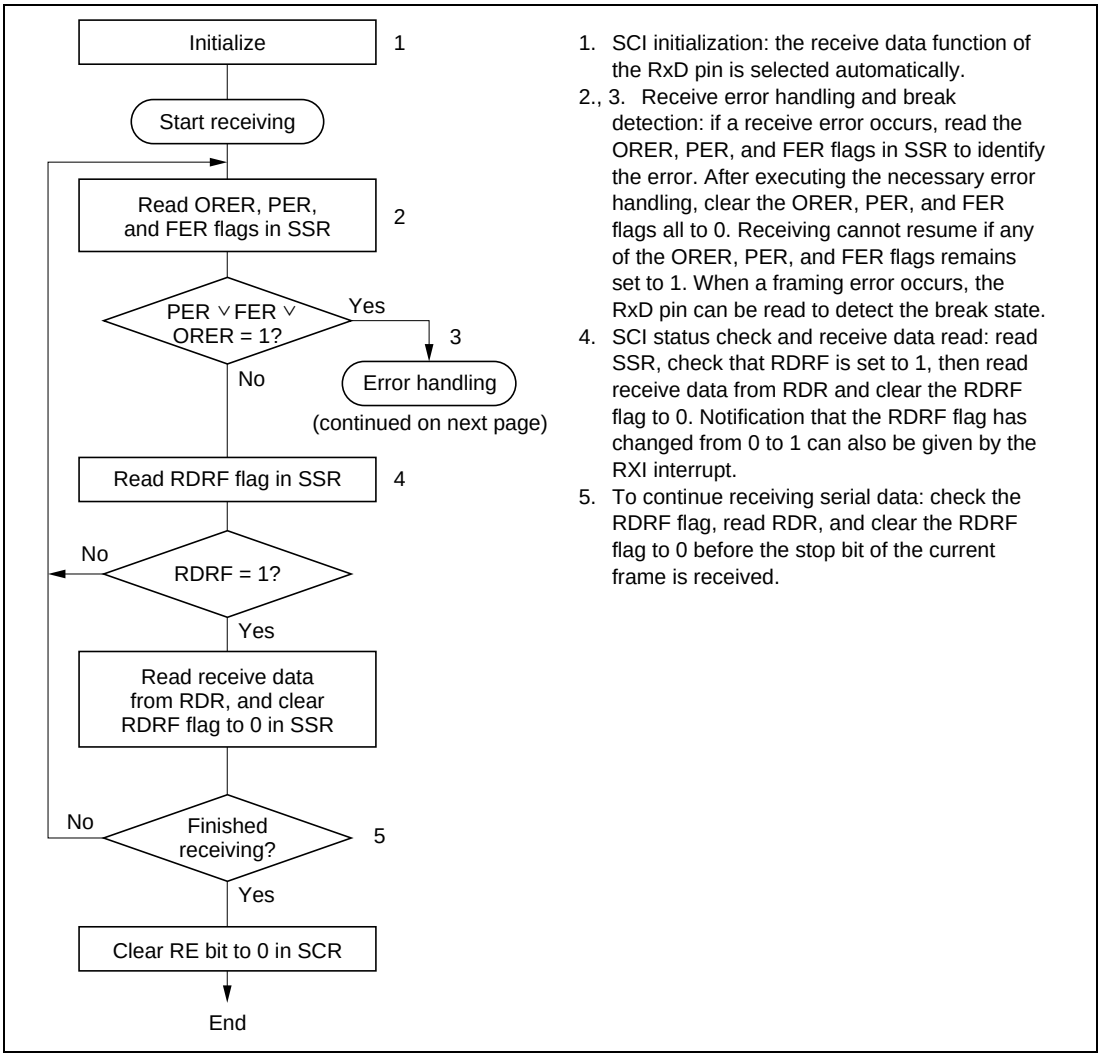


Figure 11.6 Example of SCI Transmit Operation in Asynchronous Mode (8-Bit Data with Parity and 1 Stop Bit)

Receiving Serial Data (Asynchronous Mode): Figure 11.7 shows a sample flowchart for receiving serial data and indicates the procedure to follow.



1. SCI initialization: the receive data function of the RxD pin is selected automatically.
2., 3. Receive error handling and break detection: if a receive error occurs, read the ORER, PER, and FER flags in SSR to identify the error. After executing the necessary error handling, clear the ORER, PER, and FER flags all to 0. Receiving cannot resume if any of the ORER, PER, and FER flags remains set to 1. When a framing error occurs, the RxD pin can be read to detect the break state.
4. SCI status check and receive data read: read SSR, check that RDRF is set to 1, then read receive data from RDR and clear the RDRF flag to 0. Notification that the RDRF flag has changed from 0 to 1 can also be given by the RXI interrupt.
5. To continue receiving serial data: check the RDRF flag, read RDR, and clear the RDRF flag to 0 before the stop bit of the current frame is received.

Figure 11.7 Sample Flowchart for Receiving Serial Data (1)

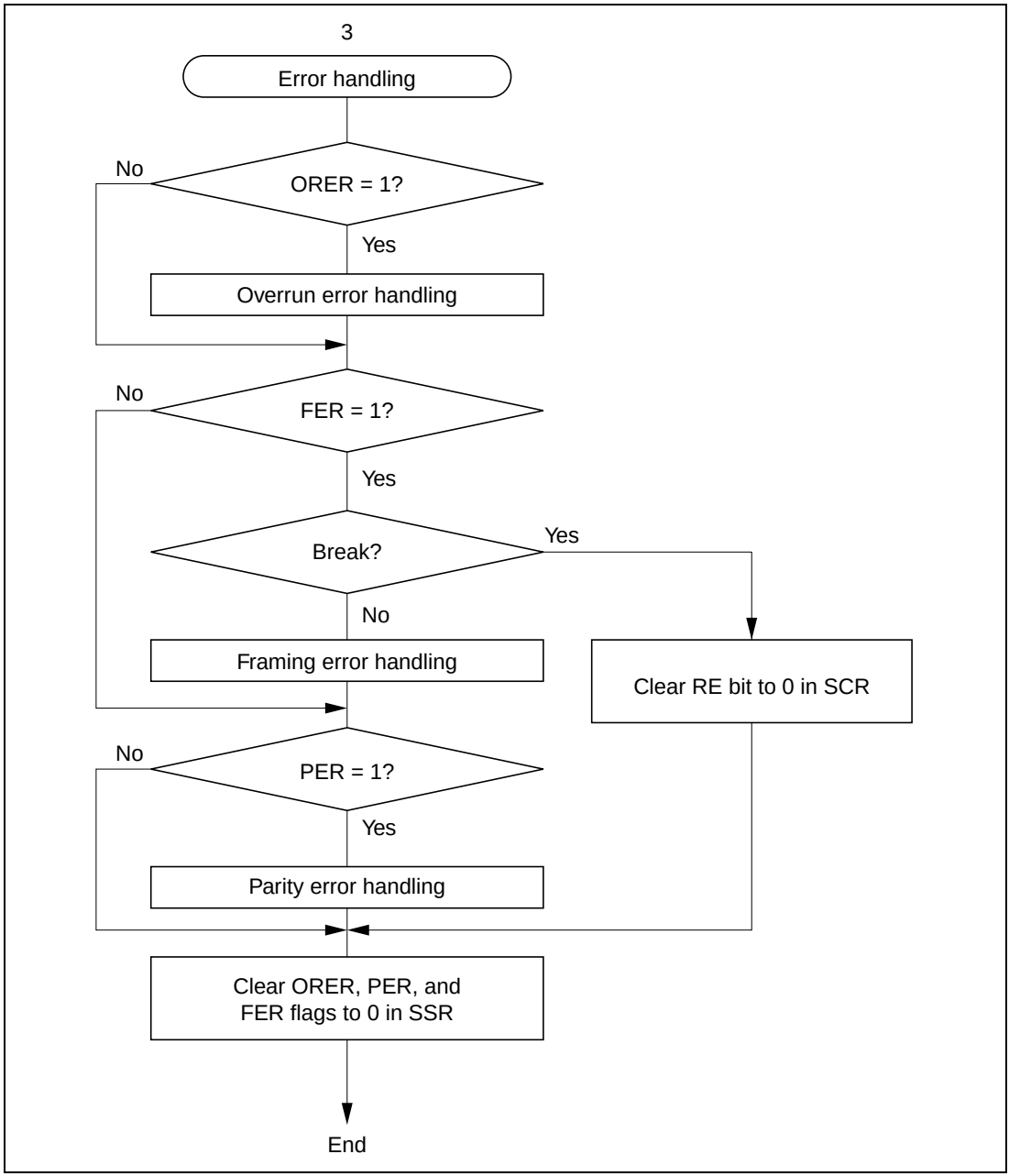


Figure 11.7 Sample Flowchart for Receiving Serial Data (2)

In receiving, the SCI operates as follows.

- The SCI monitors the receive data line. When it detects a start bit, the SCI synchronizes internally and starts receiving.
- Receive data is stored in RSR in order from LSB to MSB.
- The parity bit and stop bit are received.

  After receiving data, the SCI makes the following checks:

  — Parity check: The number of 1s in the receive data must match the even or odd parity setting of the O/$\overline{E}$ bit in SMR.

  — Stop bit check: The stop bit value must be 1. If there are two stop bits, only the first stop bit is checked.

  — Status check: The RDRF flag must be 0 so that receive data can be transferred from RSR into RDR.

  If these checks all pass, the RDRF flag is set to 1 and the received data is stored in RDR. If one of the checks fails (receive error)*, the SCI operates as indicated in table 11.11.

Note: * When a receive error occurs, further receiving is disabled. In receiving, the RDRF flag is not set to 1. Be sure to clear the error flags.

- When the RDRF flag is set to 1, if the RIE bit is set to 1 in SCR, a receive-data-full interrupt (RXI) is requested. If the ORER, PER, or FER flag is set to 1 and the RIE bit in SCR is also set to 1, a receive-error interrupt (ERI) is requested.

Table 11.11 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 transferred from RSR to RDR
Framing error	FER	Stop bit is 0	Receive data transferred from RSR to RDR
Parity error	PER	Parity of receive data differs from even/odd parity setting in SMR	Receive data transferred from RSR to RDR

Figure 11.8 shows an example of SCI receive operation in asynchronous mode.

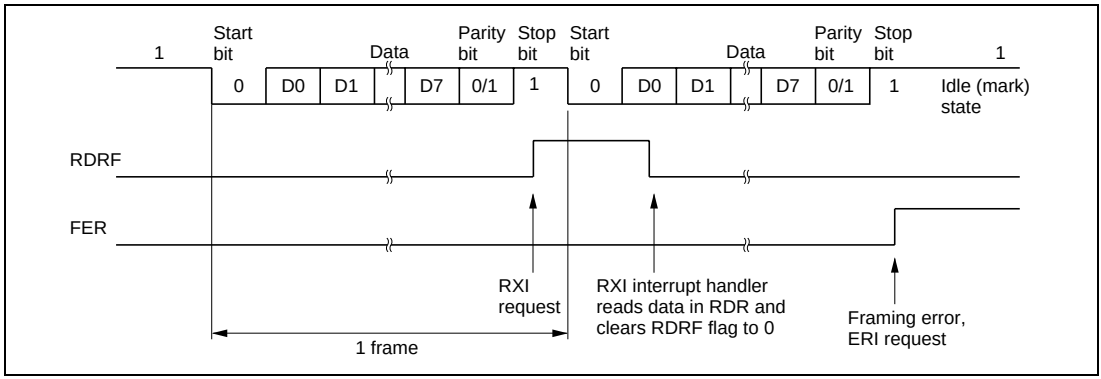


Figure 11.8 Example of SCI Receive Operation (8-Bit Data with Parity and One Stop Bit)

11.3.3 Multiprocessor Communication

The multiprocessor communication function enables several processors to share a single serial communication line. The processors communicate in asynchronous mode using a format with an additional multiprocessor bit (multiprocessor format).

In multiprocessor communication, each receiving processor is addressed by an ID. A serial communication cycle consists of an ID-sending cycle that identifies the receiving processor, and a data-sending cycle. The multiprocessor bit distinguishes ID-sending cycles from data-sending cycles.

The transmitting processor starts by sending the ID of the receiving processor with which it wants to communicate as data with the multiprocessor bit set to 1. Next the transmitting processor sends transmit data with the multiprocessor bit cleared to 0.

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

Figure 11.9 shows an example of communication among different processors using a multiprocessor format.

Communication Formats

Four formats are available. Parity-bit settings are ignored when a multiprocessor format is selected. For details see table 11.11.

Clock

See the description of asynchronous mode.

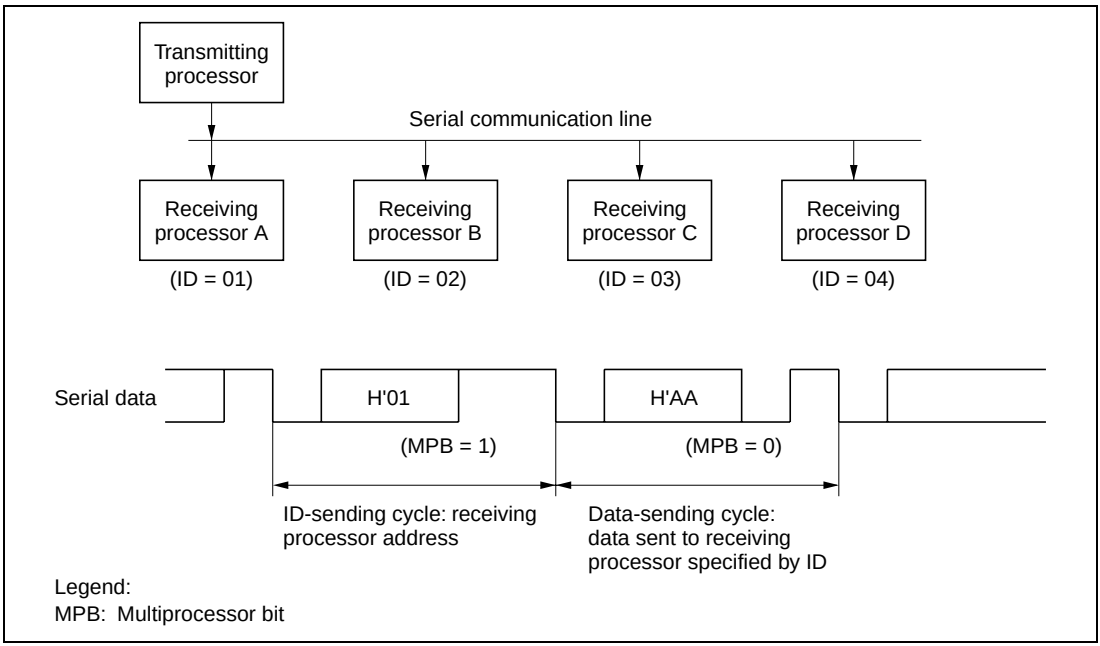


Figure 11.9 Example of Communication among Processors using Multiprocessor Format (Sending Data H'AA to Receiving Processor A)

Transmitting and Receiving Data

Transmitting Multiprocessor Serial Data: Figure 11.10 shows a sample flowchart for transmitting multiprocessor serial data and indicates the procedure to follow.

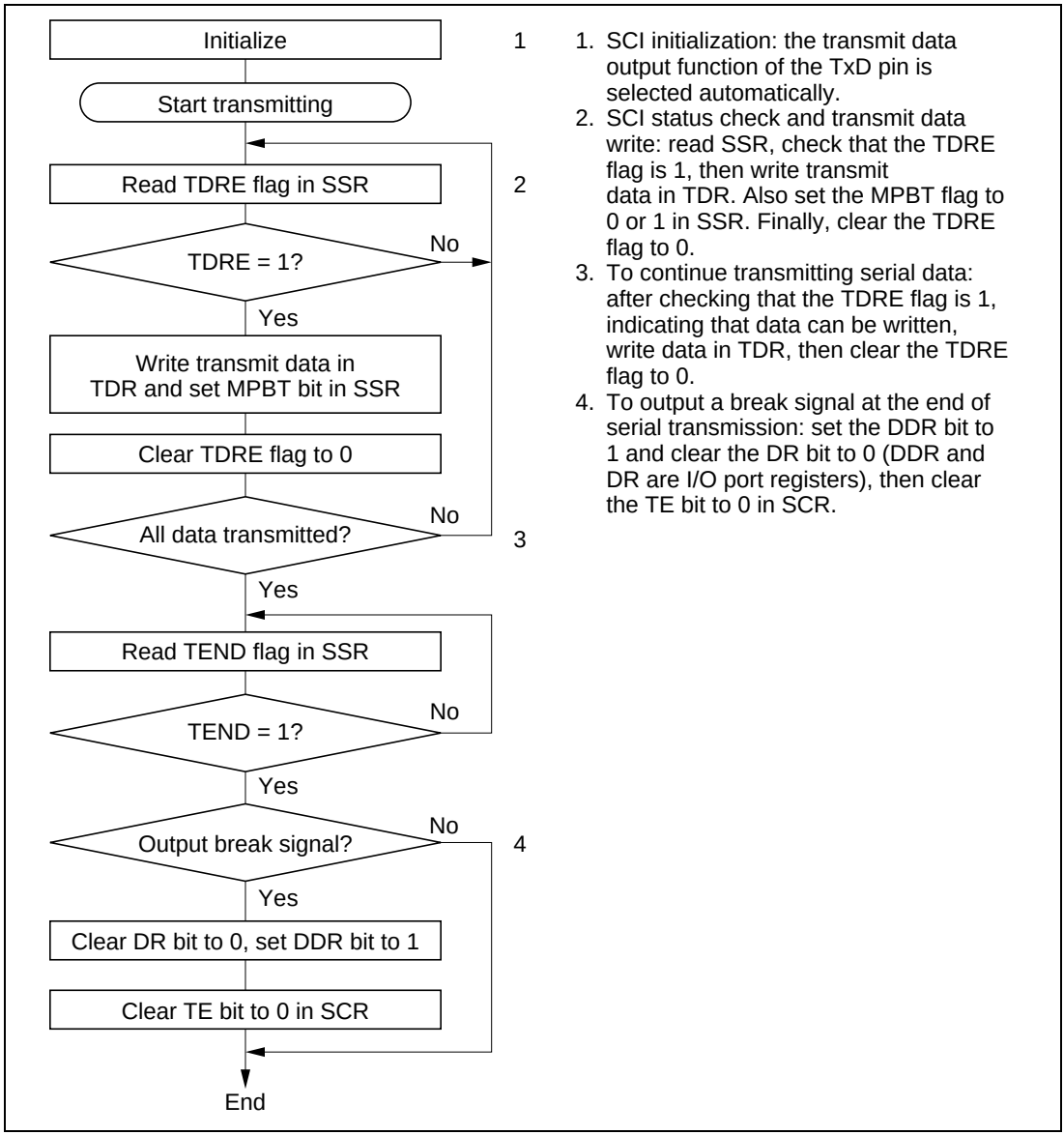


Figure 11.10 Sample Flowchart for Transmitting Multiprocessor Serial Data

In transmitting serial data, the SCI operates as follows.

- The SCI monitors the TDRE flag in SSR. When the TDRE flag is cleared to 0 the SCI recognizes that TDR contains new data, and loads this data from TDR into TSR.
- After loading the data from TDR into TSR, the SCI sets the TDRE flag to 1 and starts transmitting. If the TIE bit in SCR is set to 1, the SCI requests a transmit-data-empty interrupt (TXI) at this time.

  Serial transmit data is transmitted in the following order from the TxD pin:

  — Start bit: One 0 bit is output.
  — Transmit data: 7 or 8 bits are output, LSB first.
  — Multiprocessor bit: One multiprocessor bit (MPBT value) is output.
  — Stop bit: One or two 1 bits (stop bits) are output.
  — Mark state: Output of 1 bits continues until the start bit of the next transmit data.
- The SCI checks the TDRE flag when it outputs the stop bit. If the TDRE flag is 0, the SCI loads data from TDR into TSR, outputs the stop bit, then begins serial transmission of the next frame. If the TDRE flag is 1, the SCI sets the TEND flag in SSR to 1, outputs the stop bit, then continues output of 1 bits in the mark state. If the TEIE bit is set to 1 in SCR, a transmit-end interrupt (TEI) is requested at this time.

Figure 11.11 shows an example of SCI transmit operation using a multiprocessor format.

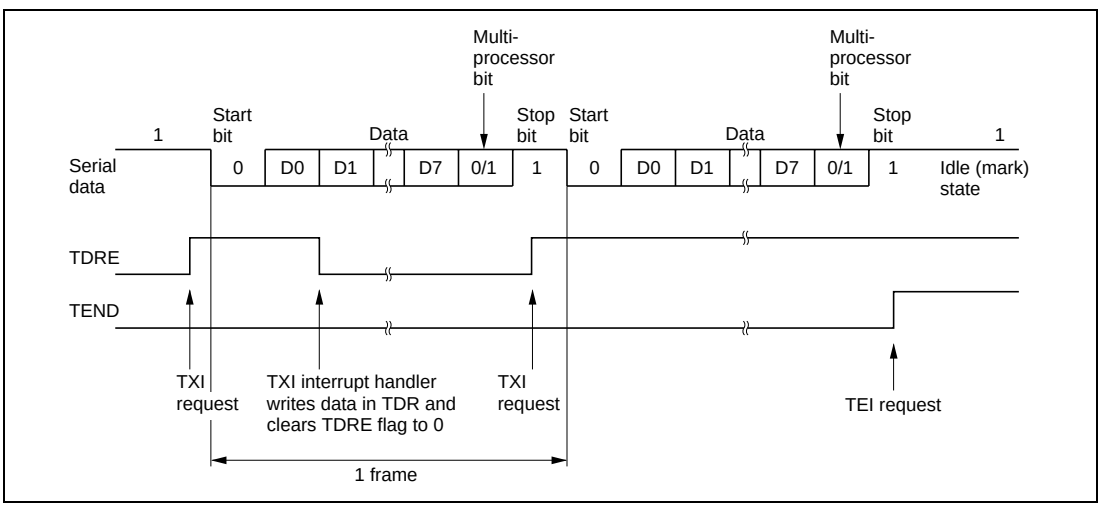


Figure 11.11 Example of SCI Transmit Operation (8-Bit Data with Multiprocessor Bit and One Stop Bit)

Receiving Multiprocessor Serial Data: Figure 11.12 shows a sample flowchart for receiving multiprocessor serial data and indicates the procedure to follow.

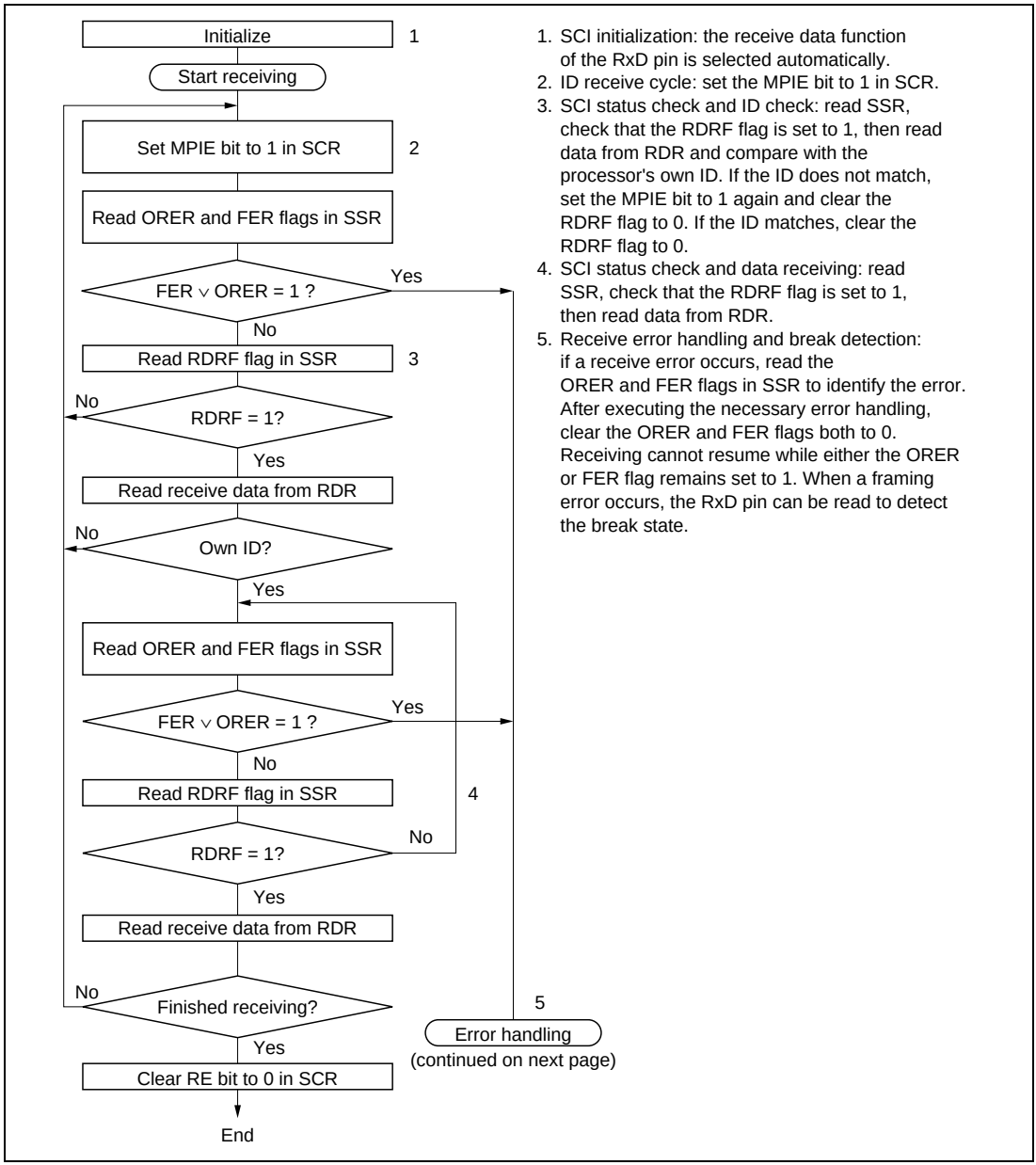


Figure 11.12 Sample Flowchart for Receiving Multiprocessor Serial Data (1)

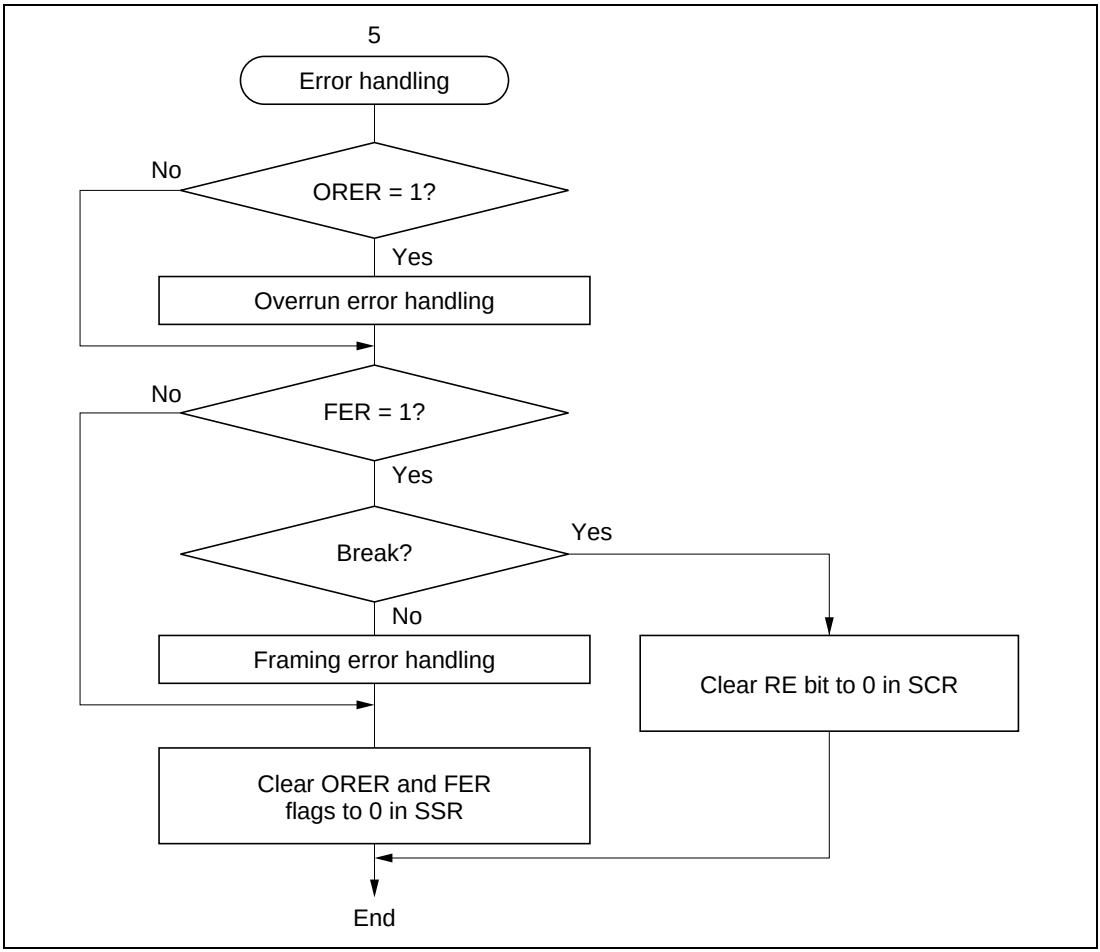


Figure 11.12 Sample Flowchart for Receiving Multiprocessor Serial Data (2)

Figure 11.13 shows an example of SCI receive operation using a multiprocessor format.

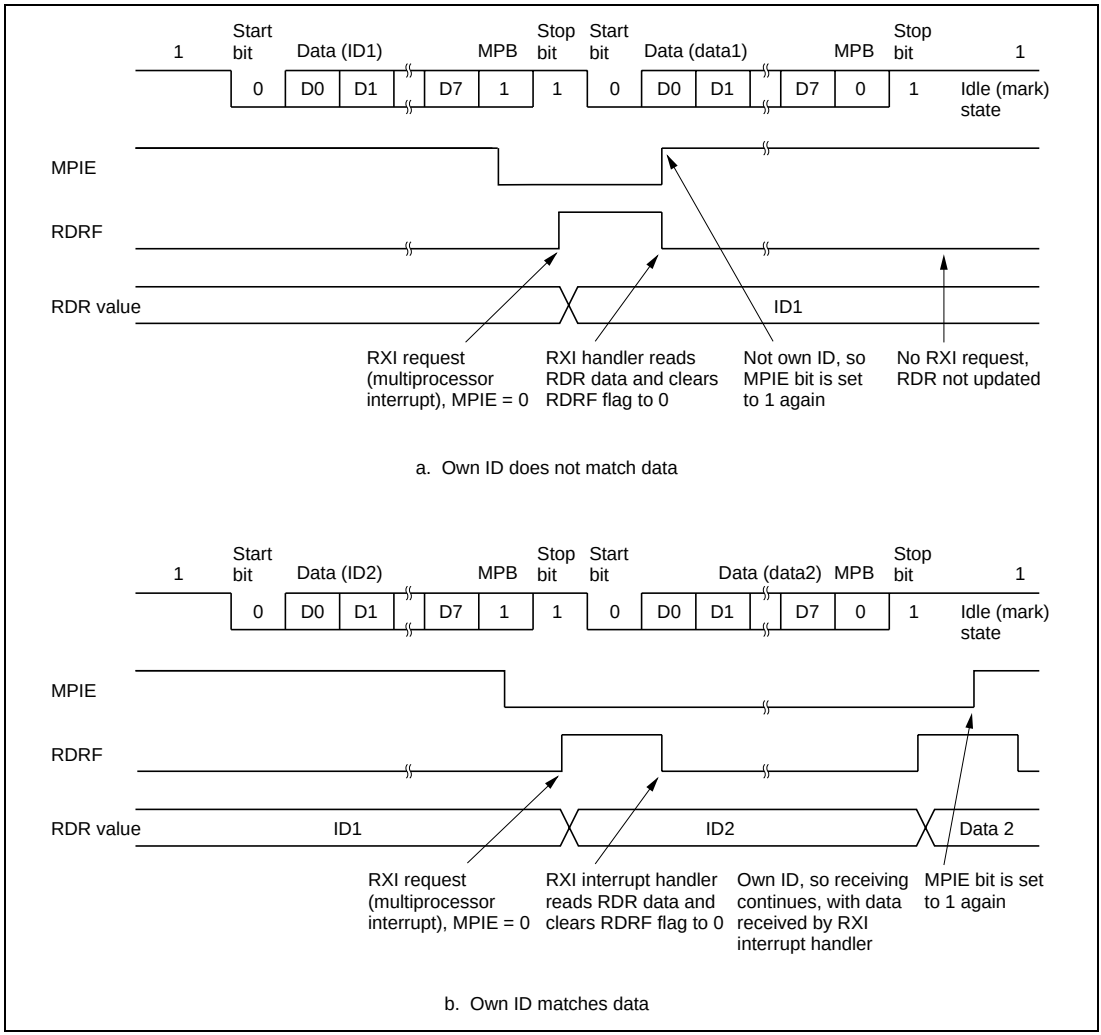


Figure 11.13 Example of SCI Receive Operation (8-Bit Data with Multiprocessor Bit and One Stop Bit)

11.3.4 Synchronous Operation

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

The SCI transmitter and receiver share the same clock but are otherwise independent, so full duplex communication is possible. The transmitter and receiver are also double buffered, so continuous transmitting or receiving is possible by reading or writing data while transmitting or receiving is in progress.

Figure 11.14 shows the general format in synchronous serial communication.

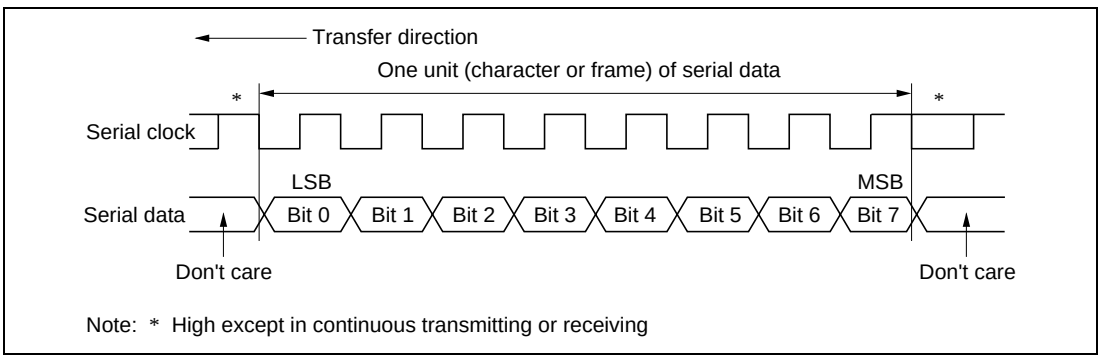


Figure 11.14 Data Format in Synchronous Communication

In synchronous serial communication, each data bit is placed on the communication line from one falling edge of the serial clock to the next. Data is guaranteed valid at the rise of the serial clock. In each character, the serial data bits are transmitted in order from LSB (first) to MSB (last). After output of the MSB, the communication line remains in the state of the MSB. In synchronous mode the SCI receives data by synchronizing with the rise of the serial clock.

Communication Format

The data length is fixed at 8 bits. No parity bit or multiprocessor bit can be added.

Clock

An internal clock generated by the on-chip baud rate generator or an external clock input from the SCK pin can be selected by clearing or setting the CKE1 and CKE0 bits in SCR and the C/$\overline{A}$ bit in SMR. See table 11.9. When the SCI operates on an internal clock, it outputs the clock signal at the SCK pin. Eight clock pulses are output per transmitted or received character. When the SCI is not transmitting or receiving, the clock signal remains in the high state.

Transmitting and Receiving Data

SCI Initialization (Synchronous Mode): Before transmitting or receiving, clear the TE and 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 0 before following the procedure given below. Clearing the TE bit to 0 sets the TDRE flag to 1 and initializes TSR. Clearing the RE bit to 0, however, does not initialize the RDRF, PER, FER, and ORE flags and RDR, which retain their previous contents.

Figure 11.15 shows a sample flowchart for initializing the SCI.

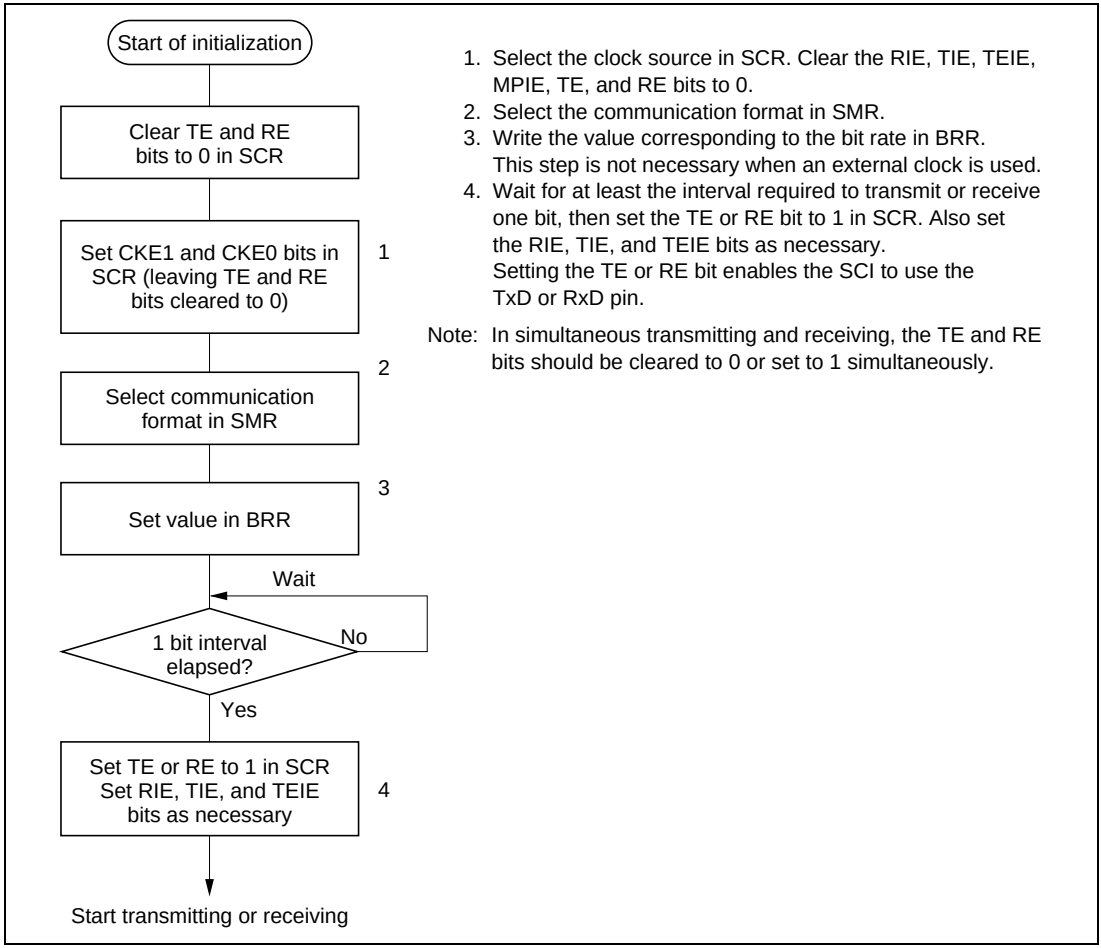


Figure 11.15 Sample Flowchart for SCI Initialization

Transmitting Serial Data (Synchronous Mode): Figure 11.16 shows a sample flowchart for transmitting serial data and indicates the procedure to follow.

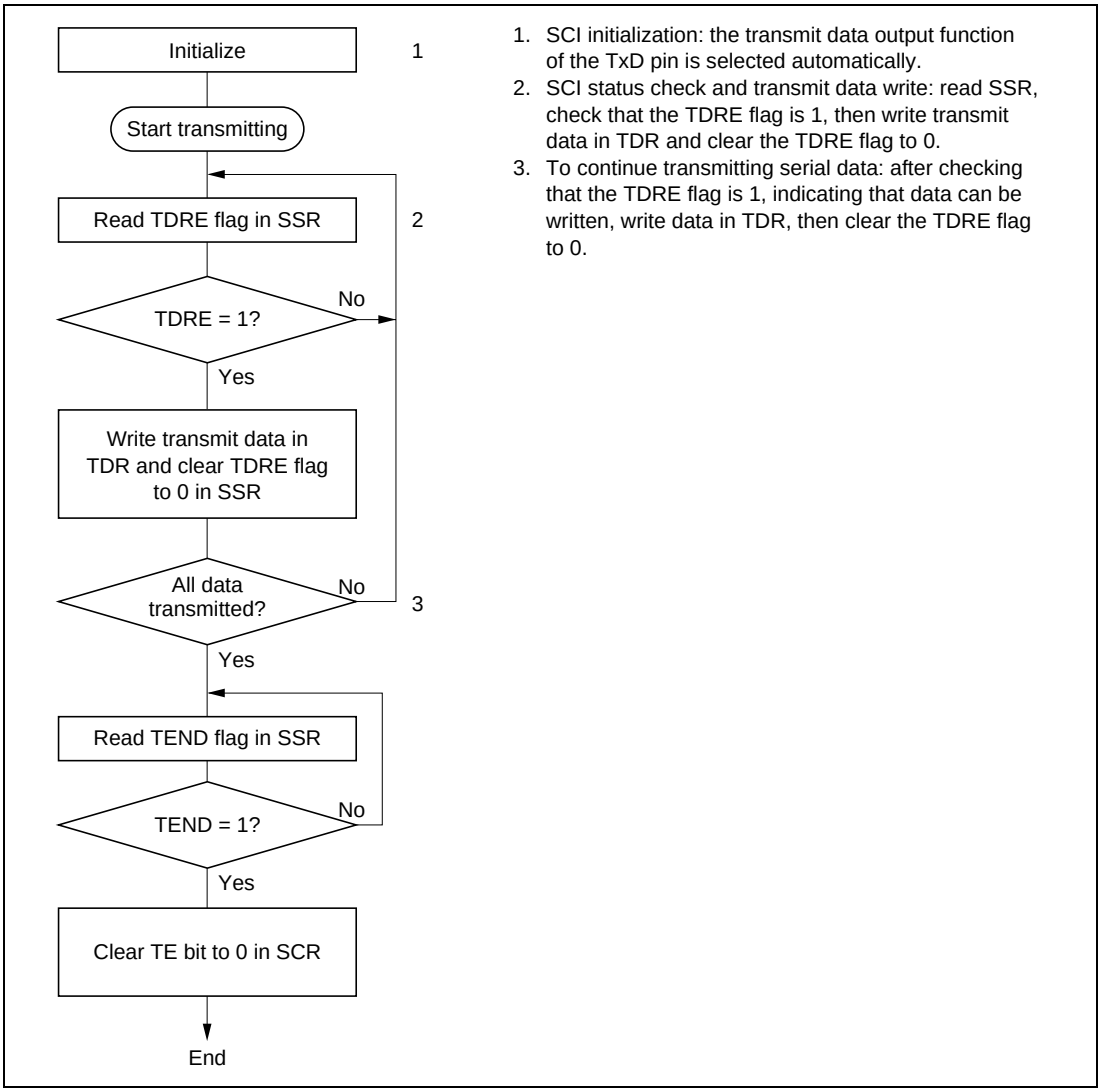


Figure 11.16 Sample Flowchart for Serial Transmitting

In transmitting serial data, the SCI operates as follows.

- The SCI monitors the TDRE flag in SSR. When the TDRE flag is cleared to 0 the SCI recognizes that TDR contains new data, and loads this data from TDR into TSR.
- After loading the data from TDR into TSR, the SCI sets the TDRE flag to 1 and starts transmitting. If the TIE bit is set to 1 in SCR, the SCI requests a transmit-data-empty interrupt (TXI) at this time.

  If clock output is selected, the SCI outputs eight serial clock pulses. If an external clock source is selected, the SCI outputs data in synchronization with the input clock. Data is output from the TxD pin in order from LSB (bit 0) to MSB (bit 7).
- The SCI checks the TDRE flag when it outputs the MSB (bit 7). If the TDRE flag is 0, the SCI loads data from TDR into TSR and begins serial transmission of the next frame. If the TDRE flag is 1, the SCI sets the TEND flag to 1 in SSR, and after transmitting the MSB, holds the TxD pin in the MSB state. If the TEIE bit in SCR is set to 1, a transmit-end interrupt (TEI) is requested at this time.
- After the end of serial transmission, the SCK pin is held in a constant state.

Figure 11.17 shows an example of SCI transmit operation.

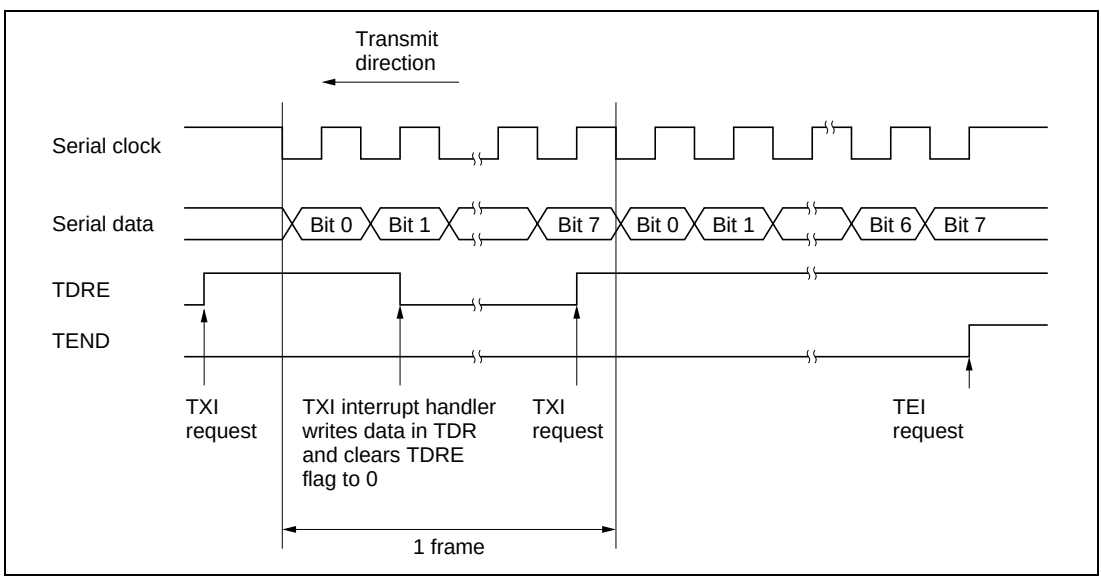


Figure 11.17 Example of SCI Transmit Operation

Receiving Serial Data (Synchronous Mode): Figure 11.18 shows a sample flowchart for receiving serial data and indicates the procedure to follow. When switching from asynchronous mode to synchronous mode, make sure that the ORER, PER, and FER flags are cleared to 0. If the FER or PER flag is set to 1 the RDRF flag will not be set and both transmitting and receiving will be disabled.

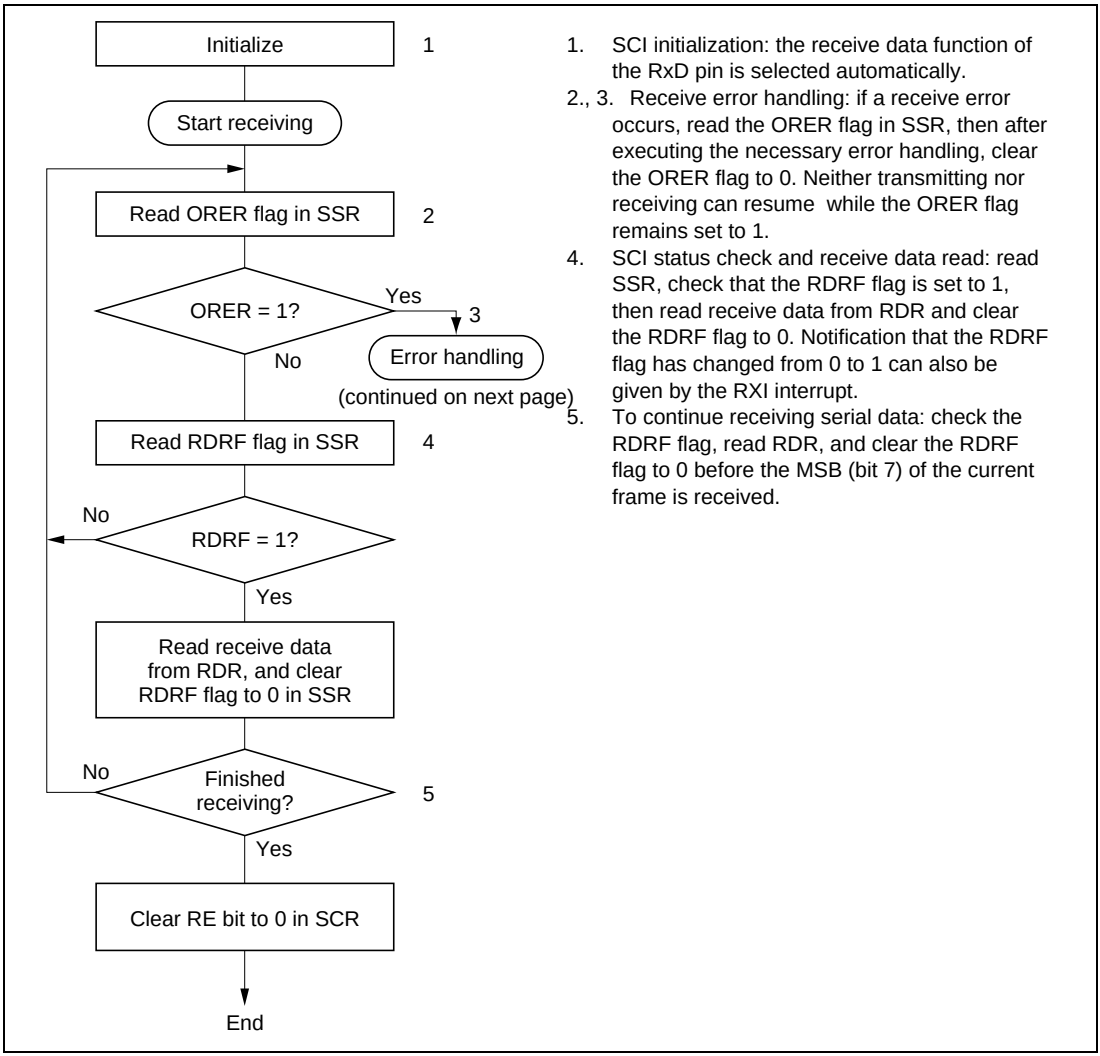


Figure 11.18 Sample Flowchart for Serial Receiving (1)

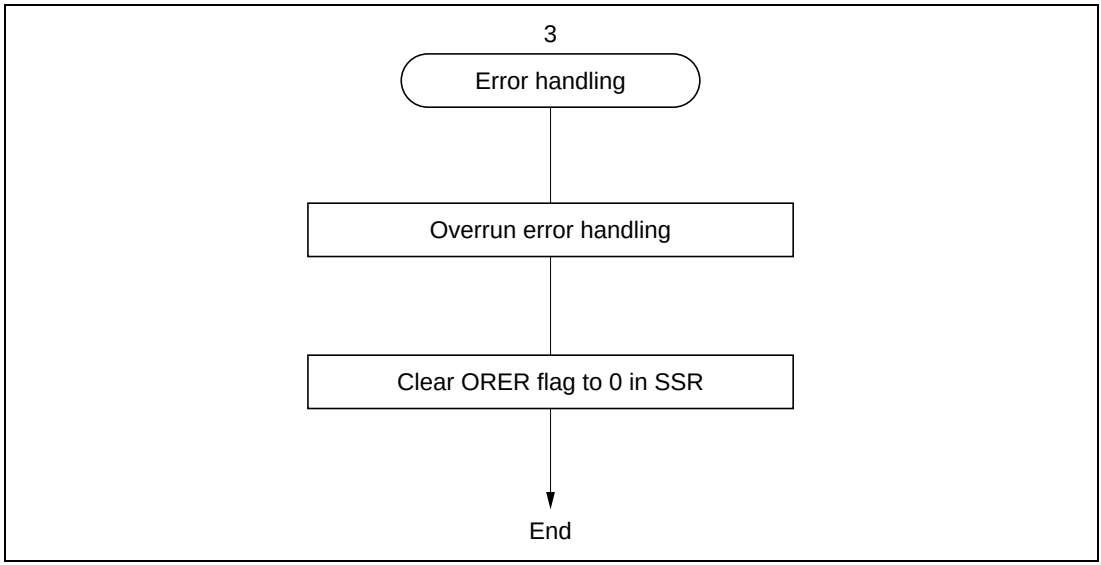


Figure 11.18 Sample Flowchart for Serial Receiving (2)

In receiving, the SCI operates as follows.

- The SCI synchronizes with serial clock input or output and initializes internally.
- Receive data is stored in RSR in order from LSB to MSB.

  After receiving the data, the SCI checks that the RDRF flag is 0 so that receive data can be transferred from RSR to RDR. If this check passes, the RDRF flag is set to 1 and the received data is stored in RDR. If the check does not pass (receive error), the SCI operates as indicated in table 11.11. If any receive error is detected, the subsequent data transmission/reception is disabled.
- After setting the RDRF flag to 1, if the RIE bit is set to 1 in SCR, the SCI requests a receive-data-full interrupt (RXI). If the ORER flag is set to 1 and the RIE bit in SCR is also set to 1, the SCI requests a receive-error interrupt (ERI).

Figure 11.19 shows an example of SCI receive operation.

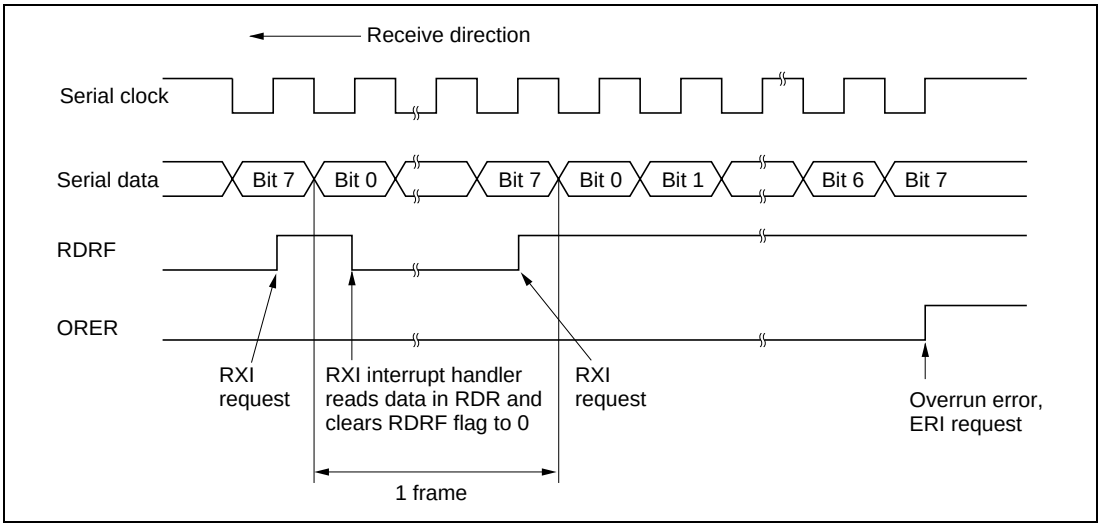
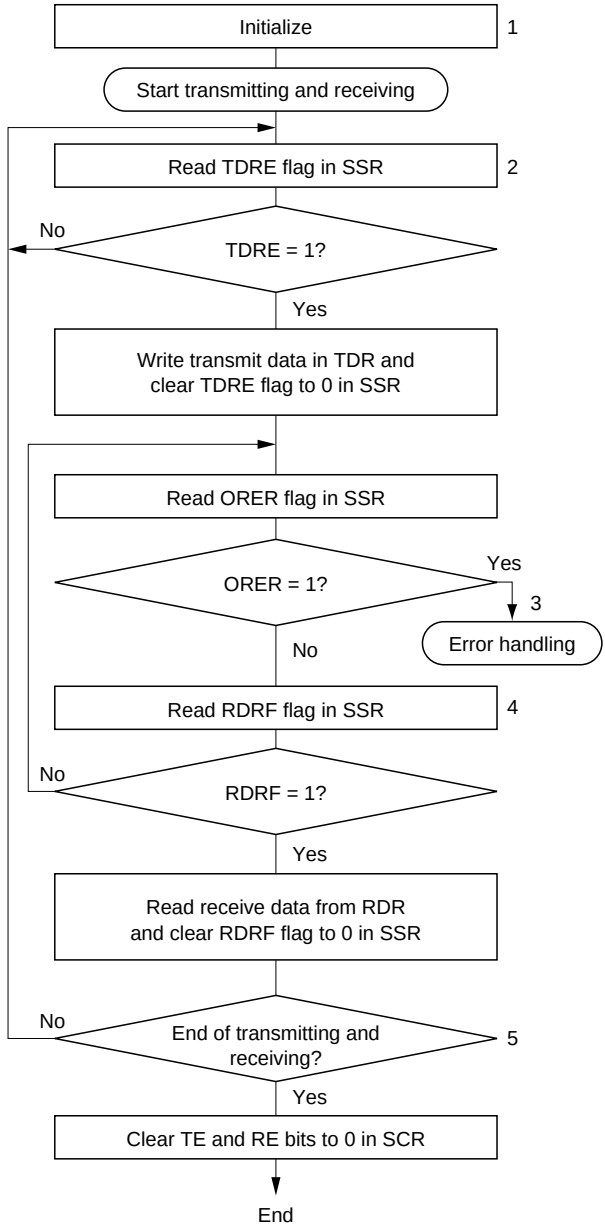


Figure 11.19 Example of SCI Receive Operation

Transmitting and Receiving Serial Data Simultaneously (Synchronous Mode): Figure 11.20 shows a sample flowchart for transmitting and receiving serial data simultaneously and indicates the procedure to follow.



Note: When switching from transmitting or receiving to simultaneous transmitting and receiving, clear the TE and RE bits both to 0, then set the TE and RE bits both to 1.

1. SCI initialization: the transmit data output function of the TxD pin and receive data input function of the RxD pin are selected, enabling simultaneous transmitting and receiving.
2. SCI status check and transmit data write: read SSR, check that the TDRE flag is 1, then write transmit data in TDR and clear the TDRE flag to 0.
   Notification that the TDRE flag has changed from 0 to 1 can also be given by the TXI interrupt.
3. Receive error handling: if a receive error occurs, read the ORER flag in SSR, then after executing the necessary error handling, clear the ORER flag to 0.
   Neither transmitting nor receiving can resume while the ORER flag remains set to 1.
4. SCI status check and receive data read: read SSR, check that the RDRF flag is 1, then read receive data from RDR and clear the RDRF flag to 0. Notification that the RDRF flag has changed from 0 to 1 can also be given by the RXI interrupt.
5. To continue transmitting and receiving serial data: check the RDRF flag, read RDR, and clear the RDRF flag to 0 before the MSB (bit 7) of the current frame is received. Also check that the TDRE flag is set to 1, indicating that data can be written, write data in TDR, then clear the TDRE flag to 0 before the MSB (bit 7) of the current frame is transmitted.

Figure 11.20 Sample Flowchart for Serial Transmitting

11.4 SCI Interrupts

The SCI has four interrupt request sources: TEI (transmit-end interrupt), ERI (receive-error interrupt), RXI (receive-data-full interrupt), and TXI (transmit-data-empty interrupt). Table 11.12 lists the interrupt sources and indicates their priority. These interrupts can be enabled and disabled by the TIE, RIE, and TEIE bits in SCR. Each interrupt request is sent separately to the interrupt controller.

The TXI interrupt is requested when the TDRE flag is set to 1 in SSR. The TEI interrupt is requested when the TEND flag is set to 1 in SSR.

The RXI interrupt is requested when the RDRF flag is set to 1 in SSR. The ERI interrupt is requested when the ORER, PER, or FER flag is set to 1 in SSR.

Table 11.12 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

11.5 Usage Notes

Note the following points when using the SCI.

TDR Write and TDRE Flag

The TDRE flag in SSR is a status flag indicating the loading of transmit data from TDR into TSR. The SCI sets the TDRE flag to 1 when it transfers data from TDR to TSR.

Data can be written into TDR regardless of the state of the TDRE flag. If new data is written in TDR when the TDRE flag is 0, the old data stored in TDR will be lost because this data has not yet been transferred to TSR. Before writing transmit data in TDR, be sure to check that the TDRE flag is set to 1.

Simultaneous Multiple Receive Errors

Table 11.13 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 RDR, so receive data is lost.

Table 11.13 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	○	Framing error
0	0	0	1	○	Parity error
1	1	1	0	×	Overrun error + framing error
1	1	0	1	×	Overrun error + parity error
0	0	1	1	○	Framing error + parity error
1	1	1	1	×	Overrun error + framing error + parity error

Notes: ○: Receive data is transferred from RSR to RDR.
×: Receive data is not transferred from RSR to RDR.

Break Detection and Processing

Break signals can be detected by reading the RxD pin directly when a framing error (FER) is detected. In the break state the input from the RxD pin consists of all 0s, so the FER flag is set and the parity error flag (PER) may also be set. In the break state the SCI receiver continues to operate, so if the FER flag is cleared to 0 it will be set to 1 again.

Sending a Break Signal

When the TE bit is cleared to 0 the TxD pin becomes an I/O port, the level and direction (input or output) of which are determined by DR and DDR bits. This feature can be used to send a break signal.

After the serial transmitter is initialized, the DR value substitutes for the mark state until the TE bit is set to 1 (the TxD pin function is not selected until the TE bit is set to 1). The DDR and DR 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 TE bit to 0. When the TE bit is cleared to 0 the transmitter is initialized, regardless of its current state, so the TxD pin becomes an input/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 transmitting, even if the TDRE flag is cleared to 0. Be sure to clear the receive error flags to 0 when starting to 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 frequency. In receiving, the SCI synchronizes internally with the fall of the start bit, which it samples on the base clock. Receive data is latched at the rising edge of the eighth base clock pulse. See figure 11.21.

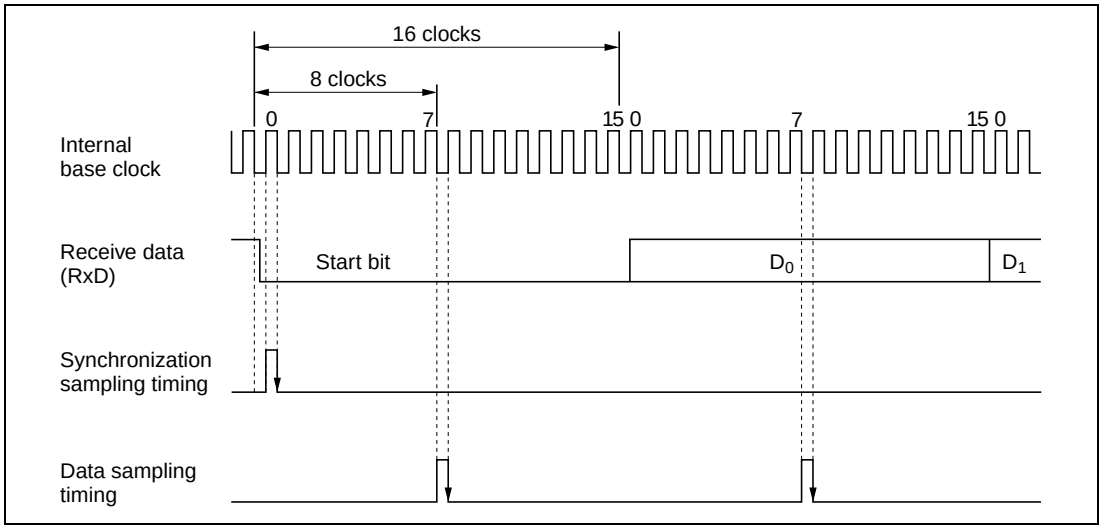


Figure 11.21 Receive Data Sampling Timing in Asynchronous Mode

The receive margin in asynchronous mode can therefore be expressed as shown in equation (1).

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

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 equation (2).

When 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 design is 20% to 30%.

Restrictions in Synchronous Mode

When data transmission is performed using an external clock source as the serial clock, an interval of at least 5 states is necessary between clearing the TDRE flag in SSR and the start (falling edge) of the first transmit clock pulse corresponding to each frame (figure 11.22). This interval is also necessary when performing continuous transmission. If this condition is not satisfied, an operation error may occur.

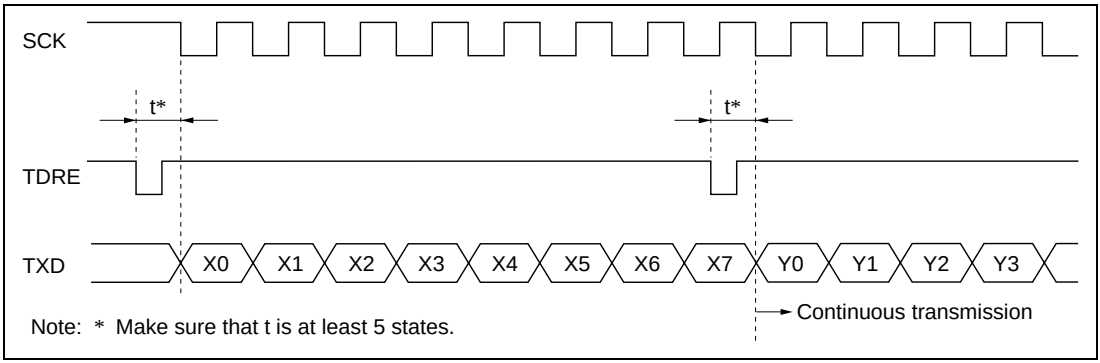


Figure 11.22 Transmission in Synchronous Mode (Example)

Restrictions when Switching from SCK Pin to Port Function in Synchronous SCI

1. Problem in Operation

After setting DDR and DR to 1 and using synchronous SCI clock output, when the SCK pin is switched to the port function at the end of transmission, a low-level signal is output for one half-cycle before the port output state is established.

When switching to the port function by making the following settings while DDR = 1, DR = 1, C/$\overline{A}$ = 1, CKE1 = 0, CKE0 = 0, and TE = 1, low-level output occurs for one half-cycle.

(1) End of serial data transmission

(2) TE bit = 0

(3) C/$\overline{A}$ bit = 0 ... switchover to port output

(4) Occurrence of low-level output (see figure 11.23)

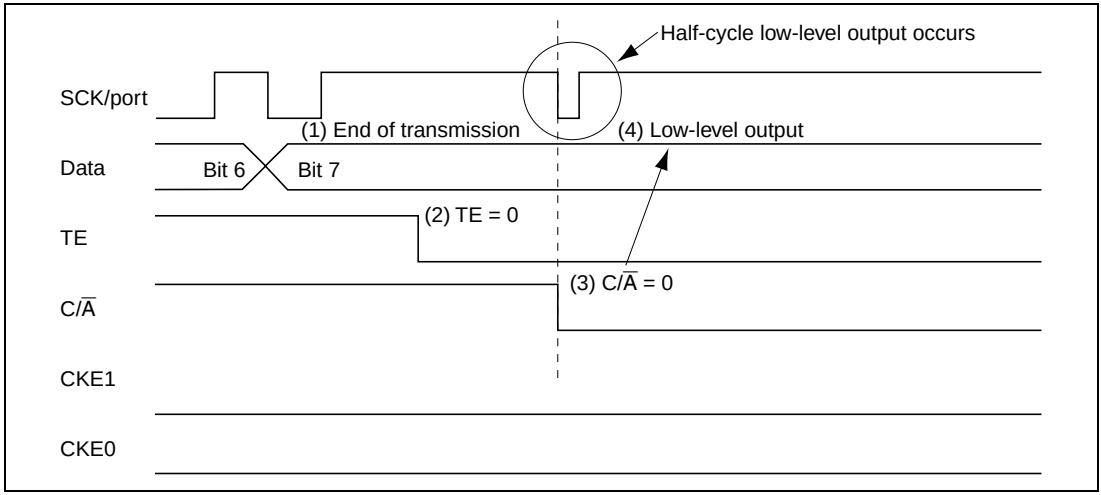


Figure 11.23 Operation when Switching from SCK Pin Function to Port Pin Function

2. Usage Note

The procedure shown below should be used to prevent low-level output when switching from the SCK pin function to the port function.

As this procedure temporarily places the SCK pin in the input state, the SCK/port pin should be pulled up beforehand with an external circuit. With DDR = 1, DR = 1, C/$\overline{A}$ = 1, CKE1 = 0, CKE0 = 0, and TE = 1, make the following settings in the order shown.

(1) End of serial data transmission

(2) TE bit = 0

(3) CKE1 bit = 1

(4) C/$\overline{A}$ bit = 0 ... switchover to port output

(5) CKE1 bit = 0

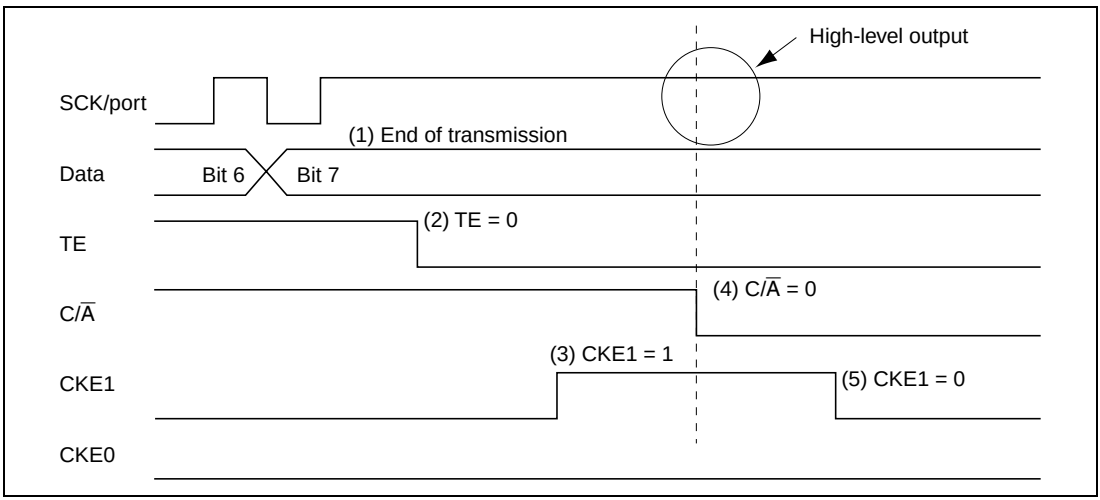


Figure 11.24 Operation when Switching from SCK Pin Function to Port Pin Function (Preventing Low-Level Output)

Section 12 Smart Card Interface

12.1 Overview

SCI0 supports an IC card (Smart Card) interface conforming to ISO/IEC 7816-3 (Identification Card) as a serial communication interface extension function.

Switching between the normal serial communication interface and the Smart Card interface is carried out by means of a register setting.

12.1.1 Features

Features of the Smart Card interface supported by the H8/3039 Group are as follows.

- Asynchronous mode
  - Data length: 8 bits
  - Parity bit generation and checking
  - Transmission of error signal (parity error) in receive mode
  - Error signal detection and automatic data retransmission in transmit mode
  - Direct convention and inverse convention both supported
- On-chip baud rate generator allows any bit rate to be selected
- Three interrupt sources
  - Three interrupt sources (transmit data empty, receive data full, and transmit/receive error) that can issue requests independently

12.1.2 Block Diagram

Figure 12.1 shows a block diagram of the Smart Card interface.

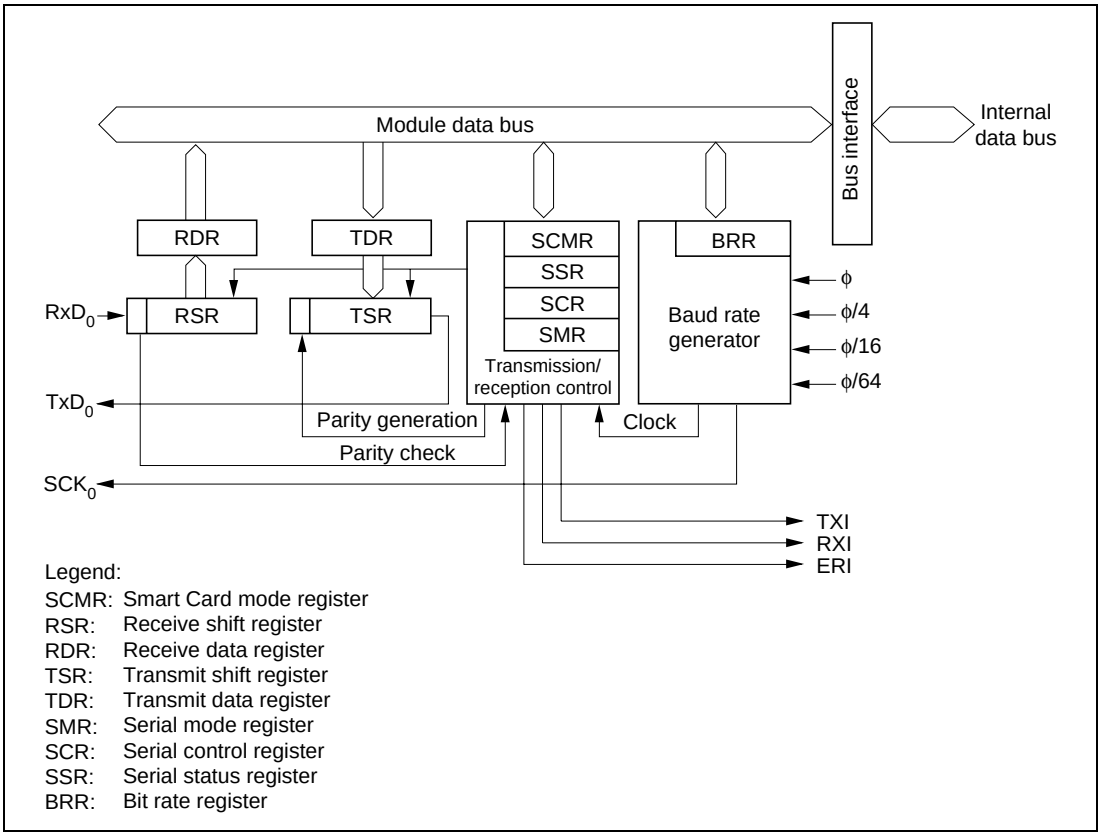


Figure 12.1 Block Diagram of Smart Card Interface

12.1.3 Pin Configuration

Table 12.1 shows the Smart Card interface pin configuration.

Table 12.1 Smart Card Interface Pins

Pin Name	Abbreviation	I/O	Function
Serial clock pin 0	SCK_0	Output	SCI_0 clock output
Receive data pin 0	RxD_0	Input	SCI_0 receive data input
Transmit data pin 0	TxD_0	Output	SCI_0 transmit data output

12.1.4 Register Configuration

Table 12.2 shows the registers used by the Smart Card interface. Details of SMR, BRR, SCR, TDR, and RDR are the same as for the normal SCI function: see the register descriptions in section 11, Serial Communication Interface.

Table 12.2 Smart Card Interface Registers

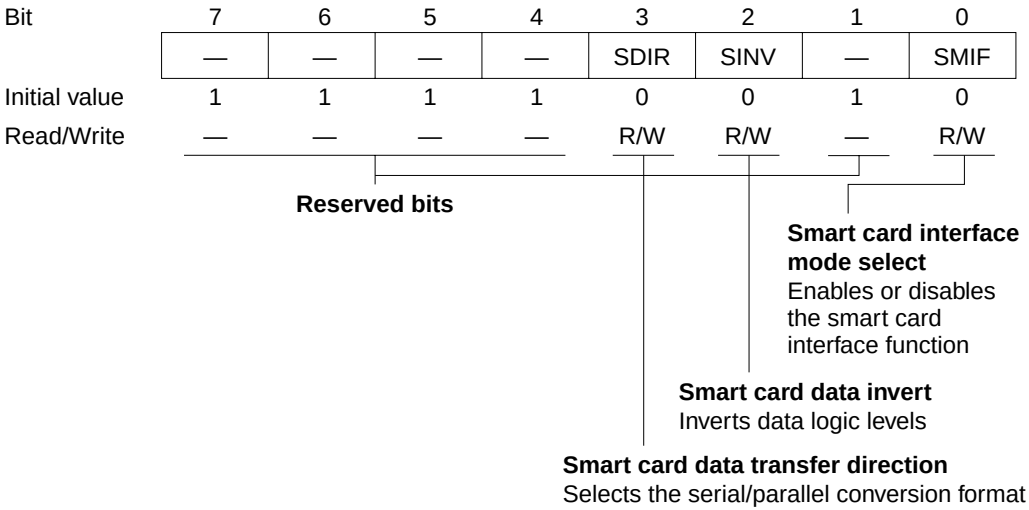
Address*[1]	Name	Abbreviation	R/W	Initial Value
H'FFB0	Serial mode register	SMR	R/W	H'00
H'FFB1	Bit rate register	BRR	R/W	H'FF
H'FFB2	Serial control register	SCR	R/W	H'00
H'FFB3	Transmit data register	TDR	R/W	H'FF
H'FFB4	Serial status register	SSR	R/(W)*[2]	H'84
H'FFB5	Receive data register	RDR	R	H'00
H'FFB6	Smart card mode register	SCMR	R/W	H'F2

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

12.2 Register Descriptions

Registers added with the Smart Card interface and bits for which the function changes are described here.

12.2.1 Smart Card Mode Register (SCMR)



SCMR is an 8-bit readable/writable register that selects the Smart Card interface function.

SCMR is initialized to H'F2 by a reset, and in standby mode.

Bits 7 to 4—Reserved: These bits cannot be modified and are always read as 1.

Bit 3—Smart Card Data Transfer Direction (SDIR): Selects the serial/parallel conversion format.

Bit 3 SDIR	Description	
0	TDR contents are transmitted LSB-first	(Initial value)
	Receive data is stored in RDR LSB-first	
1	TDR contents are transmitted MSB-first	
	Receive data is stored in RDR MSB-first	

Bit 2—Smart Card Data Invert (SINV): Specifies inversion of the data logic level. This function is used together with the SDIR bit for communication with an inverse convention card. The SINV bit does not affect the logic level of the parity bit. For parity-related setting procedures, see section 12.3.4, Register Settings.

Bit 2 SINV	Description	
0	TDR contents are transmitted as they are	(Initial value)
	Receive data is stored as it is in RDR	
1	TDR contents are inverted before being transmitted	
	Receive data is stored in inverted form in RDR	

Bit 1—Reserved: This bit cannot be modified and is always read as 1.

Bit 0—Smart Card Interface Mode Select (SMIF): This bit enables or disables the Smart Card interface function.

Bit 0 SMIF	Description	
0	Smart Card interface function is disabled	(Initial value)
1	Smart Card interface function is enabled	

12.2.2 Serial Status Register (SSR)

Bit	7	6	5	4	3	2	1	0
	TDRE	RDRF	ORER	ERS	PER	TEND	MPB	MPBT
Initial value	1	0	0	0	0	1	0	0
R/W	R/(W)*	R/(W)*	R/(W)*	R/(W)*	R/(W)*	R	R	R/W

Transmit end
Status flag indicating end of transmission

Error signal status
Status flag indicating that an error signal has been received

Note: * Only 0 can be written to bits 7 to 3, to clear these flags.

Bit 4 of SSR has a different function in Smart Card interface mode. Coupled with this, the setting conditions for bit 2, TEND, are also different.

Bits 7 to 5—Operate in the same way as for the normal SCI. For details, see section 11.2.7, Serial Status Register (SSR).

Bit 4—Error Signal Status (ERS): In Smart Card interface mode, bit 4 indicates the status of the error signal sent back from the receiving end in transmission. Framing errors are not detected in Smart Card interface mode.

Bit 4 ERS	Description
0	Indicates normal data transmission, with no error signal returned [Clearing conditions] (Initial value) • Upon reset, in standby mode, or in module stop mode • When 0 is written to ERS after reading ERS = 1
1	Indicates that the receiving device sent an error signal reporting a parity error [Setting condition] When the low level of the error signal is sampled

Note: Clearing the TE bit in SCR to 0 does not affect the ERS flag, which retains its previous state.

Bits 3 to 0—Operate in the same way as for the normal SCI. For details, see section 11.2.7, Serial Status Register (SSR).

However, the setting conditions for the TEND bit are as shown below.

Bit 2 TEND	Description
0	Transmission is in progress [Clearing condition] (Initial value) When 0 is written to TDRE after reading TDRE = 1
1	End of transmission [Setting conditions] • Upon reset and in standby mode • When the TE bit in SCR is 0 and the ERS bit is also 0 • When TDRE = 1 and ERS = 0 (normal transmission) 2.5 etu after transmission of a 1-byte serial character

Note: etu: Elementary Time Unit (time for transfer of 1 bit)

12.3 Operation

12.3.1 Overview

The main functions of the Smart Card interface are as follows.

- One frame consists of 8-bit and plus a parity bit.
- In transmission, a guard time of at least 2 etu (Elementary Time Unit: the time for transfer of one bit) is left between the end of the parity bit and the start of the next frame.
- If a parity error is detected during reception, a low error signal level is output for one etu period, 10.5 etu after the start bit.
- If the error signal is sampled during transmission, the same data is transmitted automatically after the elapse of 2 etu or longer.
- Only asynchronous communication is supported; there is no clocked synchronous communication function.

12.3.2 Pin Connections

Figure 12.2 shows a schematic diagram of Smart Card interface related pin connections.

In communication with an IC card, since both transmission and reception are carried out on a single data transmission line, the TxD_0 pin and RxD_0 pin should be connected with the LSI pin. The data transmission line should be pulled up to the V_{CC} power supply with a resistor.

When the clock generated on the Smart Card interface is used by an IC card, the SCK_0 pin output is input to the CLK pin of the IC card. No connection is needed if the IC card uses an internal clock.

LSI port output is used as the reset signal.

Other pins must normally be connected to the power supply or ground.

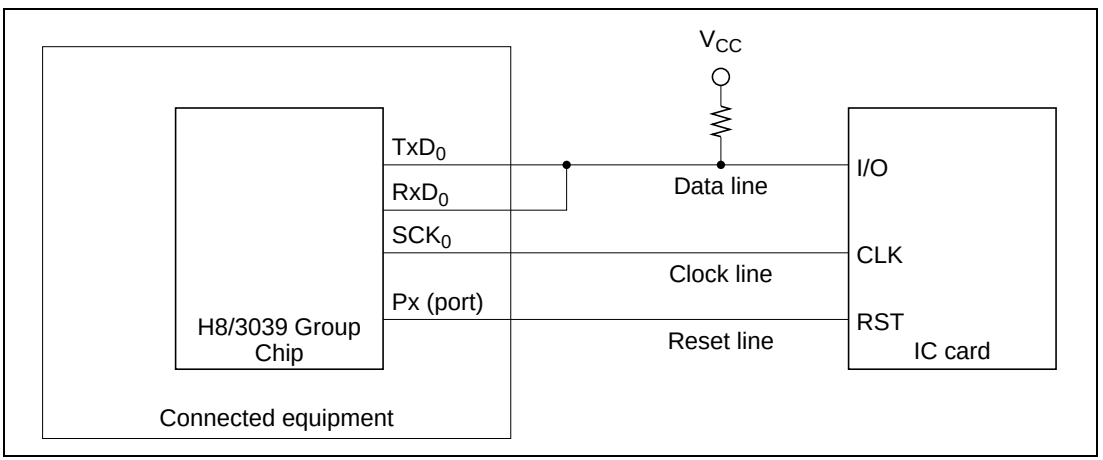


Figure 12.2 Schematic Diagram of Smart Card Interface Pin Connections

Note: If an IC card is not connected, and the TE and RE bits are both set to 1, closed transmission/reception is possible, enabling self-diagnosis to be carried out.

12.3.3 Data Format

Figure 12.3 shows the Smart Card interface data format. In reception in this mode, a parity check is carried out on each frame, and if an error is detected an error signal is sent back to the transmitting end, and retransmission of the data is requested. If an error signal is sampled during transmission, the same data is retransmitted.

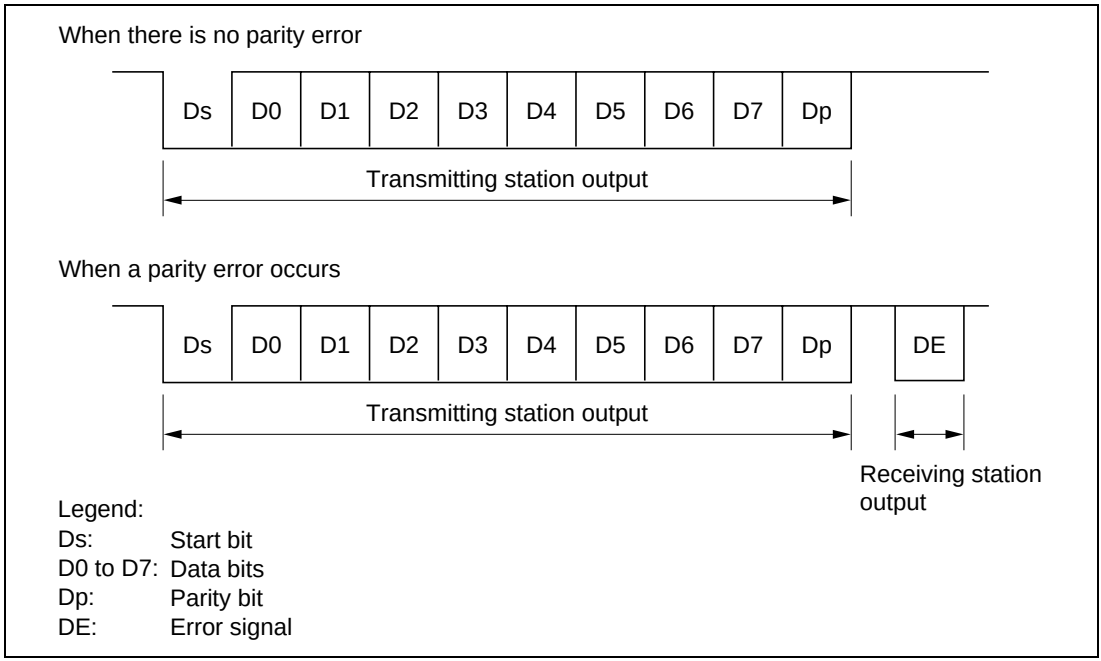


Figure 12.3 Smart Card Interface Data Format

The operation sequence is as follows.

[1] When the data line is not in use it is in the high-impedance state, and is fixed high with a pull-up resistor.

[2] The transmitting station starts a date transfer of one frame. The data frame starts with a start bit (Ds, low-level). Then 8 data bits (D0 to D7) and a parity bit (Dp) follows.

[3] With the Smart Card interface, the data line then returns to the high-impedance state. The data line is pulled high with a pull-up resistor.

[4] The receiving station carries out a parity check.

If there is no parity error and the data is received normally, the receiving station waits for reception of the next data.

If a parity error occurs, however, the receiving station outputs an error signal (DE, low-level) to request retransmission of the data. After outputting the error signal for the prescribed length of time, the receiving station places the signal line in the high-impedance state again. The signal line is pulled high again by a pull-up resistor.

[5] If the transmitting station does not receive an error signal, it proceeds to transmit the next data frame.

If it does receive an error signal, however, it returns to step [2] and retransmits the erroneous data.

12.3.4 Register Settings

Table 12.3 shows a bit map of the registers used by the smart card interface.

Bits indicated as 0 or 1 must be set to the value shown. The setting of other bits is described below.

Table 12.3 Smart Card Interface Register Settings

	Bit							
Register	**Bit 7**	**Bit 6**	**Bit 5**	**Bit 4**	**Bit 3**	**Bit 2**	**Bit 1**	**Bit 0**
SMR	0	0	1	O/$\overline{E}$	1	0	CKS1	CKS0
BRR	BRR7	BRR6	BRR5	BRR4	BRR3	BRR2	BRR1	BRR0
SCR	TIE	RIE	TE	RE	0	0	0	CKE0
TDR	TDR7	TDR6	TDR5	TDR4	TDR3	TDR2	TDR1	TDR0
SSR	TDRE	RDRF	ORER	ERS	PER	TEND	0	0
RDR	RDR7	RDR6	RDR5	RDR4	RDR3	RDR2	RDR1	RDR0
SCMR	—	—	—	—	SDIR	SINV	—	SMIF

Note: —: Unused bit.

SMR Setting: The O/$\overline{E}$ bit is cleared to 0 if the IC card is of the direct convention type, and set to 1 if of the inverse convention type.

Bits CKS1 and CKS0 select the clock source of the on-chip baud rate generator. See section 12.3.5, Clock.

BRR Setting: BRR is used to set the bit rate. See section 12.3.5, Clock, for the method of calculating the value to be set.

SCR Setting: The function of the TIE, RIE, TE, and RE bits is the same as for the normal SCI. For details, see section 11, Serial Communication Interface.

Bit CKE0 specifies the clock output. Set these bits to 0 if a clock is not to be output, or to 1 if a clock is to be output.

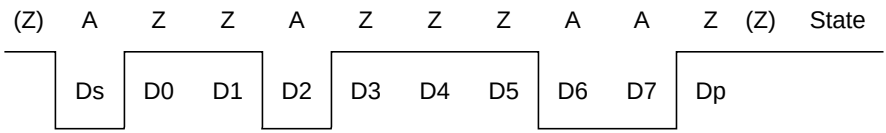
SCMR Setting: The SDIR bit is cleared to 0 if the IC card is of the direct convention type, and set to 1 if of the inverse convention type.

The SINV bit is cleared to 0 if the IC card is of the direct convention type, and set to 1 if of the inverse convention type.

The SMIF bit is set to 1 in the case of the Smart Card interface.

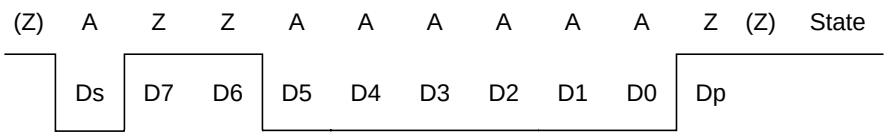
Examples of register settings and the waveform of the start character are shown below for the two types of IC card (direct convention and inverse convention).

- Direct convention (SDIR = SINV = O/$\overline{E}$ = 0)



With the direct convention type, the logic 1 level corresponds to state Z and the logic 0 level to state A, and transfer is performed in LSB-first order. The start character data above is H'3B. The parity bit is 1 since even parity is stipulated for the Smart Card.

- Inverse convention (SDIR = SINV = O/$\overline{E}$ = 1)



With the inverse convention type, the logic 1 level corresponds to state A and the logic 0 level to state Z, and transfer is performed in MSB-first order. The start character data above is H'3F. The parity bit is 0, corresponding to state Z, since even parity is stipulated for the Smart Card. With the H8/3039 Group, inversion specified by the SINV bit applies only to the data bits, D7 to D0. For parity bit inversion, the O/$\overline{E}$ bit in SMR is set to odd parity mode (the same applies to both transmission and reception).

12.3.5 Clock

Only an internal clock generated by the on-chip baud rate generator can be used as the transmit/receive clock for the smart card interface. The bit rate is set with BRR and the CKS1 and CKS0 bits in SMR. The formula for calculating the bit rate is as shown below. Table 12.5 shows some sample bit rates.

If clock output is selected by setting CKE0 to 1, a clock with a frequency of 372 times the bit rate is output from the SCK_0 pin.

$$B = \frac{\phi}{1488 \times 2^{2n-1} \times (N + 1)} \times 10^6$$

Where: N = Value set in BRR ($0 \leq N \leq 255$)
B = Bit rate (bit/s)
ϕ = Operating frequency* (MHz)
n = See table 12.4

Table 12.4 Correspondence between n and CKS1, CKS0

n	CKS1	CKS0
0	0	0
1		1
2	1	0
3		1

Note: * If the gear function is used to divide the system clock frequency, use the divided frequency to calculate the bit rate. The equation above applies directly to 1/1 frequency division.

Table 12.5 Examples of Bit Rate B (bit/s) for Various BRR Settings (When n = 0)

	ϕ (MHz)						
N	7.1424	10.00	10.7136	13.00	14.2848	16.00	18.00
0	9600.0	13440.9	14400.0	17473.1	19200.0	21505.4	24193.5
1	4800.0	6720.4	7200.0	8736.6	9600.0	10752.7	12096.8
2	3200.0	4480.3	4800.0	5824.4	6400.0	7168.5	8064.5

Note: Bit rates are rounded off to one decimal place.

The method of calculating the value from the operating frequency and bit rate, on the other hand, is shown below. N is an integer, $0 \leq N \leq 255$, and the smaller error is specified.

$$N = \frac{\phi}{1488 \times 2^{2n-1} \times B} \times 10^6 - 1$$

Table 12.6 Examples of BRR Settings for Bit Rate B (bit/s) (When n = 0)

	ϕ (MHz)													
	7.1424		10.00		10.7136		13.00		14.2848		16.00		18.00	
bit/s	N	Error	N	Error	N	Error	N	Error	N	Error	N	Error	N	Error
9600	0	0.00	1	30	1	25	1	8.99	1	0.00	1	12.01	2	15.99

Table 12.7 Maximum Bit Rate at Various Frequencies (Smart Card Interface Mode)

ϕ (MHz)	Maximum Bit Rate (bit/s)	N	n
7.1424	9600	0	0
10.00	13441	0	0
10.7136	14400	0	0
13.00	17473	0	0
14.2848	19200	0	0
16.00	21505	0	0
18.00	24194	0	0

The bit rate error is given by the following formula:

$$\text{Error (\%)} = \left(\frac{\phi}{1488 \times 2^{2n-1} \times B \times (N + 1)} \times 10^6 - 1\right) \times 100$$

12.3.6 Data Transfer Operations

Initialization

Before transmitting and receiving data, initialize the SCI as described below. Initialization is also necessary when switching from transmit mode to receive mode, or vice versa.

[1] Clear the TE and RE bits in SCR to 0.

[2] Clear the error flags ERS, PER, and ORER in SSR to 0.

[3] Set the O/$\overline{E}$ bit and CKS1 and CKS0 bits in SMR. Clear the C/$\overline{A}$, CHR, and MP bits to 0, and set the STOP and PE bits to 1.

[4] Set the SMIF, SDIR, and SINV bits in SCMR.
When the SMIF bit is set to 1, the TxD_0 and RxD_0 pins are both switched from ports to SCI pins, and are placed in the high-impedance state.

[5] Set the value corresponding to the bit rate in BRR.

[6] Set the CKE0 bit in SCR. Clear the TIE, RIE, TE, RE, MPIE, TEIE and CKE1 bits to 0.
If the CKE0 bit is set to 1, the clock is output from the SCK_0 pin.

[7] Wait at least one bit interval, then set the TIE, RIE, TE, and RE bits in SCR. Do not set the TE bit and RE bit at the same time, except for self-diagnosis.

Serial Data Transmission

As data transmission in smart card mode involves error signal sampling and retransmission processing, the processing procedure is different from that for the normal SCI. Figure 12.4 shows an example of the transmission processing flow, and figure 12.5 shows the relation between a transmit operation and the internal registers.

[1] Perform Smart Card interface mode initialization as described above in Initialization.

[2] Check that the ERS error flag in SSR is cleared to 0.

[3] Repeat steps [2] and [3] until it can be confirmed that the TEND flag in SSR is set to 1.

[4] Write the transmit data to TDR, clear the TDRE flag to 0, and perform the transmit operation. The TEND flag is cleared to 0.

[5] When transmitting data continuously, go back to step [2].

[6] To end transmission, clear the TE bit to 0.

With the above processing, interrupt servicing is possible.

If transmission ends and the TEND flag is set to 1 while the TIE bit is set to 1 and interrupt requests are enabled, a transmit data empty interrupt (TXI) request will be generated. If an error occurs in transmission and the ERS flag is set to 1 while the RIE bit is set to 1 and interrupt requests are enabled, a transfer error interrupt (ERI) request will be generated.

For details, see the following Interrupt Operations.

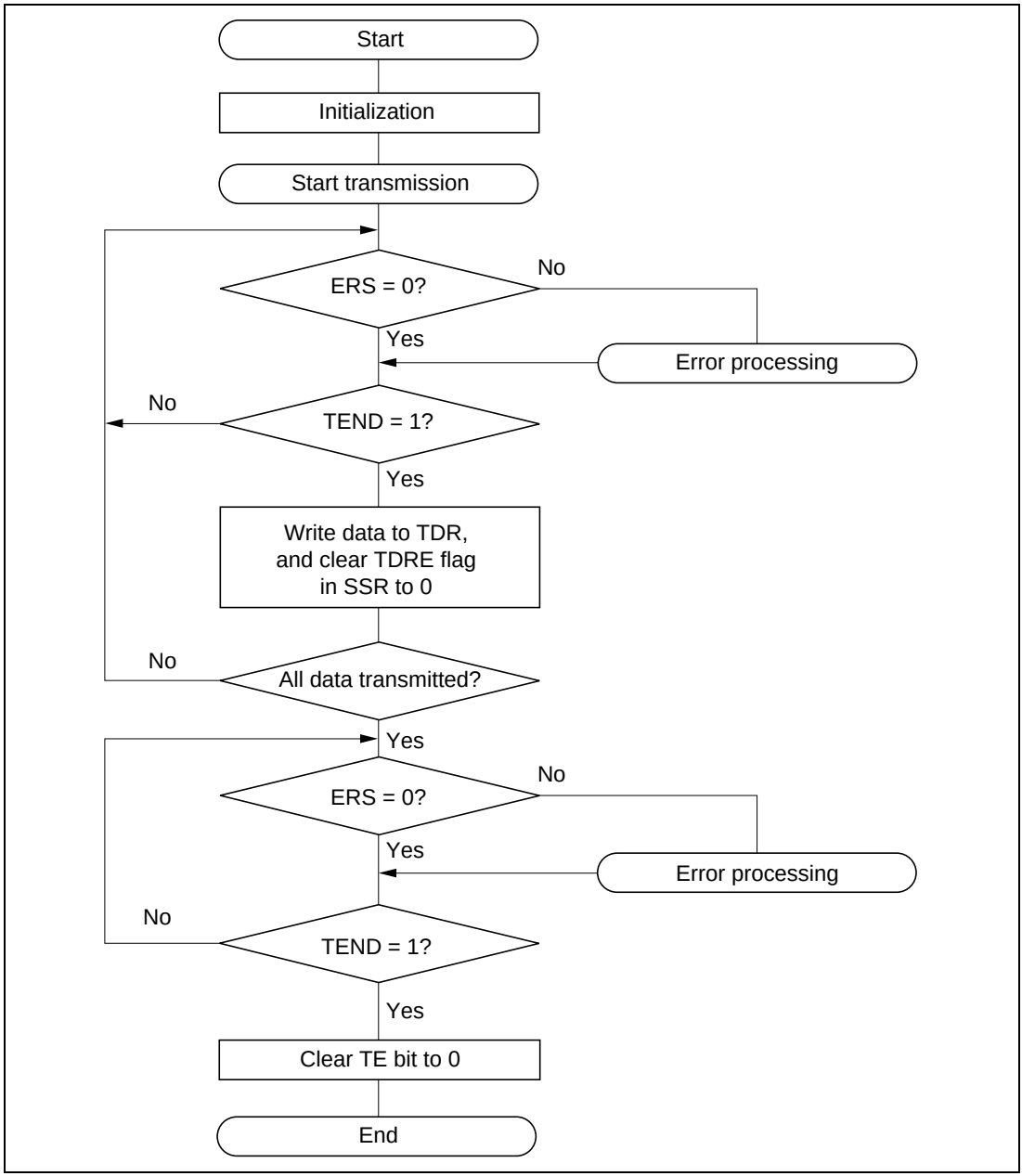


Figure 12.4 Example of Transmission Processing Flow

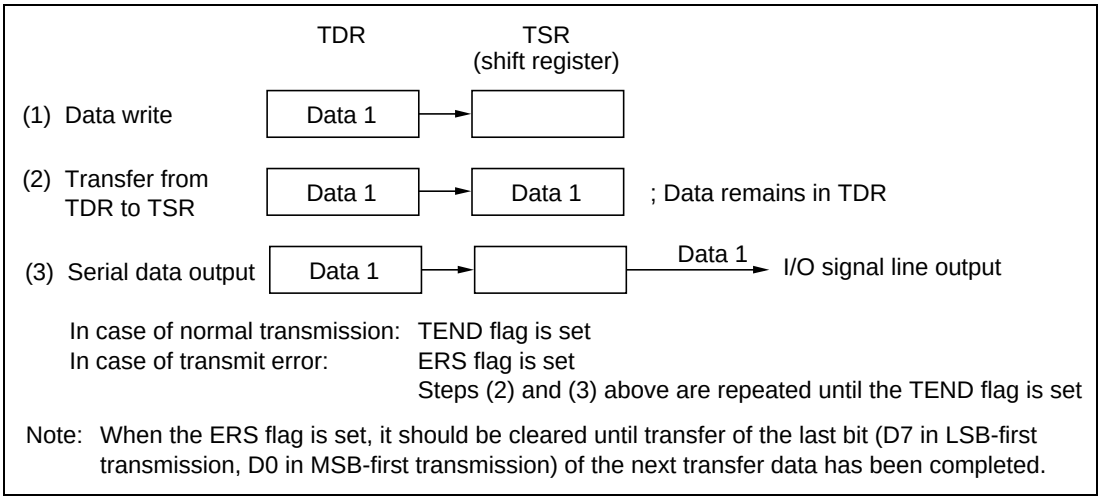


Figure 12.5 Relation Between Transmit Operation and Internal Registers

Serial Data Reception

Data reception in Smart Card mode uses the same processing procedure as for the normal SCI. Figure 12.6 shows an example of the transmission processing flow.

[1] Perform Smart Card interface mode initialization as described above in Initialization.

[2] Check that the ORER flag and PER flag in SSR are cleared to 0. If either flag is set, perform the appropriate receive error processing, then clear both the ORER and the PER flag to 0.

[3] Repeat steps [2] and [3] until it can be confirmed that the RDRF flag is set to 1.

[4] Read the receive data from RDR.

[5] When receiving data continuously, clear the RDRF flag to 0 and go back to step [2].

[6] To end reception, clear the RE bit to 0.

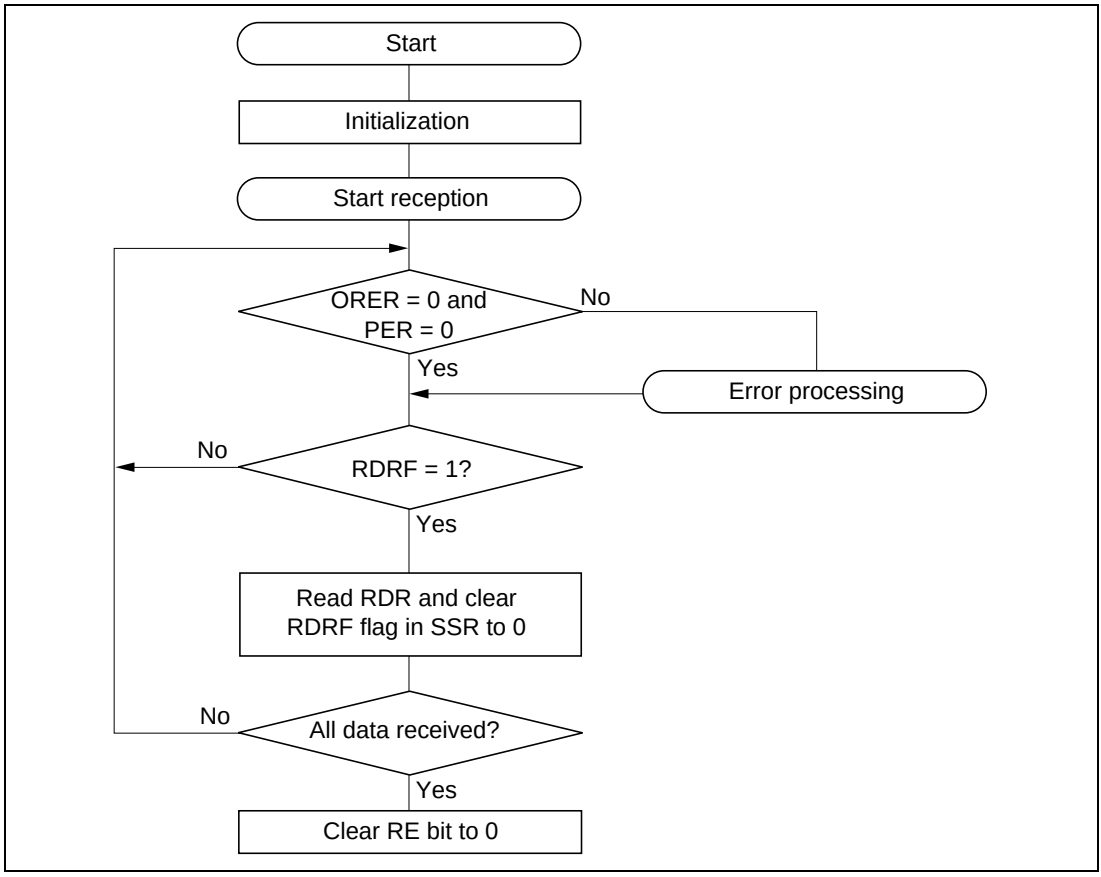


Figure 12.6 Example of Reception Processing Flow

With the above processing, interrupt servicing is possible.

If reception ends and the RDRF flag is set to 1 while the RIE bit is set to 1 and interrupt requests are enabled, a receive data full interrupt (RXI) request will be generated. If an error occurs in reception and either the ORER flag or the PER flag is set to 1, a transfer error interrupt (ERI) request will be generated.

For details, see Interrupt Operation below.

If a parity error occurs during reception and the PER is set to 1, the received data is still transferred to RDR, and therefore this data can be read.

Mode Switching Operation

When switching from receive mode to transmit mode, first confirm that the receive operation has been completed, then start from initialization, clearing RE bit to 0 and setting TE bit to 1. The RDRF flag or the PER and ORER flags can be used to check that the receive operation has been completed.

When switching from transmit mode to receive mode, first confirm that the transmit operation has been completed, then start from initialization, clearing TE bit to 0 and setting RE bit to 1. The TEND flag can be used to check that the transmit operation has been completed.

Interrupt Operation

There are three interrupt sources in smart card interface mode: transmit data empty interrupt (TXI) requests, transfer error interrupt (ERI) requests, and receive data full interrupt (RXI) requests. The transmit end interrupt (TEI) request is not used in this mode.

When the TEND flag in SSR is set to 1, a TXI interrupt request is generated.

When the RDRF flag in SSR is set to 1, an RXI interrupt request is generated.

When any of flags ORER, PER, and ERS in SSR is set to 1, an ERI interrupt request is generated. The relationship between the operating states and interrupt sources is shown in table 12.8.

Table 12.8 Smart Card Mode Operating States and Interrupt Sources

Operating State		Flag	Mask Bit	Interrupt Source
Transmit Mode	Normal operation	TEND	TIE	TXI
	Error	ERS	RIE	ERI
Receive Mode	Normal operation	RDRF	RIE	RXI
	Error	PER, ORER	RIE	ERI

12.4 Usage Note

The following points should be noted when using the SCI as a smart card interface.

Receive Data Sampling Timing and Reception Margin in Smart Card Interface Mode

In smart card interface mode, the SCI operates on a basic clock with a frequency of 372 times the transfer rate.

In reception, the SCI samples the falling edge of the start bit using the basic clock, and performs internal synchronization. Receive data is latched internally at the rising edge of the 186th pulse of the basic clock. This is illustrated in figure 12.7.

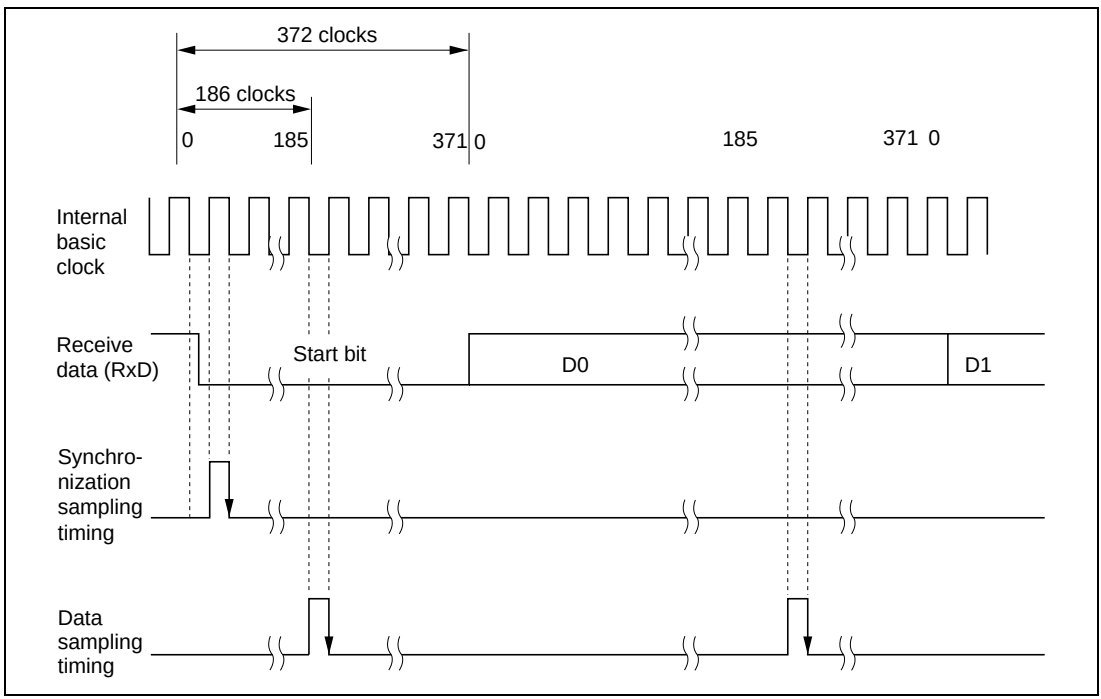


Figure 12.7 Receive Data Sampling Timing in Smart Card Mode

Thus the reception margin in smart card interface mode is given by the following formula.

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

Where M: Reception margin (%)
N: Ratio of bit rate to clock (N = 372)
D: Clock duty (D = 0 to 1.0)
L: Frame length (L = 10)
F: Absolute value of clock frequency deviation

Assuming values of F = 0 and D = 0.5 in the above formula, the reception margin formula is as follows.

When D = 0.5 and F = 0,

$$M = (0.5 - 1/2 \times 372) \times 100\%$$
$$= 49.866\%$$

Retransfer Operations

Retransfer operations are performed by the SCI in receive mode and transmit mode as described below.

- Retransfer operation when SCI is in receive mode

  Figure 12.8 illustrates the retransfer operation when the SCI is in receive mode.

[1] If an error is found when the received parity bit is checked, the PER bit in SSR is automatically set to 1. If the RIE bit in SCR is enabled at this time, an ERI interrupt request is generated. The PER bit in SSR should be kept cleared to 0 until the next parity bit is sampled.

[2] The RDRF bit in SSR is not set for a frame in which an error has occurred.

[3] If no error is found when the received parity bit is checked, the PER bit in SSR is not set to 1.

[4] If no error is found when the received parity bit is checked, the receive operation is judged to have been completed normally, and the RDRF flag in SSR is automatically set to 1. If the RIE bit in SCR is enabled at this time, an RXI interrupt request is generated.

[5] When a normal frame is received, the pin retains the high-impedance state at the timing for error signal transmission.

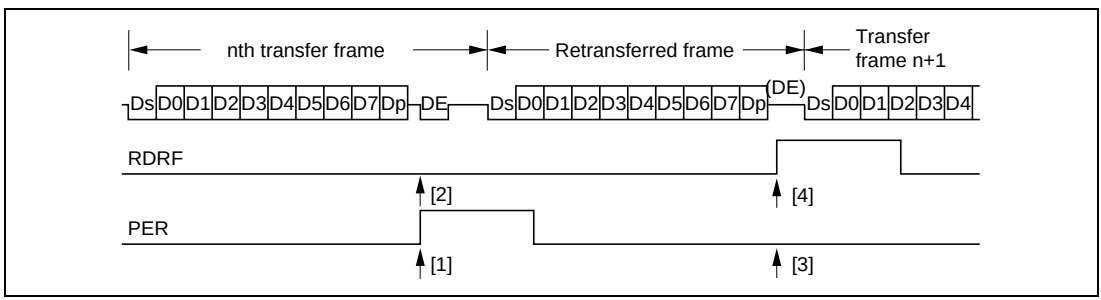


Figure 12.8 Retransfer Operation in SCI Receive Mode

- Retransfer operation when SCI is in transmit mode

  Figure 12.9 illustrates the retransfer operation when the SCI is in transmit mode.

[6] If an error signal is sent back from the receiving end after transmission of one frame is completed, the ERS bit in SSR is set to 1. If the RIE bit in SCR is enabled at this time, an ERI interrupt request is generated. The ERS bit in SSR should be kept cleared to 0 until the next parity bit is sampled.

[7] The TEND bit in SSR is not set for a frame for which an error signal indicating an abnormality is received.

[8] If an error signal is not sent back from the receiving end, the ERS bit in SSR is not set.

[9] If an error signal is not sent back from the receiving end, transmission of one frame, including a retransfer, is judged to have been completed, and the TEND bit in SSR is set to 1. If the TIE bit in SCR is enabled at this time, a TXI interrupt request is generated.

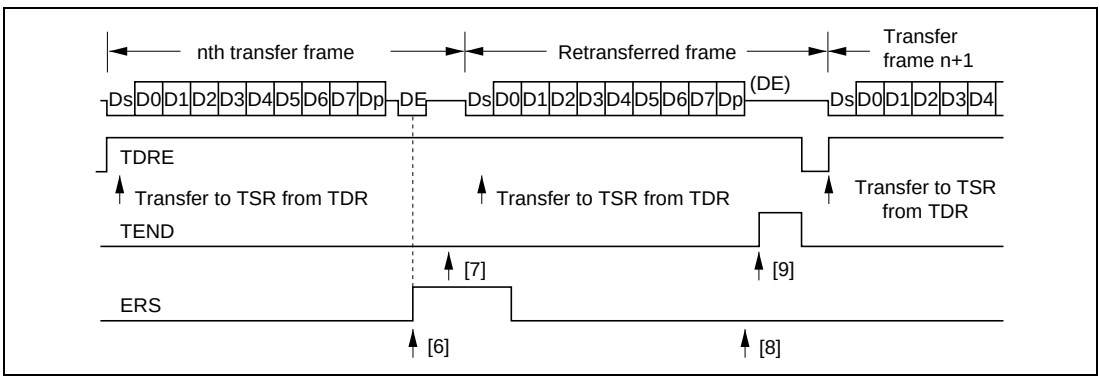


Figure 12.9 Retransfer Operation in SCI Transmit Mode

Section 13 A/D Converter

13.1 Overview

The H8/3039 Group includes a 10-bit successive-approximations A/D converter with a selection of up to eight analog input channels.

When the A/D converter is not used, it can be halted independently to conserve power. For details see section 17.6, Module Standby Function.

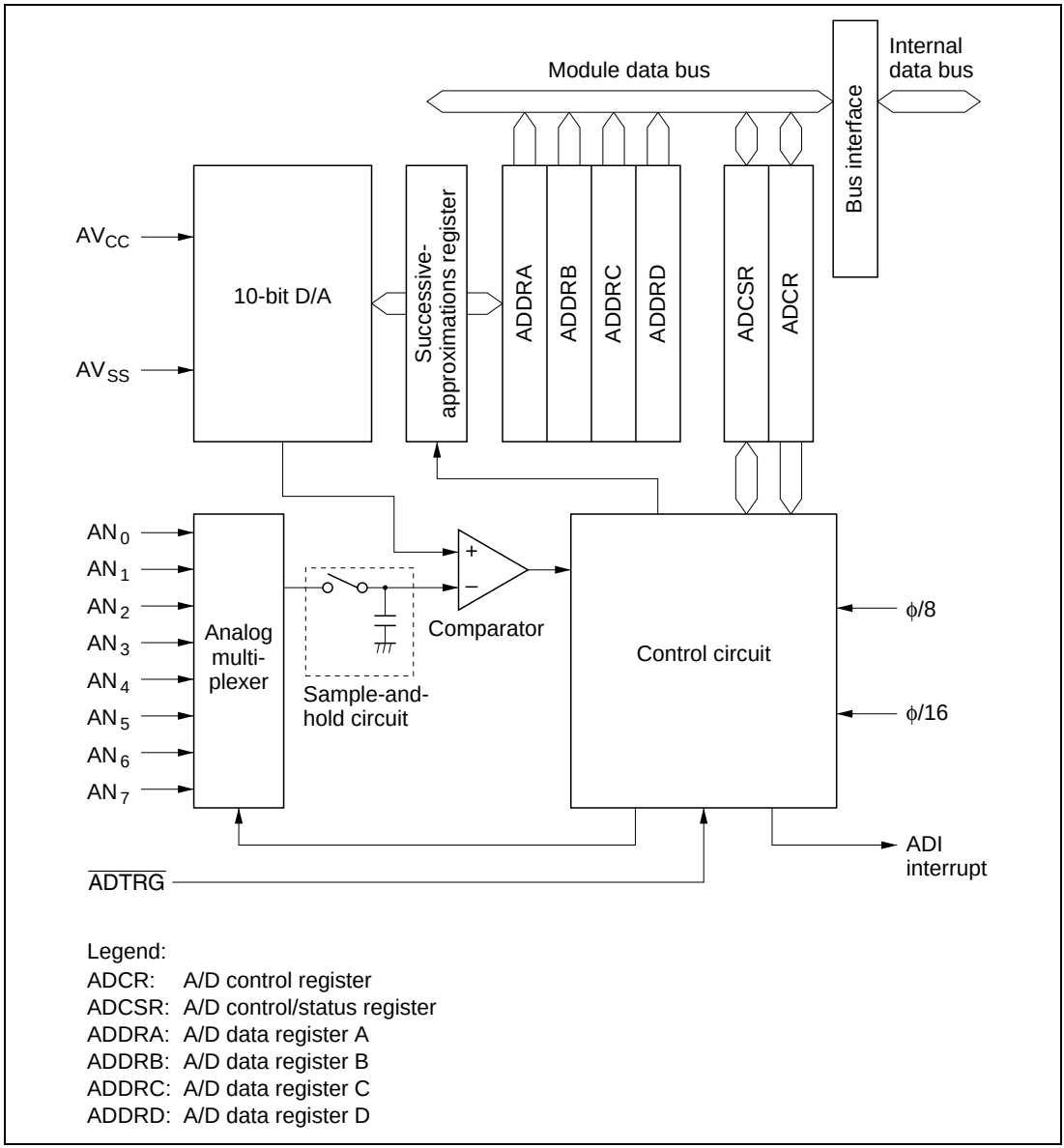
13.1.1 Features

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 reference voltage at the AV_{CC} pin.
- High-speed conversion
  Conversion time: minimum 7.4 µs per channel (with 18 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 to the 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.

13.1.2 Block Diagram

Figure 13.1 shows a block diagram of the A/D converter.



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 13.1 A/D Converter Block Diagram

13.1.3 Input Pins

Table 13.1 summarizes the A/D converter's input pins. The eight analog input pins are divided into two groups: group 0 (AN_0 to AN_3), and group 1 (AN_4 to AN_7). AV_{CC} and AV_{SS} are the power supply for the analog circuits in the A/D converter.

Table 13.1 A/D Converter Pins

Pin Name	Abbreviation	I/O	Function
Analog power supply pin	AV_{CC}	Input	Analog power supply and reference voltage
Analog ground pin	AV_{SS}	Input	Analog ground and 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	$\overline{\text{ADTRG}}$	Input	External trigger input for starting A/D conversion

13.1.4 Register Configuration

Table 13.2 summarizes the A/D converter's registers.

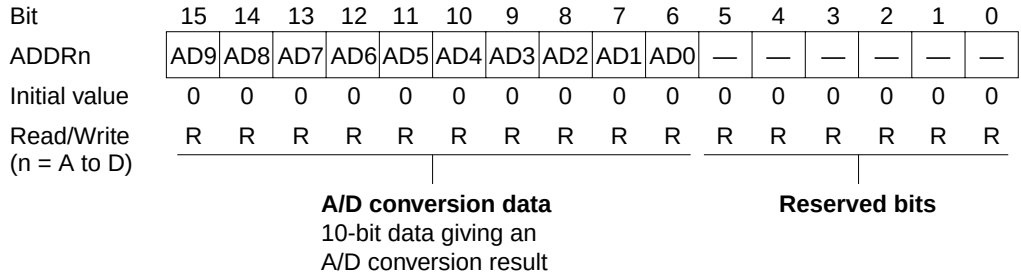
Table 13.2 A/D Converter Registers

Address*[1]	Name	Abbreviation	R/W	Initial Value
H'FFE0	A/D data register A (high)	ADDRAH	R	H'00
H'FFE1	A/D data register A (low)	ADDRAL	R	H'00
H'FFE2	A/D data register B (high)	ADDRBH	R	H'00
H'FFE3	A/D data register B (low)	ADDRBL	R	H'00
H'FFE4	A/D data register C (high)	ADDRCH	R	H'00
H'FFE5	A/D data register C (low)	ADDRCL	R	H'00
H'FFE6	A/D data register D (high)	ADDRDH	R	H'00
H'FFE7	A/D data register D (low)	ADDRDL	R	H'00
H'FFE8	A/D control/status register	ADCSR	R/(W)*[2]	H'00
H'FFE9	A/D control register	ADCR	R/W	H'7F

Notes: 1. Lower 16 bits of the address
2. Only 0 can be written in bit 7 to clear the flag.

13.2 Register Descriptions

13.2.1 A/D Data Registers A to D (ADDRA to ADDRD)



The four A/D data registers (ADDRA to ADDRD) are 16-bit read-only registers that store the results of A/D conversion.

An A/D conversion produces 10-bit data, which is transferred for storage into the A/D data register corresponding to the selected channel. The upper 8 bits of the result are stored in the upper byte of the A/D data register. The lower 2 bits are stored in the lower byte. Bits 5 to 0 of an A/D data register are reserved bits that always read 0. Table 13.3 indicates the pairings of analog input channels and A/D data registers.

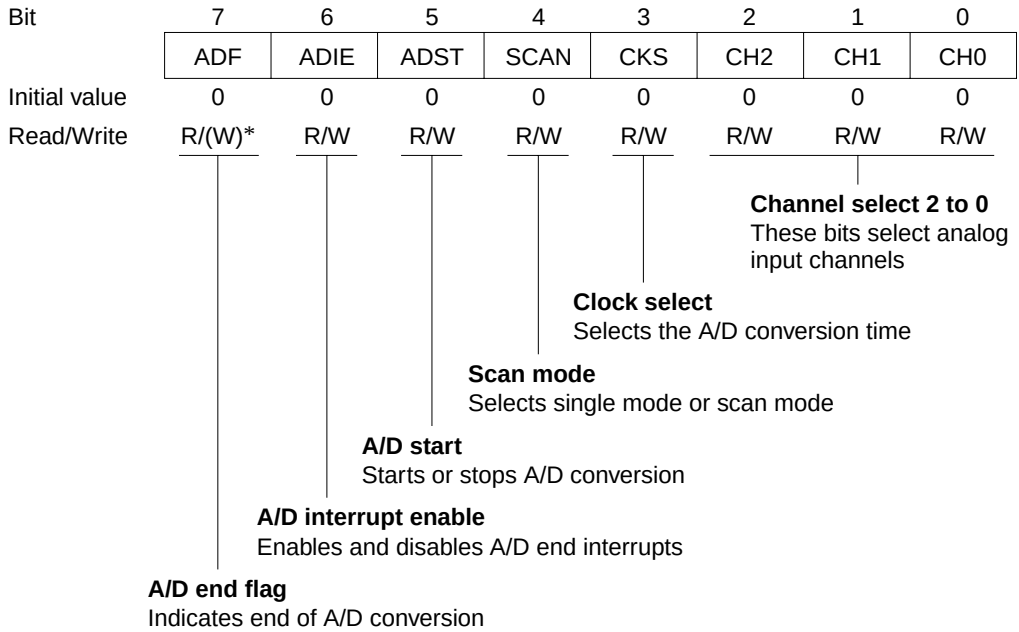
The CPU can always read the A/D data registers. The upper byte can be read directly, but the lower byte is read through a temporary register (TEMP). For details see section 13.3, CPU Interface.

The A/D data registers are initialized to H'0000 by a reset and in standby mode.

Table 13.3 Analog Input Channels and A/D Data Registers

Analog Input Channel		
Group 0	**Group 1**	**A/D Data Register**
AN0	AN4	ADDRA
AN1	AN5	ADDRB
AN2	AN6	ADDRC
AN3	AN7	ADDRD

13.2.2 A/D Control/Status Register (ADCSR)



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 converter. 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]	(Initial value)
	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	

Bit 6—A/D Interrupt Enable (ADIE): Enables or disables the interrupt (ADI) requested at the end of A/D conversion.

Bit 6 ADIE	Description	
0	A/D end interrupt request (ADI) is disabled	(Initial value)
1	A/D end interrupt request (ADI) is enabled	

Bit 5—A/D Start (ADST): Starts or stops A/D conversion. The ADST bit remains set to 1 during 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	(Initial value)
1	Single mode: A/D conversion starts; ADST is automatically cleared to 0 when conversion ends.	
	Scan mode: A/D conversion starts and continues, cycling among the selected channels, until ADST is cleared to 0 by software, by a reset, or by a transition to standby mode.	

Bit 4—Scan Mode (SCAN): Selects single mode or scan mode. For further information on operation in these modes, see section 13.4, Operation. Clear the ADST bit to 0 before switching the conversion mode.

Bit 4 SCAN	Description	
0	Single mode	(Initial value)
1	Scan mode	

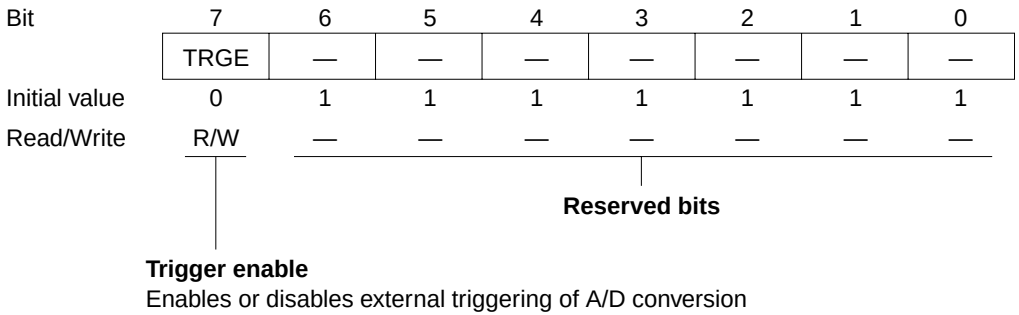
Bit 3—Clock Select (CKS): Selects the A/D conversion time. Clear the ADST bit to 0 before switching the conversion time.

Bit 3 CKS	Description	
0	Conversion time = 266 states (maximum)	(Initial value)
1	Conversion time = 134 states (maximum)	

Bits 2 to 0—Channel Select 2 to 0 (CH2 to CH0): These bits and the SCAN bit select the analog input channels. Clear the ADST bit to 0 before changing the channel selection.

Group Selection	Channel Selection		Description	
CH2	CH1	CH0	Single Mode	Scan Mode
0	0	0	AN_0 (Initial value)	AN_0
		1	AN_1	AN_0, AN_1
	1	0	AN_2	AN_0 to AN_2
		1	AN_3	AN_0 to AN_3
1	0	0	AN_4	AN_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

13.2.3 A/D Control Register (ADCR)



ADCR is an 8-bit readable/writable register that enables or disables external triggering of A/D conversion. ADCR is initialized to H'7F by a reset and in standby mode.

Bit 7—Trigger Enable (TRGE): Enables or disables external triggering of A/D conversion.

Bit 7 TRGE	Description	
0	A/D conversion cannot be externally triggered	(Initial value)
1	A/D conversion starts at the falling edge of the external trigger signal ($\overline{\text{ADTRG}}$)	

Bits 6 to 0—Reserved: These bits cannot be modified and are always read as 1.

13.3 CPU Interface

ADDRA to ADDRD are 16-bit registers, but they are connected to the CPU by an 8-bit data bus. Therefore, although the upper byte can be accessed directly by the CPU, the lower byte is read through an 8-bit temporary register (TEMP).

An A/D data register is read as follows. When the upper byte is read, the upper-byte value is transferred directly to the CPU and the lower-byte value is transferred into TEMP. Next, when the lower byte is read, the TEMP contents are transferred to the CPU.

When reading an A/D data register, always read the upper byte before the lower byte. It is possible to read only the upper byte, but if only the lower byte is read, incorrect data may be obtained.

Figure 13.2 shows the data flow for access to an A/D data register.

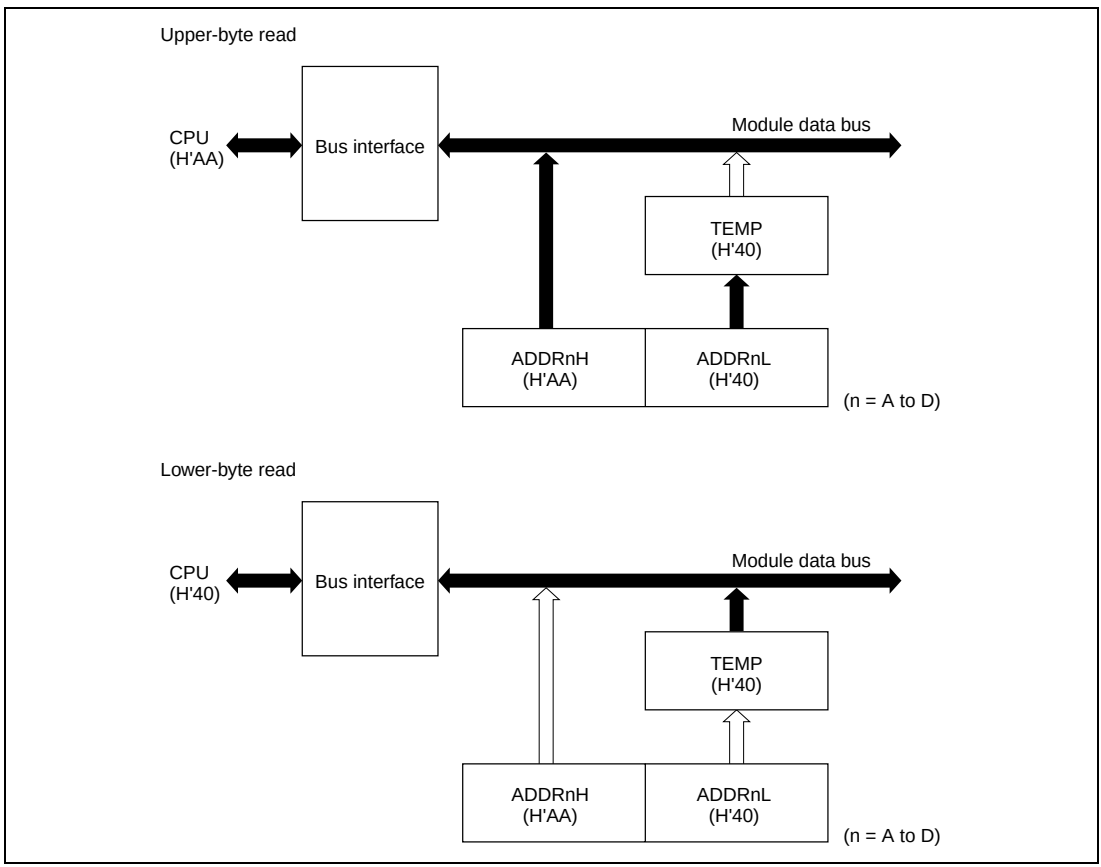


Figure 13.2 A/D Data Register Access Operation (Reading H'AA40)

13.4 Operation

The A/D converter operates by successive approximations with 10-bit resolution. It has two operating modes: single mode and scan mode.

13.4.1 Single Mode (SCAN = 0)

Single mode should be selected when only one A/D conversion on one channel is required. A/D conversion starts when the ADST bit is set to 1 by software, or by external trigger input. The ADST bit remains set to 1 during A/D conversion and is automatically cleared to 0 when conversion ends.

When conversion ends the ADF bit is set to 1. If the ADIE bit is also set to 1, an ADI interrupt is requested at this time. To clear the ADF flag to 0, first read ADCSR, then write 0 in ADF.

When the mode or analog input channel must be switched during analog conversion, to prevent incorrect operation, first clear the ADST bit to 0 in ADCSR to halt A/D conversion. After making the necessary changes, set the ADST bit to 1 to start A/D conversion again. The ADST bit can be 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 13.3 shows a timing diagram for this example.

1. Single mode is selected (SCAN = 0), input channel AN_1 is selected (CH2 = CH1 = 0, CH0 = 1), the A/D interrupt is enabled (ADIE = 1), and A/D conversion is started (ADST = 1).
2. When A/D conversion is completed, the result is transferred into ADDRB. At the same time the ADF flag is set to 1, the ADST bit is cleared to 0, and the A/D converter becomes idle.
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 set to 1, A/D conversion starts again and steps 2 to 7 are repeated.

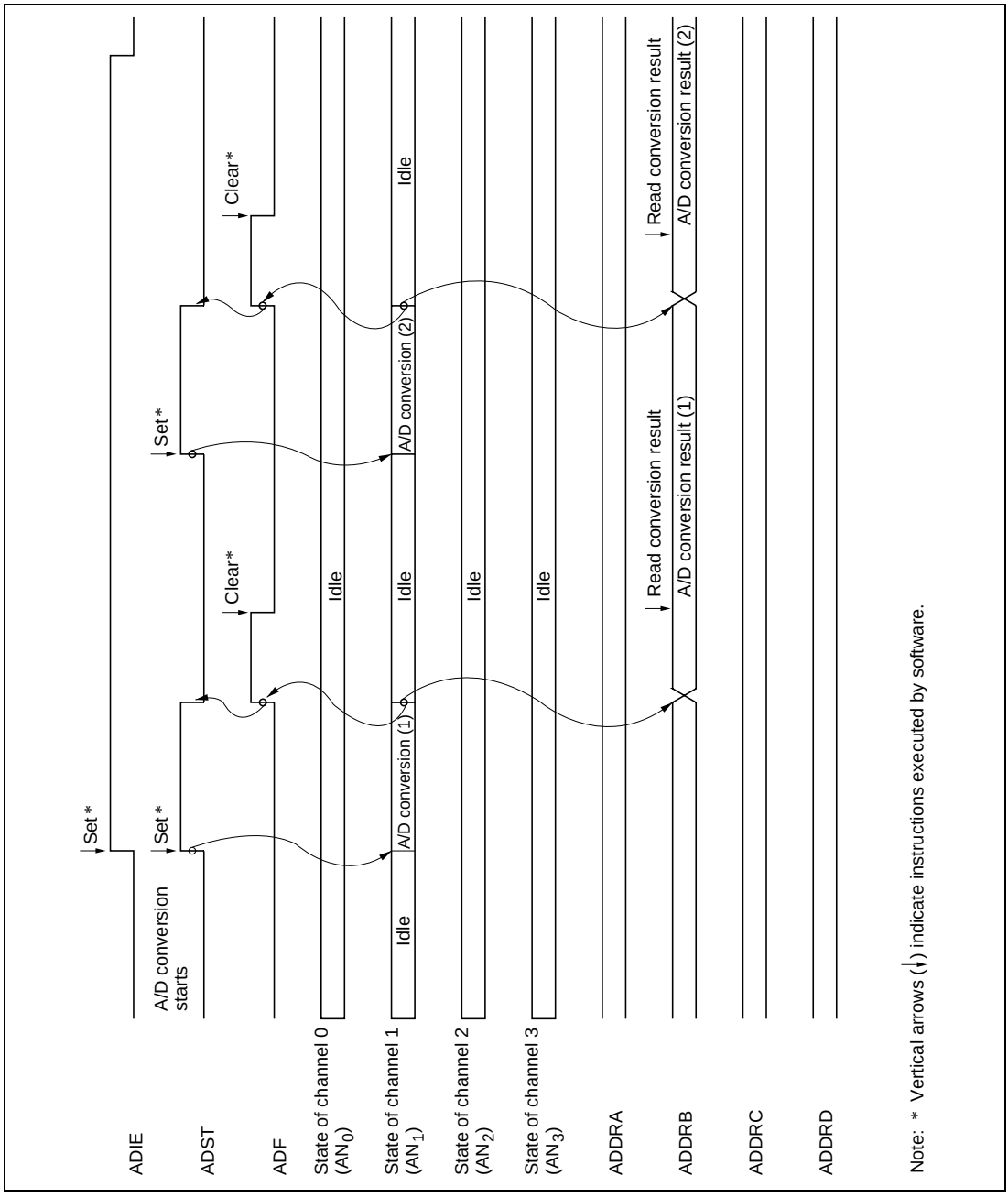


Figure 13.3 Example of A/D Converter Operation (Single Mode, Channel 1 Selected)

13.4.2 Scan Mode (SCAN = 1)

Scan mode is useful for monitoring analog inputs in a group of one or more channels. When the ADST bit is set to 1 by software or external trigger input, A/D conversion starts on the first channel in the group (AN_0 when CH2 = 0, AN_4 when CH2 = 1). When two or more channels are selected, after conversion of the first channel ends, conversion of the second channel (AN_1 or AN_5) starts immediately. A/D conversion continues cyclically on the selected channels until the ADST bit is cleared to 0. The conversion results are transferred for storage into the A/D data registers corresponding to the channels.

When the mode or analog input channel selection must be changed during analog conversion, to prevent incorrect operation, first clear the ADST bit to 0 in ADCSR to halt A/D conversion. After making the necessary changes, set the ADST bit to 1. A/D conversion will start again from the first channel in the group. The ADST bit can be set at the same time as the mode or channel selection is changed.

Typical operations when three channels in group 0 (AN_0 to AN_2) are selected in scan mode are described next. Figure 13.4 shows a timing diagram for this example.

1. Scan mode is selected (SCAN = 1), scan group 0 is selected (CH2 = 0), analog input channels AN_0 to AN_2 are selected (CH1 = 1, CH0 = 0), and A/D conversion is started (ADST = 1).
2. When A/D conversion of the first channel (AN_0) is completed, the result is transferred into 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 is set to 1 and conversion of the first channel (AN_0) starts again. If the ADIE bit is set to 1, an ADI 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 bit is cleared to 0, A/D conversion stops. After that, if the ADST bit is set to 1, A/D conversion starts again from the first channel (AN_0).

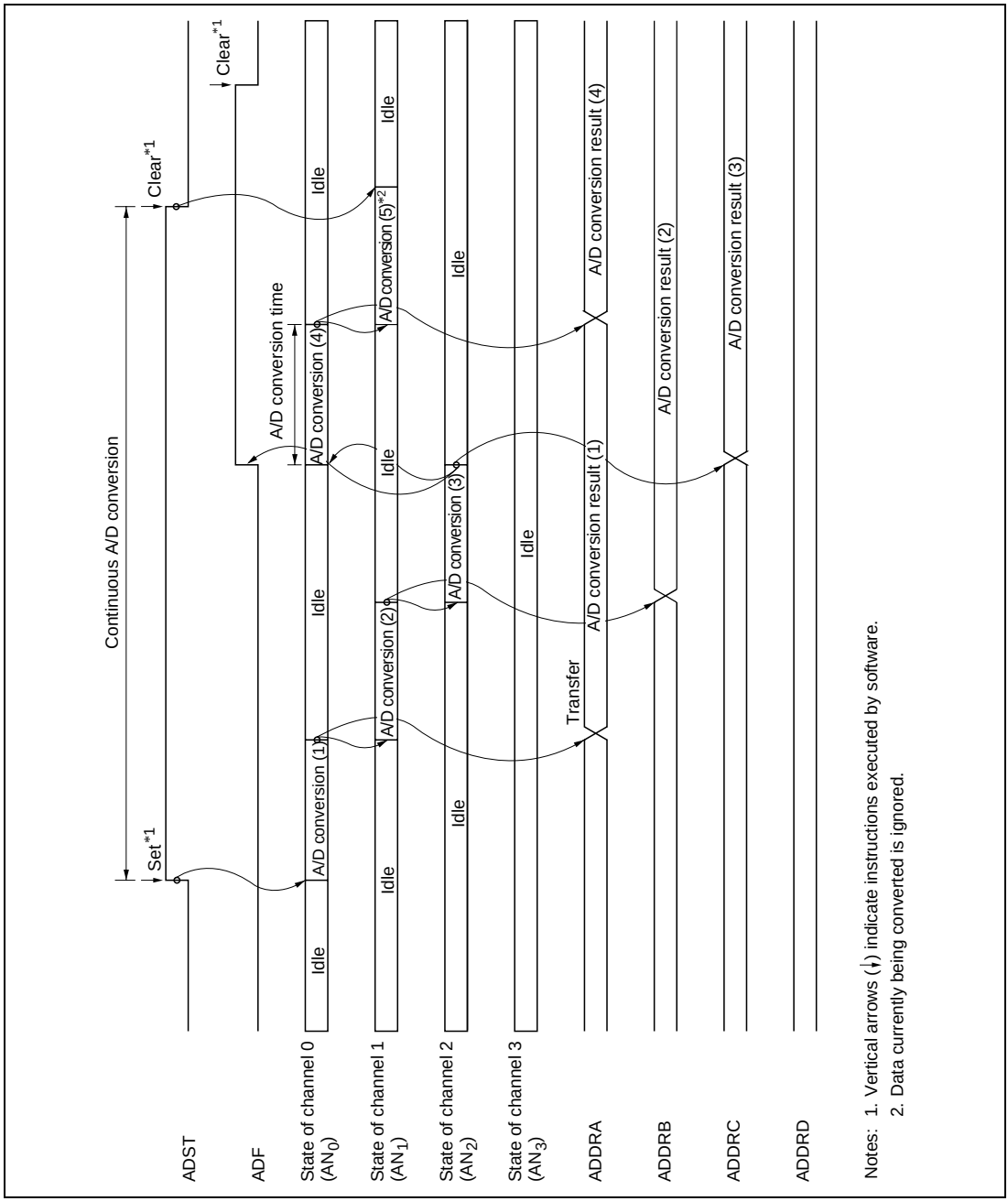


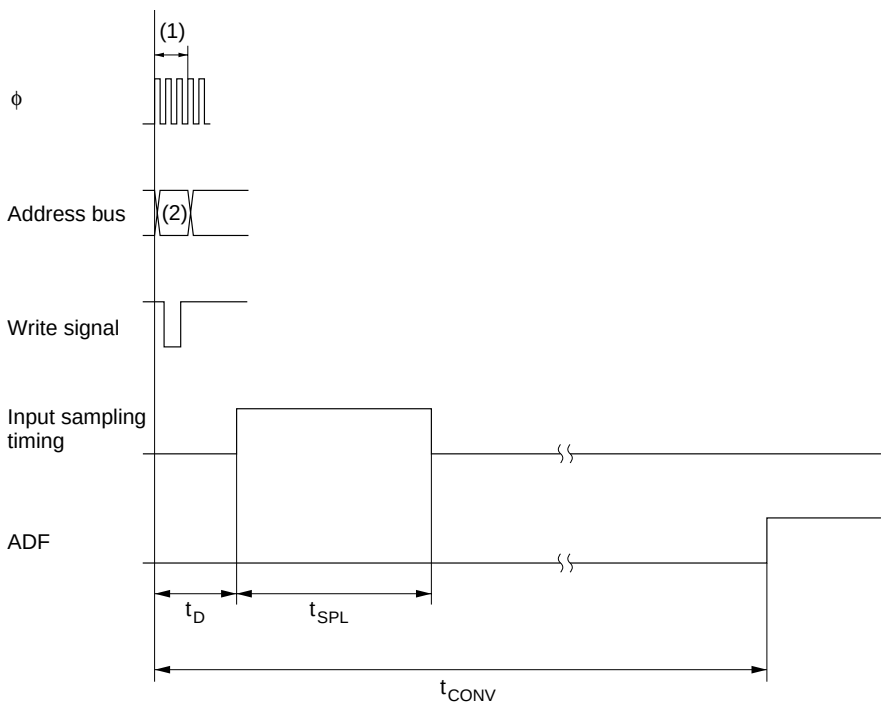
Figure 13.4 Example of A/D Converter Operation (Scan Mode, Channels AN_0 to AN_2 Selected)

13.4.3 Input Sampling and A/D Conversion Time

The A/D converter has a built-in sample-and-hold circuit. The A/D converter samples the analog input at a time t_D after the ADST bit is set to 1, then starts conversion. Figure 13.5 shows the A/D conversion timing. Table 13.4 indicates the A/D conversion time.

As indicated in figure 13.5, the A/D conversion time includes t_D and the input sampling time. The length of t_D varies depending on the timing of the write access to ADCSR. The total conversion time therefore varies within the ranges indicated in table 13.4.

In scan mode, the values given in table 13.4 apply to the first conversion. In the second and subsequent conversions the conversion time is fixed at 256 states when CKS = 0 or 128 states when CKS = 1.



Legend:
(1): ADCSR write cycle
(2): ADCSR address
t_D: Synchronization delay
t_{SPL}: Input sampling time
t_{CONV}: A/D conversion time

Figure 13.5 A/D Conversion Timing

Table 13.4 A/D Conversion Time (Single Mode)

		CKS = 0			CKS = 1		
	Symbol	**Min**	**Typ**	**Max**	**Min**	**Typ**	**Max**
Synchronization delay	t_D	10	—	17	6	—	9
Input sampling time	t_{SPL}	—	63	—	—	31	—
A/D conversion time	t_{CONV}	259	—	266	131	—	134

Note: Values in the table are numbers of states.

13.4.4 External Trigger Input Timing

A/D conversion can be externally triggered. When the TRGE bit is set to 1 in ADCR, external trigger input is enabled at the $\overline{\text{ADTRG}}$ pin. A high-to-low transition at the $\overline{\text{ADTRG}}$ pin sets the ADST bit to 1 in ADCSR, starting A/D conversion. Other operations, in both single and scan modes, are the same as if the ADST bit had been set to 1 by software. Figure 13.6 shows the timing.

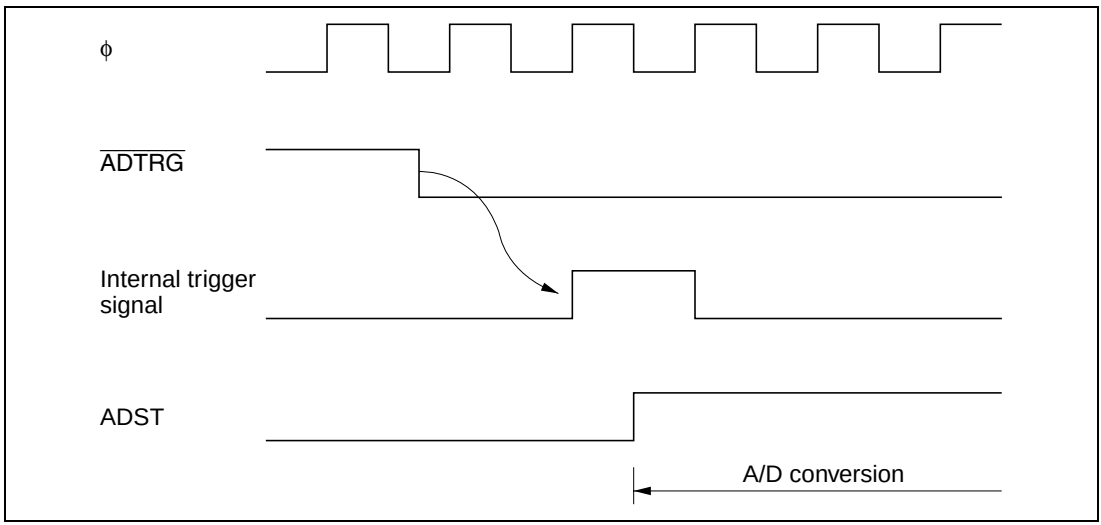


Figure 13.6 External Trigger Input Timing

13.5 Interrupts

The A/D converter generates an interrupt (ADI) at the end of A/D conversion. The ADI interrupt request can be enabled or disabled by the ADIE bit in ADCSR.

13.6 Usage Notes

The following points should be noted when using the A/D converter.

Setting Range of Analog Power Supply and Other Pins

(1) Analog input voltage range

The voltage applied to analog input pins AN_0 to AN_7 during A/D conversion should be in the range $AV_{SS} \leq ANn \leq AV_{CC}$.

(2) Relation between AV_{CC}, AV_{SS} and V_{CC}, V_{SS}

As the relationship between AV_{CC}, AV_{SS} and V_{CC}, V_{SS}, set $AV_{SS} = V_{SS}$. If the A/D converter is not used, the AV_{CC} and AV_{SS} pins must on no account be left open.

If conditions (1) and (2) above are not met, the reliability of the device may be adversely affected.

Notes on Board Design

In board design, digital circuitry and analog circuitry should be as mutually isolated as possible, and layout in which digital circuit signal lines and analog circuit signal lines cross or are in close proximity should be avoided as far as possible. Failure to do so may result in incorrect operation of the analog circuitry due to inductance, adversely affecting A/D conversion values.

Also, digital circuitry must be isolated from the analog input signals (AN_0 to AN_7), and analog power supply and reference voltage (AV_{CC}) by the analog ground (AV_{SS}). Also, the analog ground (AV_{SS}) should be connected at one point to a stable digital ground (V_{SS}) on the board.

Notes on Noise Countermeasures

A protection circuit connected to prevent damage due to an abnormal voltage such as an excessive surge at the analog input pins (AN_0 to AN_7) and analog power supply (AV_{CC}) should be connected between AV_{CC} and AV_{SS} as shown in figure 13.7.

Also, the bypass capacitors connected to AV_{CC} and the filter capacitor connected to AN_0 to AN_7 must be connected to AV_{SS}.

If a filter capacitor is connected as shown in figure 13.7, the input currents at the analog input pins (AN_0 to AN_7) are averaged, and so an error may arise. Also, when A/D conversion is performed frequently, as in scan mode, if the current charged and discharged by the capacitance of the sample-and-hold circuit in the A/D converter exceeds the current input via the input impedance (R_{in}), an error will arise in the analog input pin voltage. Therefore careful consideration is required when deciding the circuit constants.

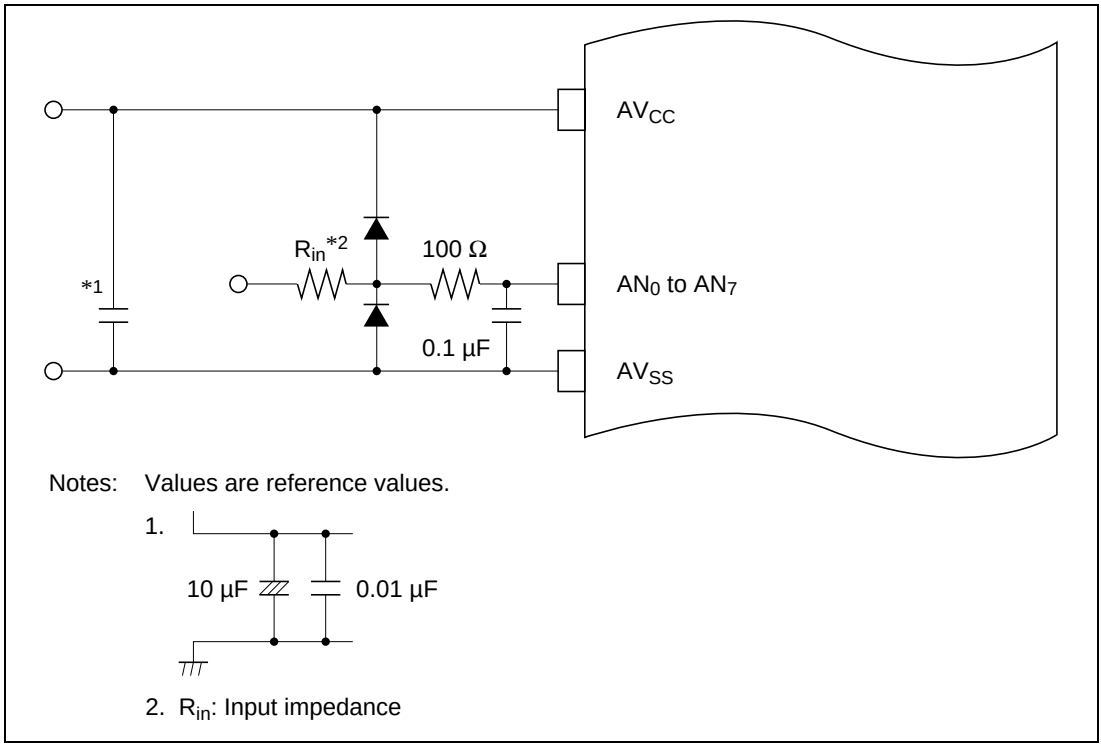
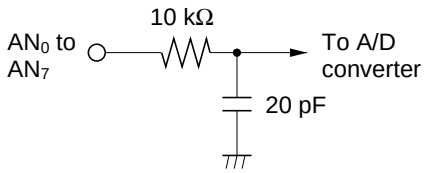


Figure 13.7 Example of Analog Input Protection Circuit

Table 13.5 Analog Pin Specifications

Item	Min	Max	Unit
Analog input capacitance	—	20	pF
Permissible signal source impedance	—	10*	kΩ

Note: * When V_{cc} = 4.0 V to 5.5 V and $\phi \leq 12$ MHz



Note: Values are reference values.

Figure 13.8 Analog Input Pin Equivalent Circuit

A/D Conversion Precision Definitions

H8/3039 Group A/D conversion precision definitions are given below.

- Resolution

  The number of A/D converter digital output codes

- Offset error

  The deviation of the analog input voltage value from the ideal A/D conversion characteristic when the digital output changes from the minimum voltage value B'0000000000 (H'000) to B'0000000001 (H'001) (see figure 13.10).

- Full-scale error

  The deviation of the analog input voltage value from the ideal A/D conversion characteristic when the digital output changes from B'1111111110 (H'3FE) to B'1111111111 (H'3FF) (see figure 13.10).

- Quantization error

  The deviation inherent in the A/D converter, given by 1/2 LSB (see figure 13.9).

- Nonlinearity error

  The error with respect to the ideal A/D conversion characteristic between the zero voltage and the full-scale voltage. Does not include the offset error, full-scale error, or quantization error.

- Absolute precision

  The deviation between the digital value and the analog input value. Includes the offset error, full-scale error, quantization error, and nonlinearity error.

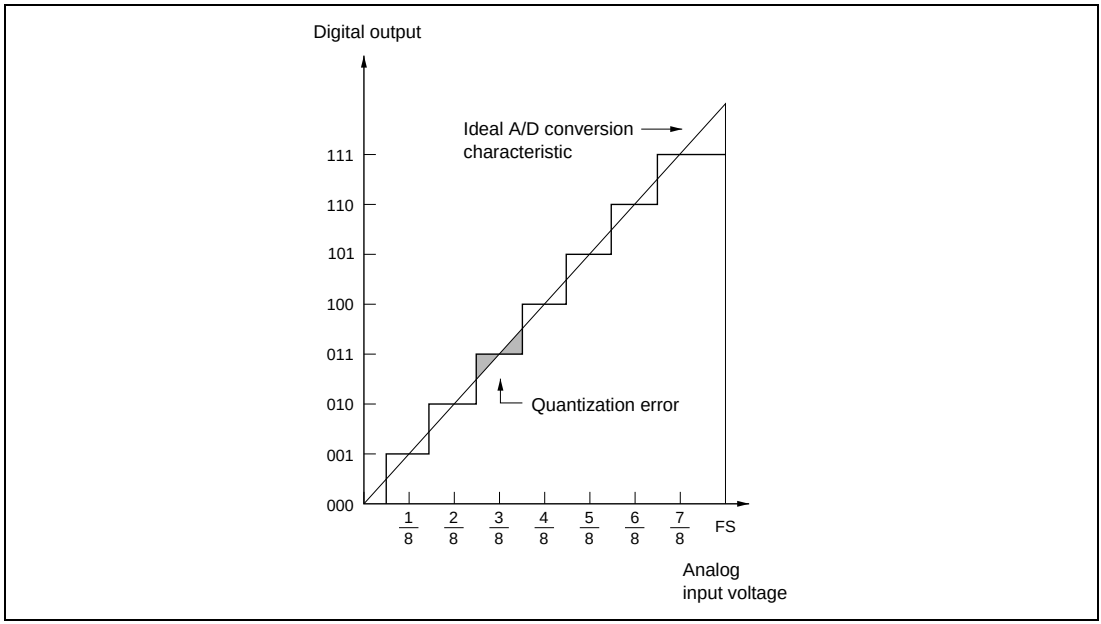


Figure 13.9 A/D Conversion Precision Definitions (1)

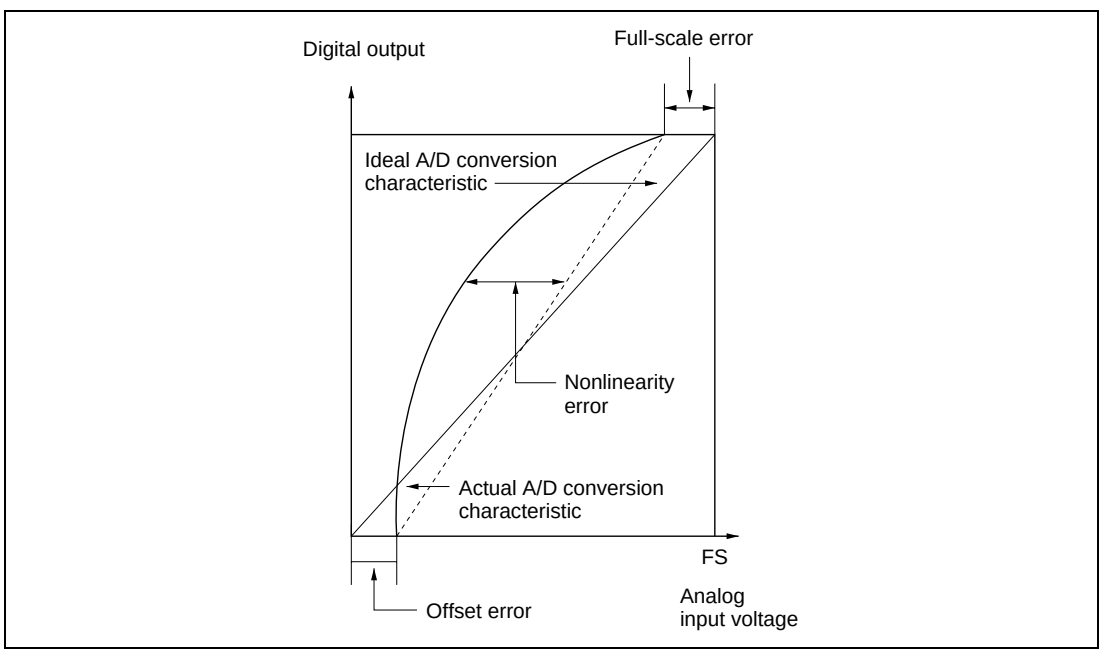


Figure 13.10 A/D Conversion Precision Definitions (2)

Permissible Signal Source Impedance

H8/3039 Group analog input is designed so that conversion precision is guaranteed for an input signal for which the signal source impedance is 10 kΩ or less. This specification is provided to enable the A/D converter's sample-and-hold circuit input capacitance to be charged within the sampling time; if the sensor output impedance exceeds 10 kΩ, charging may be insufficient and it may not be possible to guarantee the A/D conversion precision.

When converting in the single mode, if a large capacitance is provided externally, the input load will essentially comprise only the internal input resistance of 10 kΩ, and the signal source impedance is ignored.

However, since a low-pass filter effect is obtained in this case, it may not be possible to follow an analog signal with a large differential coefficient (e.g., voltage regulation 5 mV/µs or greater).

When converting a high-speed analog signal and when performing conversion in the scan mode, a low-impedance buffer should be inserted.

Influences on Absolute Precision

Adding capacitance results in coupling with GND, and therefore noise in GND may adversely affect absolute precision. Be sure to make the connection to an electrically stable GND such as AV_{SS}.

Care is also required to insure that filter circuits do not communicate with digital signals on the mounting board, thus acting as antennas.

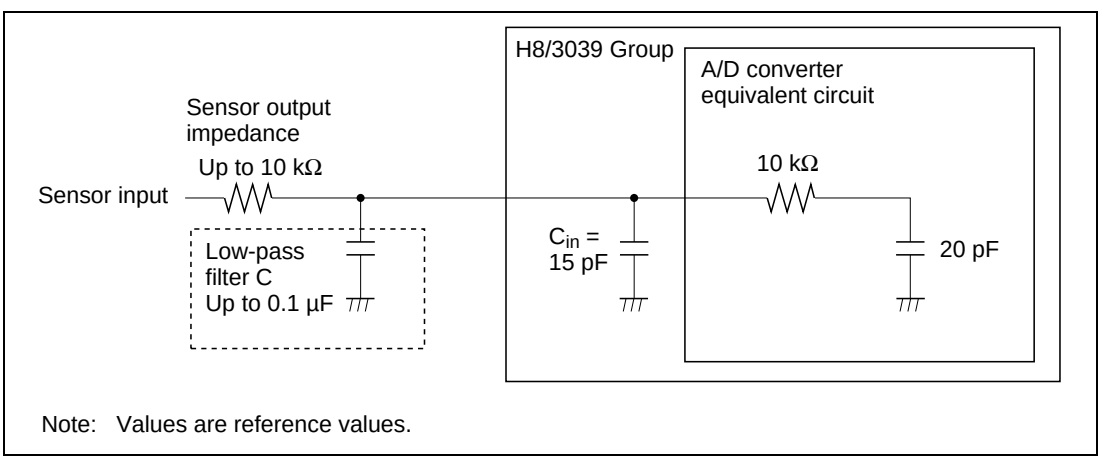


Figure 13.11 Example of Analog Input Circuit

Section 14 RAM

14.1 Overview

The H8/3039 has 4 kbytes of on-chip static RAM, H8/3038 has 2 kbytes, H8/3037 has 1 kbyte, and H8/3036 has 512 bytes. The RAM is connected to the CPU by a 16-bit data bus. The CPU accesses both byte data and word data in two states, making the RAM suitable for rapid data transfer.

The RAM enable bit (RAME) in the system control register (SYSCR) can enable or disable the on-chip RAM.

Table 14.1 shows the address of the on-chip RAM in each operating mode.

Table 14.1 The Address of the On-Chip RAM in Each Operating Mode

Mode	H8/3039 (4 kbytes)	H8/3038 (2 kbytes)	H8/3037 (1k byte)	H8/3036 (512 bytes)
Modes 1, 5, 7	H'FEF10 to H'FFF0F	H'FF710 to H'FFF0F	H'FFB10 to H'FFF0F	H'FFD10 to H'FFF0F
Mode 3	H'FFEF10 to H'FFFF0F	H'FFF710 to H'FFFF0F	H'FFFB10 to H'FFFF0F	H'FFFD10 to H'FFFF0F
Mode 6	H'F710 to H'FF0F	H'F710 to H'FF0F	H'FB10 to H'FF0F	H'FD10 to H'FF0F

14.1.1 Block Diagram

Figure 14.1 shows a block diagram of the on-chip RAM.

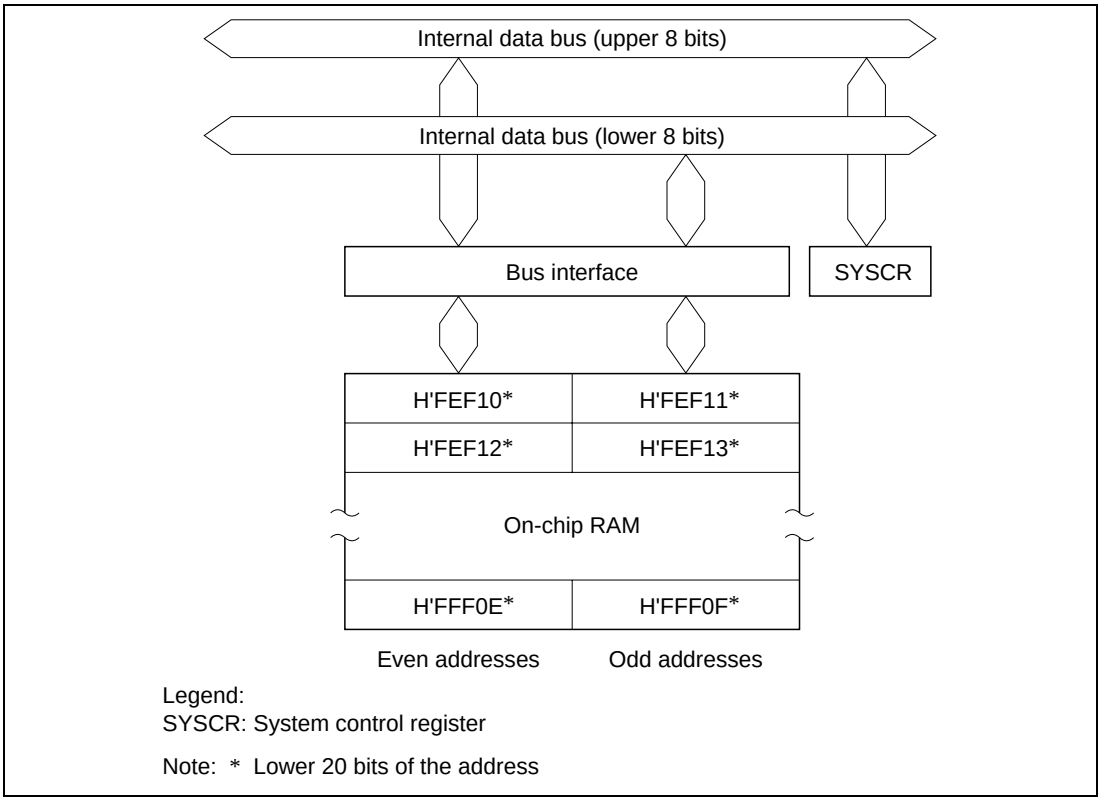


Figure 14.1 RAM Block Diagram (H8/3039 in Modes 1, 5 and 7)

14.1.2 Register Configuration

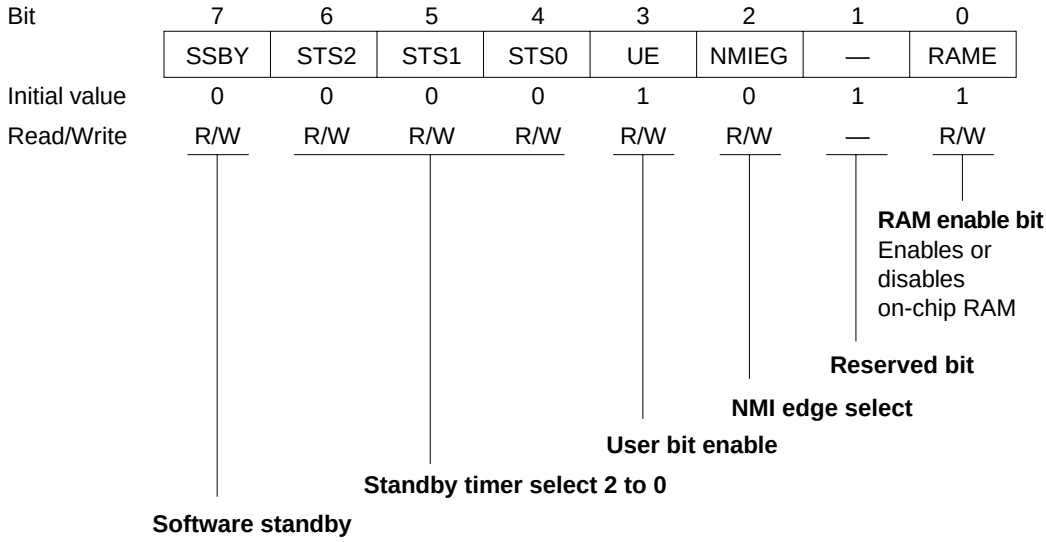
The on-chip RAM is controlled by the system control register (SYSCR). Table 14.2 gives the address and initial value of SYSCR.

Table 14.2 RAM Control Register

Address*	Name	Abbreviation	R/W	Initial Value
H'FFF2	System control register	SYSCR	R/W	H'0B

Note: * Lower 16 bits of the address

14.2 System Control Register (SYSCR)



One function of SYSCR is to enable or disable access to the on-chip RAM. The on-chip RAM is enabled or disabled by the RAME bit in SYSCR. For details about the other bits, see section 3.3, System Control Register (SYSCR).

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

Bit 0 RAME	Description	
0	On-chip RAM is disabled	
1	On-chip RAM is enabled	(Initial value)

14.3 Operation

When the RAME bit is set to 1, the on-chip RAM is enabled. This LSI can access the on-chip RAM when addressing the addresses shown in table 14.1 in each operation mode. When the RAME bit is cleared to 0 in modes 1, 3, and 5 (expanded modes), external address space is accessed. When the RAME bit is cleared to 0 in modes 6 and 7 (single-chip modes), the on-chip RAM is not accessed. Read operation always reads H'FF and disables writing.

The on-chip RAM is connected to the CPU by a 16-bit wide data bus and can be read and written on a byte or a word basis.

Byte data can be accessed in two states using the higher 8 bits of the data bus. Word data beginning from an even address can be accessed in two states using the 16-bit data bus.

Section 15 ROM

15.1 Overview

The H8/3039 has 128 kbytes of on-chip ROM (flash memory or mask ROM), the H8/3038 has 64 kbytes, the H8/3037 has 32 kbytes and H8/3036 has 16 kbytes. The ROM is connected to the CPU by a 16-bit data bus. The CPU accesses both byte and word data in two states, enabling rapid data transfer.

The mode pins (MD_2 to MD_0) can be set to enable or disable the on-chip ROM. See table 15.1.

The on-chip flash memory product (H8/3039F-ZTAT) can be erased and programmed on-board as well as with a general-purpose PROM programmer.

Table 15.1 Operating Mode and ROM

	Mode Pins			
Mode	**MD_2**	**MD_1**	**MD_0**	**On-Chip ROM**
Mode 1 (1-Mbyte expanded mode with on-chip ROM disabled)	0	0	1	Disabled (external address area)
Mode 2 (1-Mbyte expanded mode with on-chip ROM disabled)*	0	1	0	
Mode 3 (16-Mbyte expanded mode with on-chip ROM disabled)	0	1	1	
Mode 4 (16-Mbyte expanded mode with on-chip ROM disabled)*	1	0	0	
Mode 5 (16-Mbyte expanded mode with on-chip ROM enabled)	1	0	1	Enabled
Mode 6 (single-chip normal mode)	1	1	0	
Mode 7 (single-chip advanced mode)	1	1	1	

Note: * Modes 2 and 4 cannot be used with this LSI. Do not set the mode pin to mode 2 or 4.

15.2 Overview of Flash Memory

15.2.1 Features

The features of the flash memory are summarized below.

- Four flash memory operating modes
  - Program mode
  - Erase mode
  - Program-verify mode
  - Erase-verify mode
- Programming/erase methods

  The flash memory is programmed 32 bytes at a time. Erasing is performed by block erase. The block to be erased can be specified by setting the corresponding bit. There are block areas of 32 kbytes × 3 blocks, 28 kbytes × 1 block, and 1 kbyte × 4 blocks.
- Programming/erase times

  The flash memory programming time is 10 ms (typ.) for simultaneous 32-byte programming, equivalent to 300 µs (typ.) per byte, and the erase time is 100 ms (typ.) per block.
- Reprogramming capability

  The flash memory can be reprogrammed up to 100 times.
- On-board programming modes

  There are two modes in which flash memory can be programmed/erased/verified on-board:
  - Boot mode
  - User program mode
- Automatic bit rate adjustment

  With data transfer in boot mode, the this LSI's bit rate can be automatically adjusted to match the transfer bit rate of the host (9600 bps and 4800 bps).
- Flash memory emulation by RAM

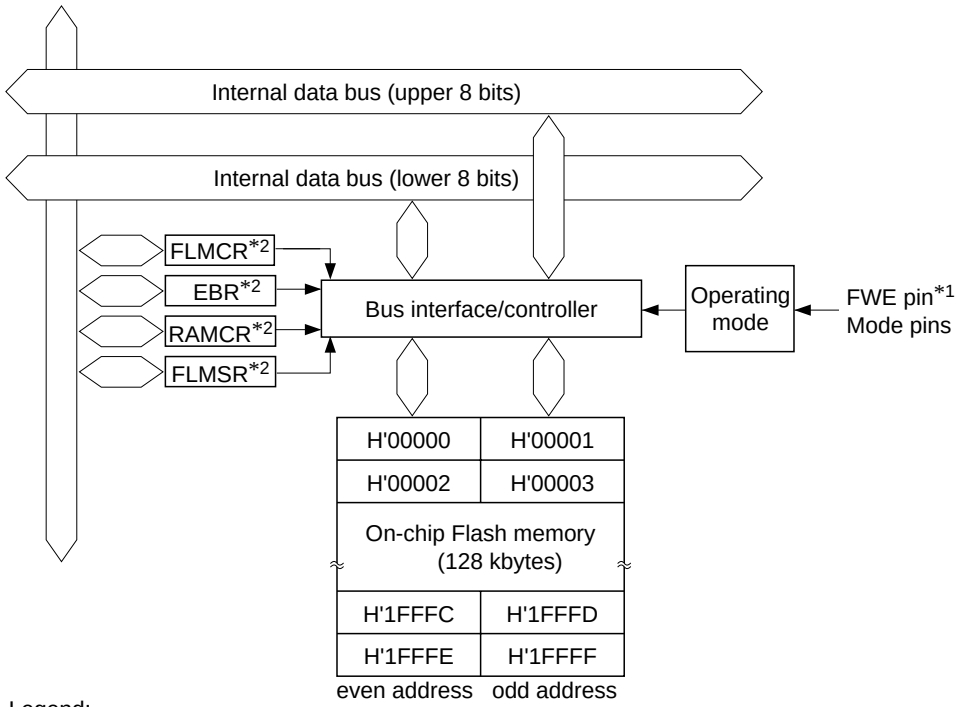
  Part of the RAM area can be overlapped onto flash memory, to emulate flash memory updates in real time.
- PROM mode

  Flash memory can be programmed/erased in PROM mode, using a PROM programmer, as well as in on-board programming mode.
- Protect modes

  There are three protect modes, hardware, software, and error protect, which allow protected status to be designated for flash memory program/erase/verify operations.

15.2.2 Block Diagram

Figure 15.1 shows a block diagram of the flash memory.



Legend:
FLMCR: Flash memory control register
EBR: Erase block register
RAMCR: RAM control register
FLMSR: Flash memory status register

Notes: 1. Functions as the FWE pin in the flash memory versions and as the $\overline{\text{RESO}}$ pin in the mask ROM versions.
2. The registers that control the flash memory versions (FLMCR, EBR, RAMCR, and FLMSR) are used in the flash memory versions only. They are not provided in the mask ROM versions. Reading the corresponding addresses in a mask ROM version will always return 1s, and writes to these addresses are disabled.

Figure 15.1 Block Diagram of Flash Memory

15.2.3 Pin Configuration

The flash memory is controlled by means of the pins shown in table 15.2.

Table 15.2 Flash Memory Pins

Pin Name	Abbreviation	I/O	Function
Reset	$\overline{RES}$	Input	Reset
Flash write enable	FWE*	Input	Flash program/erase protection by hardware
Mode 2	MD_2	Input	Sets this LSI operating mode
Mode 1	MD_1	Input	Sets this LSI operating mode
Mode 0	MD_0	Input	Sets this LSI operating mode
Transmit data	TxD_1	Output	Serial transmit data output
Receive data	RxD_1	Input	Serial receive data input

Notes: The transmit data and receive data pins are used in boot mode.

* In the mask ROM versions, the FWE pin functions as the $\overline{RESO}$ pin.

15.2.4 Register Configuration

The registers used to control the on-chip flash memory when enabled are shown in table 15.3.

Table 15.3 Flash Memory Registers

Register Name	Abbreviation	R/W	Initial Value	Address*[1]
Flash memory control register	FLMCR	R/W	H'00*[2]	H'FF40
Erase block register	EBR	R/W	H'00	H'FF42
RAM control register	RAMCR	R/W	H'F1	H'FF47
Flash memory status register	FLMSR	R	H'7F	H'FF4D

Notes: 1. Lower 16 bits of the address.

2. When a high level is input to the FWE pin, the initial value is H'80.

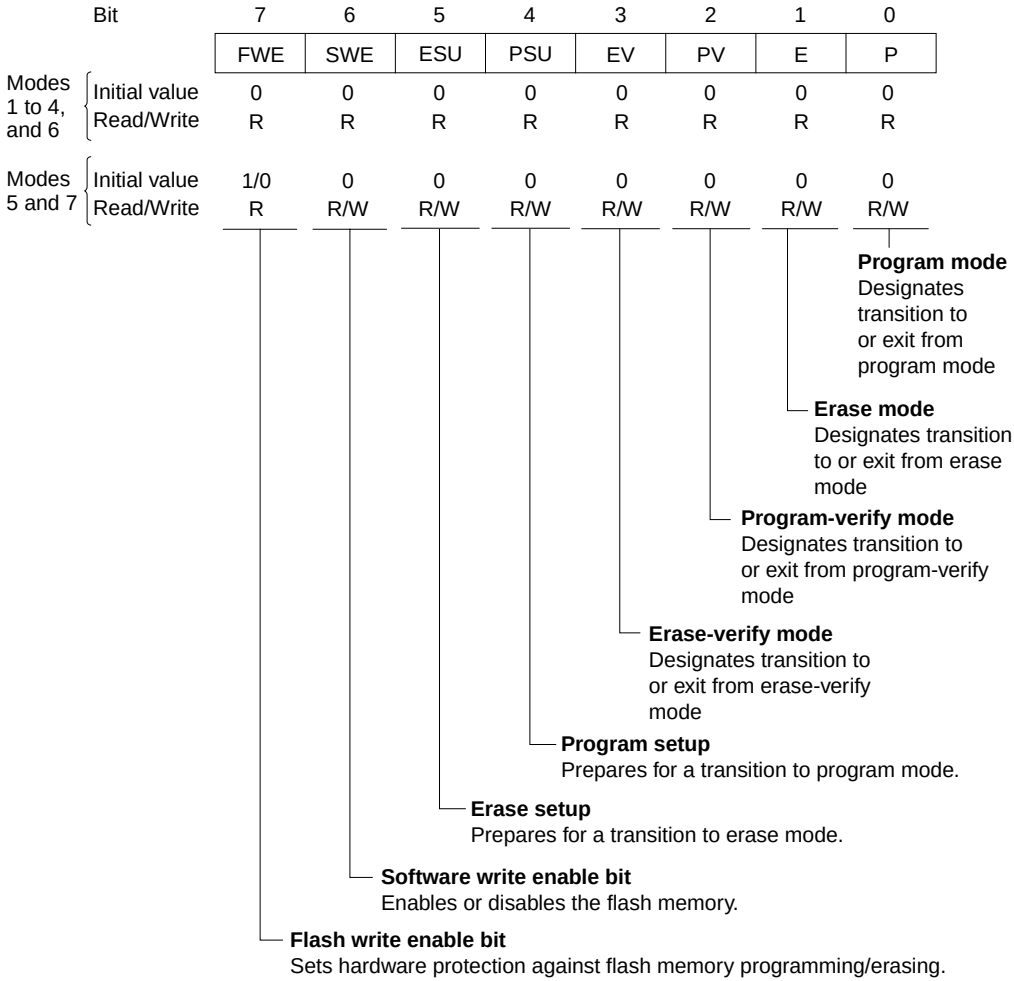
The registers in table 15.3 are used in the flash memory versions only. Reading the corresponding addresses in a mask ROM version will always return 1s, and writes to these addresses are disabled.

15.3 Register Descriptions

15.3.1 Flash Memory Control Register (FLMCR)

FLMCR is an 8-bit register used for flash memory operating mode control. Program-verify mode or erase-verify mode is entered by setting SWE to 1 when FWE = 1. Program mode is entered by setting SWE to 1 when FWE = 1, then setting the PSU bit, and finally setting the P bit. Erase mode is entered by setting SWE to 1 when FWE = 1, then setting the ESU bit, and finally setting the E bit. FLMCR is initialized by a reset, and in hardware standby mode and software standby mode. Its initial value is H'80 when a high level is input to the FWE pin, and H'00 when a low level is input. In mode 6 the FWE pin must be fixed low, as flash memory on-board programming is not supported. Therefore, bits in this register cannot be set to 1 in mode 6. When on-chip flash memory is disabled, a read will return H'00, and writes are invalid. When setting bits 6 to 0 in this register to 1, each bit should be set individually.

Writes to the ESU, PSU, EV and PV bits in FLMCR are enabled only when FWE = 1 and SWE = 1; writes to the E bit only when FWE = 1, SWE = 1, and ESU = 1; and writes to the P bit only when FWE = 1, SWE = 1, and PSU = 1.



Bit 7—Flash Write Enable Bit (FWE): Sets hardware protection against flash memory programming/erasing. When using this bit, refer to section 15.9, Notes on Flash Memory Programming/Erasing.

Bit 7 FWE	Description
0	When a low level is input to the FWE pin (hardware-protected state)
1	When a high level is input to the FWE pin

Bit 6—Software Write Enable Bit (SWE)*[1] *[2]: This bit enables/disables flash memory programming/erasing. This bit should be set before setting FLMCR bits 5 to 0, and EBR bits 7 to 0. Do not set the ESU, PSU, EV, PV, E, or P bits at the same time.

Bit 6 SWE	Description	
0	Program/erase disabled	(Initial value)
1	Program/erase enabled [Setting condition] When FWE = 1	

Bit 5—Erase Setup Bit (ESU)*[1]: Prepares for a transition to erase mode. Do not set the SWE, PSU, EV, PV, E, or P bit at the same time.

Bit 5 ESU	Description	
0	Erase setup cleared	(Initial value)
1	Erase setup [Setting condition] When FWE = 1, and SWE = 1	

Bit 4—Program Setup Bit (PSU)*[1]: Prepares for a transition to program mode. Do not set the SWE, ESU, EV, PV, E, or P bit at the same time.

Bit 4 PSU	Description	
0	Program setup cleared	(Initial value)
1	Program setup [Setting condition] When FWE = 1, and SWE = 1	

Bit 3—Erase-Verify (EV)*[1]: Selects erase-verify mode transition or clearing. Do not set the SWE, ESU, PSU, PV, E, or P bit at the same time.

Bit 3 EV	Description	
0	Erase-verify mode cleared	(Initial value)
1	Transition to erase-verify mode [Setting condition] When FWE = 1, and SWE = 1	

Bit 2—Program-Verify (PV)*[1]: Selects program-verify mode transition or clearing. Do not set the SWE, ESU, PSU, EV, E, or P bit at the same time.

Bit 2 PV	Description	
0	Program-verify mode cleared	(Initial value)
1	Transition to program-verify mode [Setting condition] When FWE = 1, and SWE = 1	

Bit 1—Erase (E)*[1]*[3]: Selects erase mode transition or clearing. Do not set the SWE, ESU, PSU, EV, PV, or P bit at the same time.

Bit 1 E	Description	
0	Erase mode cleared	(Initial value)
1	Transition to erase mode [Setting condition] When FWE = 1, SWE = 1, and ESU = 1	

Bit 0—Program (P)*[1] *[3]: Selects program mode transition or clearing. Do not set the SWE, ESU, PSU, EV, PV, or E bit at the same time.

Bit 0 P	Description	
0	Program mode cleared	(Initial value)
1	Transition to program mode [Setting condition] When FWE = 1, SWE = 1, and PSU = 1	

Notes: 1. Do not set two or more bits at the same time.
Do not turn off V_{CC} when a bit is set.
2. Do not set/clear the SWE bit simultaneously with other bits (ESU, PSU, EV, PV, E, P).
3. Set the P and E bits according to the program and erase algorithms shown in section 15.5, Programming/Erasing Flash Memory.
For the usage precautions, see section 15.9, Notes on Flash Memory Programming/Erasing.

15.3.2 Erase Block Register (EBR)

EBR is an 8-bit register that designates the flash memory block for erasure. EBR is initialized to H'00 by a reset, in hardware standby mode, or software standby mode, when a high level is not input to the FWE terminal, or when the FLMCR SWE bit is 0 when a high level is applied to the FWE terminal. When a bit is set in EBR, the corresponding block can be erased. Other blocks are erase - protected. The blocks are erased block by block. Therefore, set only one bit in EBR; do not set bits in EBR to erase two or more blocks at the same time.

Each bit in EBR cannot be set until the SWE bit in FLMCR is set. The flash memory block configuration is shown in table 15.4. To erase all the blocks, erase each block sequentially.

This LSI does not support the on-board programming mode in mode 6, so bits in this register cannot be set to 1 in mode 6.

	Bit	7	6	5	4	3	2	1	0
		EB7	EB6	EB5	EB4	EB3	EB2	EB1	EB0
Modes 1 to 4, and 6	Initial value	0	0	0	0	0	0	0	0
	Read/Write	R	R	R	R	R	R	R	R
Modes 5 and 7	Initial value	0	0	0	0	0	0	0	0
	Read/Write	R/W	R/W	R/W	R/W	R/W	R/W	R/W	R/W

Bits 7 to 0—Block 7 to 0 (EB7 to EB0): These bits select blocks (EB7 to EB0) to be erased.

Bits 7 to 0 EB7 to EB0	Description	
0	Block EB7 to EB0 is not selected.	(Initial value)
1	Block EB7 to EB0 is selected.	

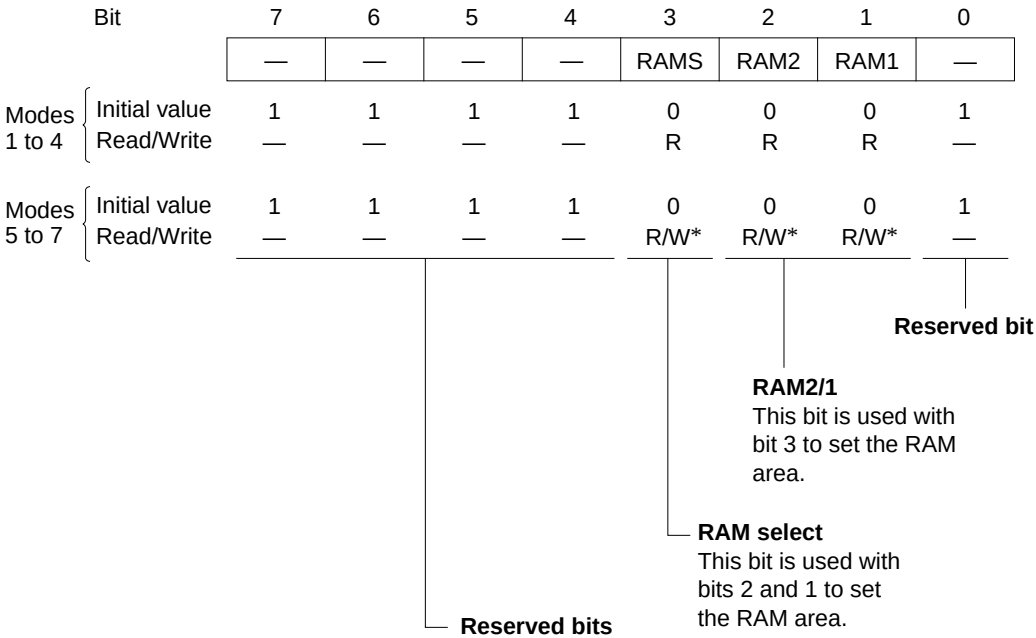
Note: Set each bit of EBR to H'00 except when erasing.

Table 15.4 Flash Memory Erase Blocks

Block (Size)	Address
EB0 (1 kbyte)	H'00000 to H'003FF
EB1 (1 kbyte)	H'00400 to H'007FF
EB2 (1 kbyte)	H'00800 to H'00BFF
EB3 (1 kbyte)	H'00C00 to H'00FFF
EB4 (28 kbytes)	H'01000 to H'07FFF
EB5 (32 kbytes)	H'08000 to H'0FFFF
EB6 (32 kbytes)	H'10000 to H'17FFF
EB7 (32 kbytes)	H'18000 to H'1FFFF

15.3.3 RAM Control Register (RAMCR)

RAMCR selects the RAM area used when emulating real-time reprogramming of the flash memory.



Note: * Cannot be set to 1 in mode 6.

Bits 7 to 4—Reserved: These bits cannot be modified and are always read as 1.

Bit 3—RAM Select (RAMS): Is used with bits 2 to 1 to reassign an area to RAM (see table 15.5). The initial setting for this bit is 0 in modes 5, 6, and 7 (internal flash memory enabled) and programming is enabled.* In modes other than 5 to 7, 0 is always read and writing is disabled. It is initialized by a reset and in hardware standby mode. It is not initialized in software standby mode.
When bit 3 is set, all flash-memory blocks are protected from programming and erasing.

Bits 2 to 1—RAM2 to RAM1: These bits are used with bit 3 to reassign an area to RAM (see table 15.5). The initial setting for this bit is 0 in modes 5, 6, and 7 (internal flash memory enabled) and programming is enabled.* In modes other than 5 to 7, 0 is always read and writing is disabled. They are initialized by a reset and in hardware standby mode. They are not initialized in software standby mode.

Bit 0—Reserved: This bit cannot be modified and is always read as 1.

Note: * Flash memory emulation by RAM is not supported for Mode 6 (single chip normal mode), so programming is possible, but do not set 1.

When performing flash memory emulation by RAM, the RAME bit in SYSCR must be set to 1.

Table 15.5 RAM Area Reassignment

RAM Area	Bit 3 RAMS	Bit 2 RAM2	Bit 1 RAM1	RAM Emulation State
H'FFF800 to H'FFFBFF	0	0/1	0/1	No emulation
H'000000 to H'0003FF	1	0	0	Mapping RAM
H'000400 to H'0007FF	1	0	1	
H'000800 to H'000BFF	1	1	0	
H'000C00 to H'000FFF	1	1	1	

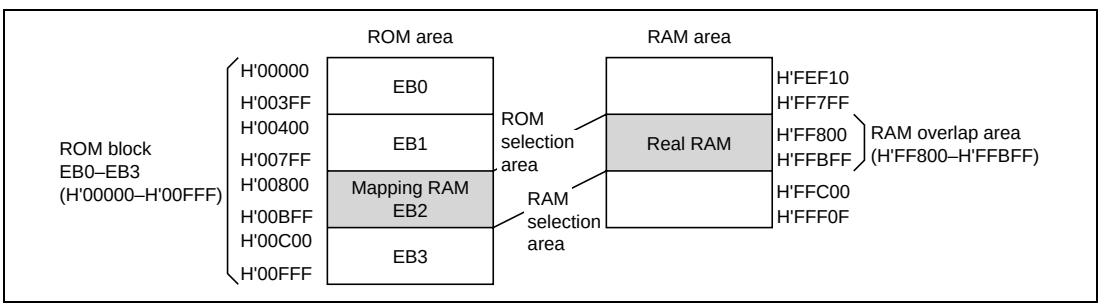
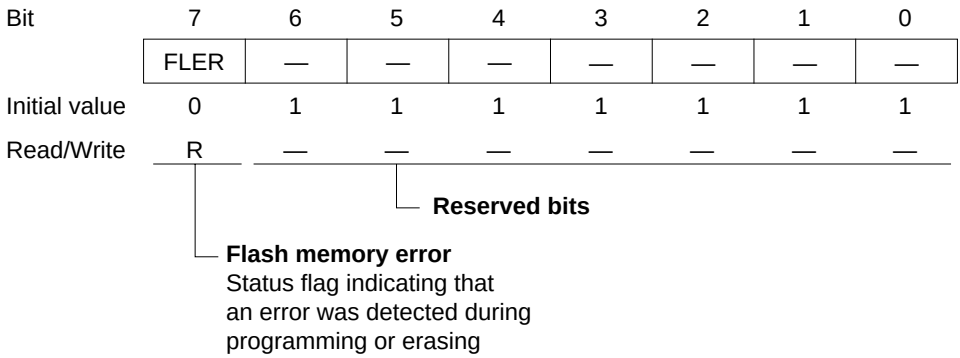


Figure 15.2 Example of Overlap ROM Area and RAM Area

15.3.4 Flash Memory Status Register (FLMSR)

The flash memory status register (FLMSR) detects flash memory errors.



Bit 7—Flash Memory Error (FLER): Indicates that an error occurred while flash memory was being programmed or erased. When bit 7 is set, flash memory is placed in an error-protect mode.

Bit 7 FLER	Description
0	Flash memory program/erase protection (error protection*[1]) is disabled (Initial value) [Clearing condition] WDT reset, reset by $\overline{\text{RES}}$ pin, or hardware standby mode
1	An error has occurred during flash memory programming/erasing Flash memory program/erase protection (error protection*[1]) is enabled [Setting conditions] 1. Flash memory was read*[2] while being programmed or erased (including vector or instruction fetch, but not including reading of a RAM area overlapped onto flash memory). 2. A hardware exception-handling sequence (other than a reset, invalid instruction, trap instruction, or zero-divide exception) was executed just before programming or erasing.*[3] 3. The SLEEP instruction (including software standby mode) was executed during programming or erasing.

Notes: 1. For details, see section 15.6.3, Error Protection.
2. The read data has undetermined values.
3. Before stack and vector read by exception handling.

Bits 6 to 0—Reserved: These bits cannot be modified and are always read as 1.

15.4 On-Board Programming Modes

When pins are set to on-board programming mode, program/erase/verify operations can be performed on the on-chip flash memory. There are two on-board programming modes: boot mode and user program mode. The pin settings for transition to each of these modes are shown in table 15.6. In mode 6 (on-chip ROM enabled) in this LSI, the boot mode and user program mode cannot be used. For the notes on FWE pin set/reset, see section 15.9, Notes on Flash Memory Programming/Erasing.

Table 15.6 Setting On-Board Programming Modes

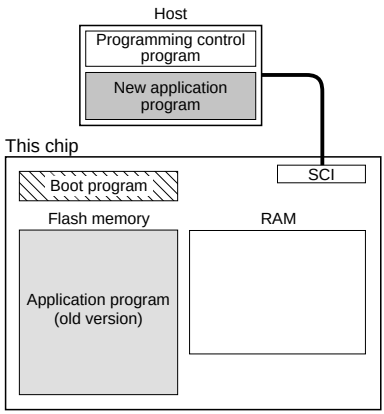
Mode		FWE	MD_2	MD_1	MD_0	Notes
Boot mode	mode 5	1*[1]	0*[2]	0	1	0: V_{IL}
	mode 7		0*[2]	1	1	1: V_{IH}
User program mode	mode 5		1	0	1	
	mode 7		1	1	1	

Notes: 1. For the High level input timing, see items (6) and (7) of Notes on Using the Boot Mode.
2. In the boot mode, the MD_2 setting becomes inverted input.
In the boot mode, the mode control register (MDCR) can be used to monitor the status of modes 5 and 7 in the same way as in the normal mode.

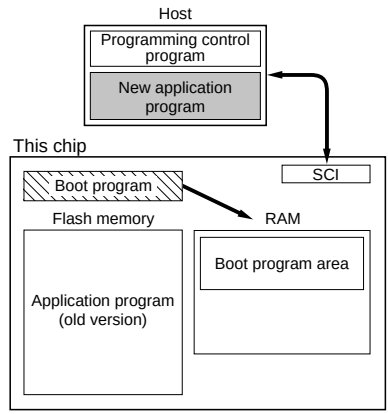
On-Board Programming Modes

- Boot mode

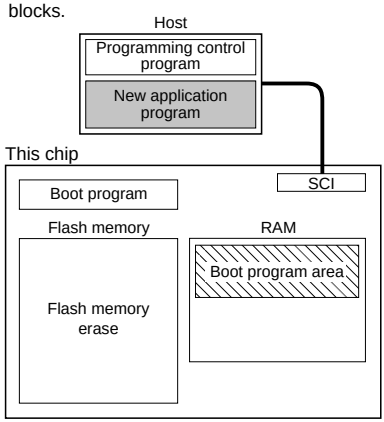
1. Initial state
   The flash memory is in the erased state when the device is shipped. The description here applies to the case where the old program version or data is being rewritten. The user should prepare the programming control program and new application program beforehand in the host.



2. Programming control program transfer
   When boot mode is entered, the boot program in this chip (originally incorporated in the chip) is started, an SCI communication check is carried out, and the boot program required for flash memory erasing is automatically transferred 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, entire flash memory erasure is performed, without regard to blocks.



4. Writing new application program
   The programming control program transferred from the host to RAM by SCI communication is executed, and the new application program in the host is written into the flash memory.

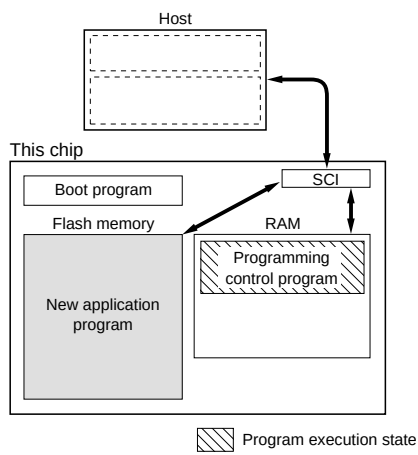
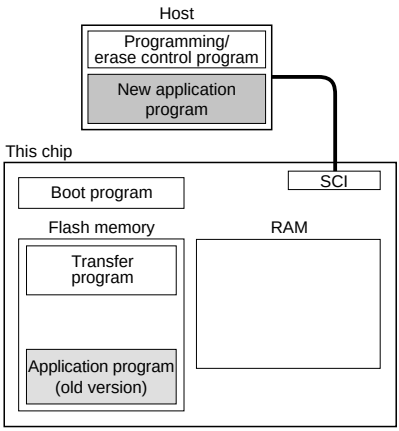


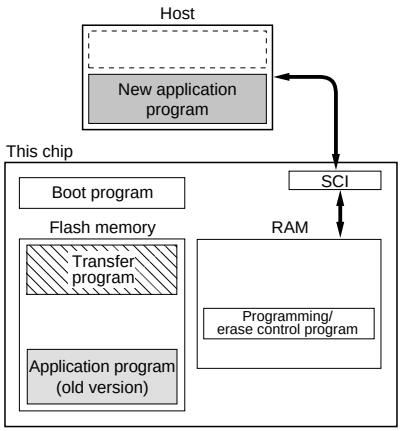
Figure 15.3 Boot Mode

- User program mode

1. Initial state
   (1) The program that will transfer the programming/ erase control program to on-chip RAM should be written into the flash memory by the user beforehand. (2) The programming/erase control program should be prepared in the host or in the flash memory.

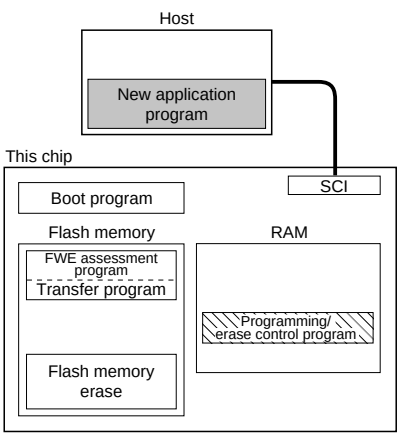
2. Programming/erase control program transfer
   When the FWE pin is driven high, user software confirms this fact, executes the transfer program in the flash memory, and transfers the programming/erase control program to RAM.





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 host is written into the erased flash memory blocks. Do not write to unerased blocks.



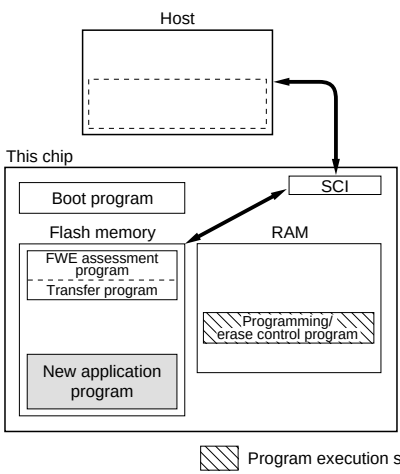


Figure 15.4 User Program Mode (Example)

15.4.1 Boot Mode

When boot mode is used, the flash memory programming control program must be prepared in the host beforehand. The channel 1 SCI to be used is set to asynchronous mode.

In reset start, after setting this LSI pin to the boot mode, start the microcomputer boot program, measure the Low period of the data sent from the host, and select the bit rate register (BRR) value beforehand. Then enable reception of the user program from the outside using the serial communication interface (SCI) on this LSI, and write the received user program to on-chip RAM.

After the program has been stored the end of writing, execution branches to the top address (H'FF300) of the on-chip RAM, execute the program written on the on-chip RAM, and enable flash memory program/erase.

The system configuration in boot mode is shown in figure 15.5, and the boot program mode execution procedure in figure 15.6.

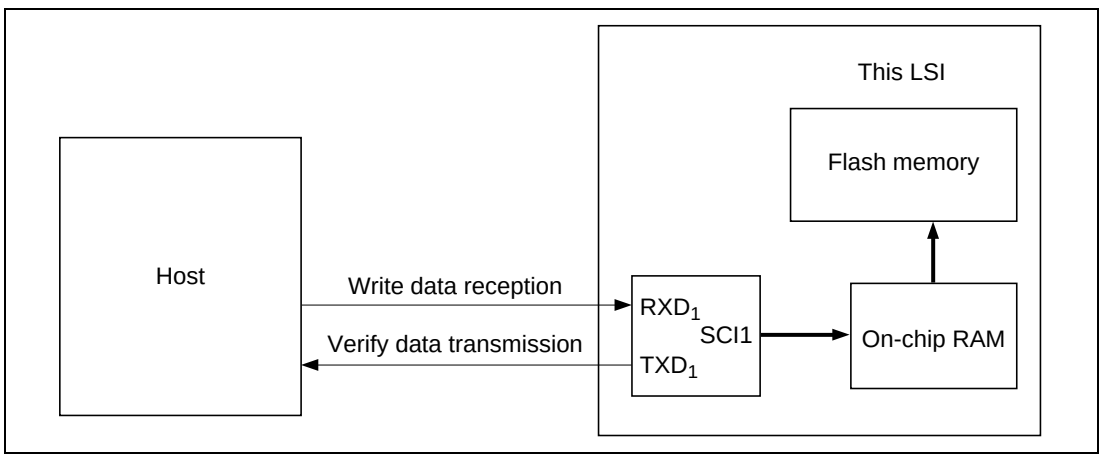
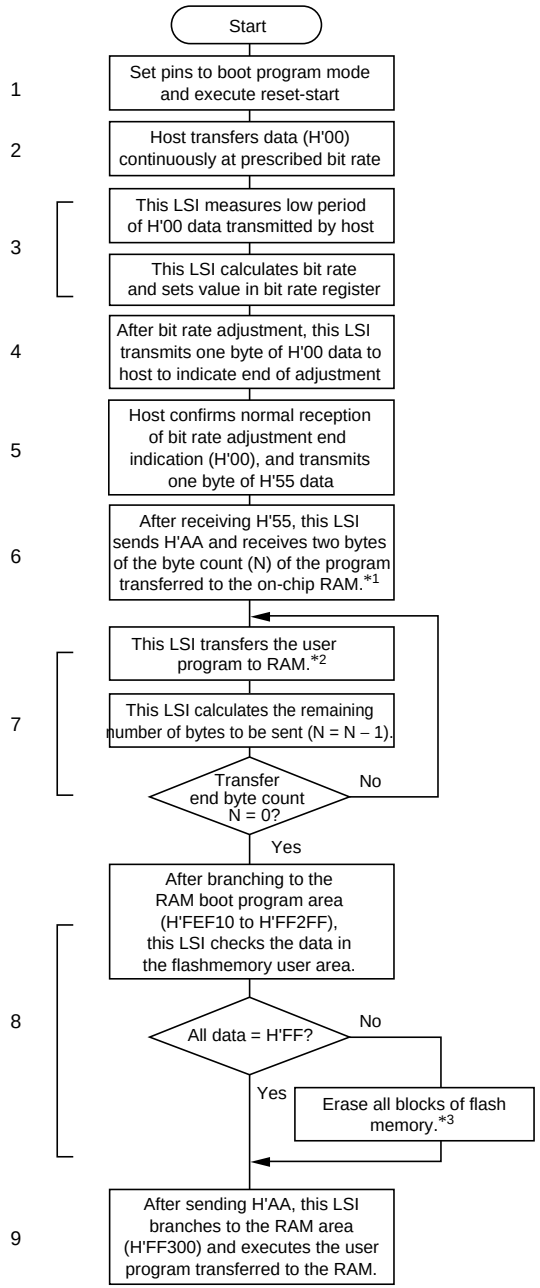


Figure 15.5 System Configuration in Boot Mode



1 Set this LSI to the boot mode and reset starts the LSI.

2 Set the host to the prescribed bit rate (4800, 9600) and consecutively send H'00 data in 8-bit data, 1 stop bit format.

3 This LSI repeatedly measures the RXD1 pin Low period and calculates the asynchronous communication bit rate at which the host performs transfer.

4 At the end of SCI bit rate adjustment, this LSI sends one byte of H'00 data to signal the end of adjustment.

5 Check if the host normally received the one byte bit rate adjustment end signal sent from this LSI and sent one byte of H'55 data.

6 After H'55 is sent, the host receives H'AA and sends the byte count of the user program that is transferred to this LSI.
Send the 2-byte count in upper byte and lower byte order. Then sequentially send the program set by the user.
This LSI sequentially sends (echo back) each byte of the received byte count and user program to the host as verification data.

7 This LSI sequentially writes the received user program to the on-chip RAM area (H'FF300–H'FFEFF).

8 Before executing the transferred user program, this LSI checks if data was written to flash memory after control branched to the RAM boot program area (H'FEF10–H'FF2FF). If data was already written to flash memory, all the blocks are erased.

9 After sending H'AA, this LSI branches to the on-chip RAM area (H'FF300) and executes the user program written to that area.

Notes: 1. The RAM area that can be used by the user is 3.0 kbytes. Set the transfer byte count to within 3.0 kbytes. Always send the 2-byte transfer byte count in upper byte and lower byte order.
Transfer byte count example: For 256 bytes (H'0100), upper byte H'01, lower byte H'00.

2. Set the part that controls the user program flash memory at the program according to the flash memory programming/erase algorithms described later.

3. When a memory cell malfunctions and cannot be erased, this LSI sends one H'FF byte as an erase error and stops the erase operation and subsequent operations.

Figure 15.6 Boot Mode Execution Procedure

Automatic SCI Bit Rate Adjustment

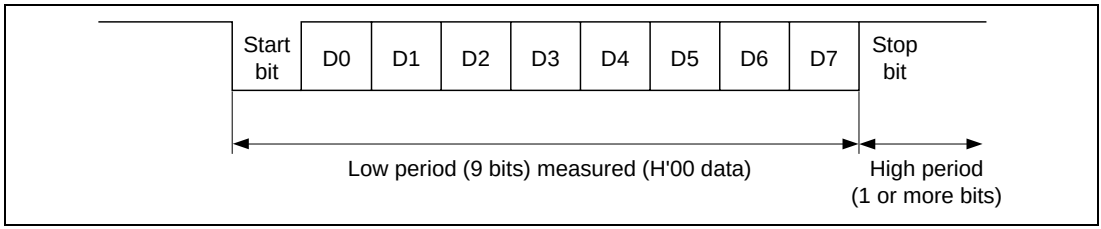


Figure 15.7 Measuring the Low Period of the Communication Data from the Host

When boot mode is initiated, this LSI measures the low period of the asynchronous SCI communication data (H'00) transmitted continuously from the host (figure 15.7). The SCI transmit/receive format should be set as follows: 8-bit data, 1 stop bit, no parity. This LSI calculates the bit rate of the transmission from the host from the measured low period, and transmits one H'00 byte to the host to indicate the end of bit rate adjustment. The host should confirm that this adjustment end indication (H'00) has been received normally, and transmit one H'55 byte to the LSI. If reception cannot be performed normally, initiate boot mode again (reset), and repeat the above operations. Depending on the host's transmission bit rate and the system clock frequency of this LSI, there will be a discrepancy between the bit rates of the host and the LSI. To ensure correct SCI operation, the host's transfer bit rate should be set to 4800 and 9600 bps*[1].

Table 15.7 shows typical host transfer bit rates and system clock frequencies for which automatic adjustment of this LSI bit rate is possible. The boot program should be executed within this system clock range*[2].

Table 15.7 System Clock Frequencies for which Automatic Adjustment of This LSI Bit Rate Is Possible

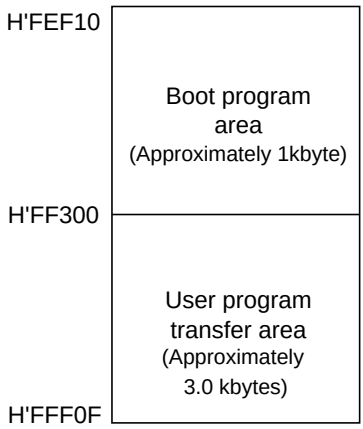
Host Bit Rate (bps)	System Clock Frequency for which Automatic Adjustment of This LSI Bit Rate Is Possible (MHz)
9600	8 to 18
4800	4 to 18

Notes: 1. The host bit rate settings are 4800 and 9600bps only. Do not use any other setting.

2. This LSI may automatically adjusts the bit rate except for bit rate and system clock combinations as shown in table 15.7. However, the bit rate of the host and this LSI will be different and subsequent transfers will not be carried out normally. Therefore, always execute the boot mode within the range of the bit rate and system clock combinations shown in table 15.7.

On-Chip RAM Area Divisions in Boot Mode

In boot mode, the RAM area is divided into an area used by the boot program and an area to which the user program is transferred via the SCI, as shown in figure 15.8. The boot program area can be used when a transition is made to the execution state for the user program transferred to RAM.



Notes: 1. The boot program area cannot be used until a transition is made to the execution state for the user program transferred to RAM. Note also that the boot program remains in this RAM area even after control branches to the user program.
2. When flash memory emulation is performed using RAM, part of the user program transfer area (H'FF800 to H'FFBFF) is used as an area for carrying out emulation, and therefore user program transfer must not be performed to this area.

Figure 15.8 RAM Areas in Boot Mode

Notes on using the boot mode

(1) When this LSI comes out of reset in boot mode, it measures the low period the input at the SCI's RXD_1 pin. The reset should end with RXD_1 high. After the reset ends, it takes about 100 states for this LSI to get ready to measure the low period of the RXD_1 input.

(2) If any data has been written to the flash memory (if all data is not H'FF), all flash memory blocks are erased when this mode is executed. Therefore, boot mode should be used for initial on-board programming, or for forced recovery if the program to be activated in user program mode is 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) (by clearing the RE and TE bits in serial control register (SCR)) before branching to the user program. However, the adjusted bit rate is held in the bit rate register (BRR). At this time, the TXD_1 pin is in the high level output state (P9DDR $P9_1$DDR=1, P9DR $P9_1$DR=1).

Before branching to the user program the value of the general registers in the CPU are also undefined. Therefore, the general registers must be initialized immediately after control branches to the user program. Since the stack pointer (SP) is implicitly used during subroutine call, etc., a stack area must be specified for use by the user 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 to MD_2 and FWE pins according to the mode setting conditions shown in table 15.6.

At this time, this LSI latches the status of the mode pin inside the microcomputer to maintain the boot mode status at the reset clear (startup with Low → High) timing*[1].

To clear boot mode, it is necessary to drive the FWE pin low during the reset, and then execute reset release*[1]. The following points must be noted:

(a) Before making a transition from the boot mode to the regular mode, the microcomputer boot mode must be reset by reset input via the $\overline{RES}$ pin. At this time, the $\overline{RES}$ pin must be hold at low level for at least 20 system clock.*[3]

(b) Do not change the input levels at the mode pins (MD_2 to MD_0) or the FWE pin while in boot mode. When making a mode transition, first enter the reset state by inputting a low level to the $\overline{RES}$ pin. When a watchdog timer reset was generated in the boot mode, the microcomputer mode is not reset and the on-chip boot program is restarted regardless of the state of the mode pin.

(c) Do not input low level to the FWE pin while the boot program is executing and when programming/erasing flash memory.*[2]

(7) If the mode pin and FWE pin input levels are changed from 0 V to V_{CC} or from V_{CC} to 0V during a reset (while a low level is being input to the $\overline{RES}$ pin), the microcomputer's operating mode will change.

Therefore, since the state of the address dual port and bus control output signals ($\overline{AS}$, $\overline{RD}$, $\overline{WR}$) changes, use of these pins as output signals during reset must be disabled outside the microcomputer.

Notes: 1. The mode pin and FWE pin input must satisfy the mode programming setup time (t_{MDS}) relative to the reset clear timing.

2. For notes on FWE pin High/Low, see section 15.9, Notes on Flash Memory Programming/Erasing.

3. See section 4.2.2, Reset Sequence and 15.9, Notes on Flash Memory Programming/Erasing. With the mask ROM version of the H8/3039, H8/3038, H8/3037, and H8/3036, the minimum reset period during operation is 10 system clocks. However, the flash memory versions of the H8/3039 requires a minimum of 20 system clocks.

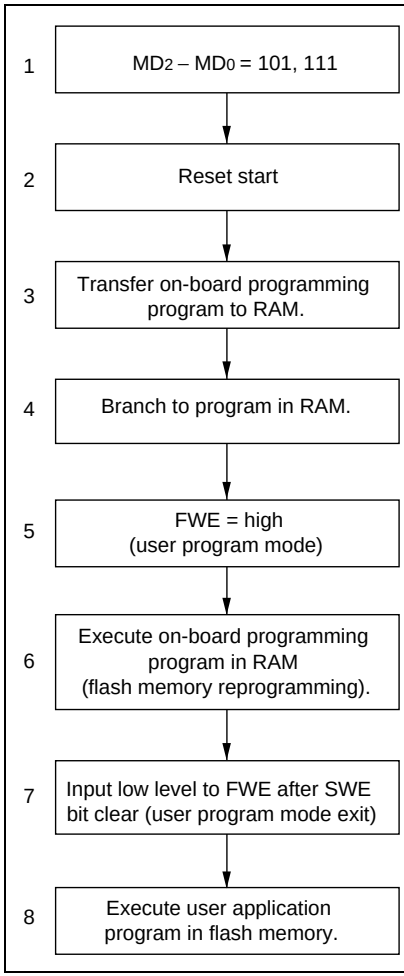
15.4.2 User Program Mode

When set to the user program mode, this LSI can erase and program its flash memory by executing a user program. Therefore, on-chip flash memory on-board programming can be performed by providing a means of controlling FWE and supplying the write data on the board and providing a write program in a part of the program area.

To select this mode, set the LSI to on-chip ROM enable modes 5 and 7 and apply a high level to the FWE pin. In this mode, the peripheral functions, other than flash memory, are performed the same as in modes 5 and 7.

Since the flash memory cannot be read while it is being programmed/erased, place a programming program on external memory, or transfer the programming program to RAM area, and execute it in the RAM. In mode 6, do not program/erase the flash memory. When setting mode 6, always input low level to the FWE pin.

Figure 15.9 shows the procedure for executing when transferred to on-chip RAM. During reset start, starting from the user program mode is possible.



<Procedure>
The user writes a program that executes steps 3 to 8 in advance as shown below .

1. Sets the mode pin to an on-chip ROM enable mode (mode 5 or 7).
2. Starts the CPU via reset. (The CPU can also be started from the user program mode by setting the FWE pin to High level during reset; that is, during the period the $\overline{\text{RES}}$ pin is a low level.)
3. Transfers the on-board programming program to RAM.
4. Branches to the program in RAM.
5. Sets the FWE pin to a high level.* (Switches to user program mode.)
6. After confirming that the FWE pin is a high level, executes the on-board programming program in RAM. This reprograms the user application program in flash memory.
7. At the end of reprograming, clears the SWE bit, and exits the user program mode by switching the FWE pin from a high level to a low level.*
8. Branches to, and executes, the user application program reprogrammed in flash memory.

Note: * For notes on FWE pin High/Low, see section 15.9, Notes on Flash Memory Programming/Erasing.

Figure 15.9 User Program Mode Execution Procedure (Example)

Note: Normally do not apply a high level to the FWE pin. To prevent erroneous programming or erasing in the event of program runaway, etc., apply a high level to the FWE pin only when programming/erasing flash memory (including flash memory emulation by RAM). If program runaway, etc. causes overprogramming or overerasing of flash memory, the memory cells will not operate normally.
Also, while a high level is applied to the FWE pin, the watchdog timer should be activated to prevent overprogramming or overerasing due to program runaway, etc.
In mode 6, do not reprogram flash memory. When setting mode 6, always set the FWE pin to a low level.

15.5 Programming/Erasing Flash Memory

A software method, using the CPU, is employed to program and erase flash memory in the on-board programming modes. There are four flash memory operating modes: program mode, erase mode, program-verify mode, and erase-verify mode. Transitions to these modes can be made by setting the PSU, ESU, P, E, PV, and EV bits in FLMCR.

For a description of state transition by FLMCR bit setting, see figure 15.10.

The flash memory cannot be read while being programmed or erased. Therefore, the program that controls flash memory programming/erasing (the programming control program) should be located and executed in on-chip RAM or external memory.

For the programming/erasing notes, see section 15.9, Notes on Flash Memory Programming/Erasing. For the wait time after each bit in FLMCR is set or cleared, see section 18.2.5, Flash Memory Characteristics.

Notes: 1. Operation is not guaranteed if setting/resetting of the SWE, ESU, PSU, EV, PV, E, and P bits in FLMCR is executed by a program in flash memory.
2. When programming or erasing, set the FWE pin input level to the high level, and set FWE to 1. (programming/erasing will not be executed if FWE = 0).

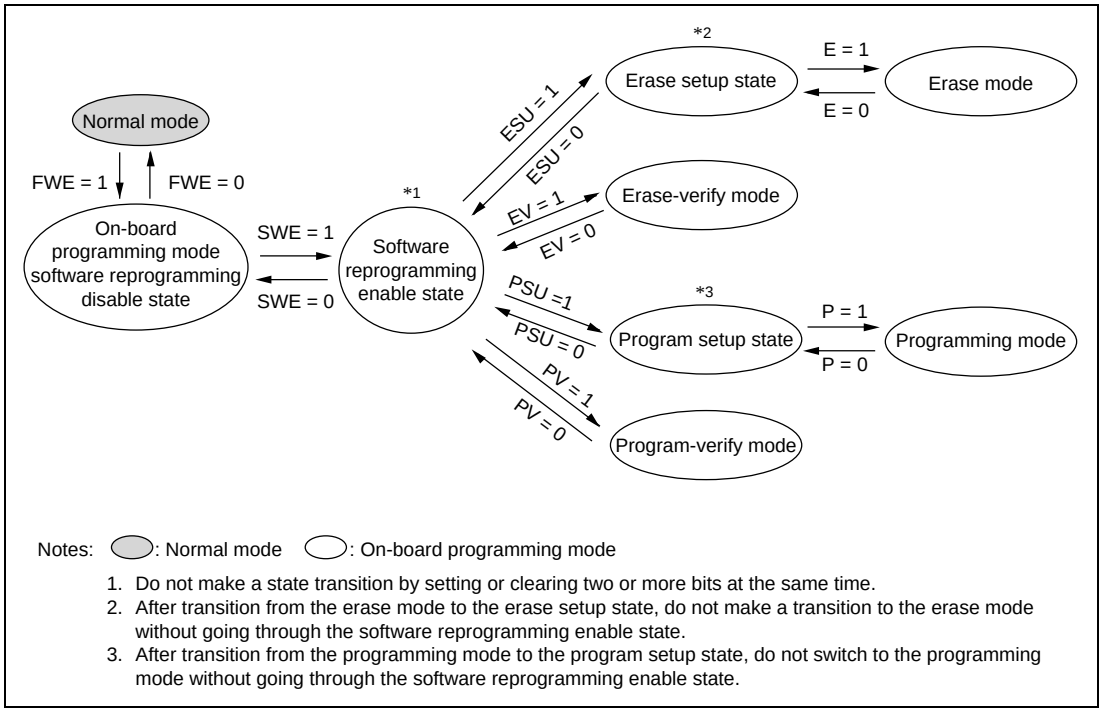


Figure 15.10 State Transition by Setting of Each Bit of FLMCR

15.5.1 Program Mode

Follow the procedure shown in the program/program-verify flowchart in figure 15.11 to write data or programs to flash memory. Performing program operations according to this flowchart will enable data or programs to be written to flash memory without subjecting the device to voltage stress or sacrificing program data reliability. Programming should be carried out 32 bytes at a time.

For the wait time (x, y, z, α, β, γ, ε, η) after setting or clearing each bit in the flash memory control register (FLMCR) and the maximum programming count (N), see table 18.15.

Following the elapse of (x) µs or more after the SWE bit is set to 1 in flash memory control register (FLMCR), 32-byte program data is stored in the program data area and reprogram data area, and the 32-byte data in the reprogram data area written consecutively to the write addresses. (The lower 8 bits of the first address written to must be H'00, H'20, H'40, H'60, H'80, H'A0, H'C0, or H'E0.) 32 consecutive byte data transfers are performed. The program address and program data are latched in the flash memory. A 32-byte data transfer must be performed even if writing fewer than 32 bytes; in this case, H'FF data must be written to the extra addresses.

Next, the watchdog timer (WDT) is set to prevent overprogramming due to program runaway, etc. Set a value greater than (y + z + α + ß) µs as the WDT overflow period. Preparation for entering program mode (program setup) is performed next by setting the PSU bit in FLMCR. The operating mode is then switched to program mode by setting the P bit in FLMCR after the elapse of at least (y) µs.

The time while the P bit is set is the flash memory programming time. Make a program setting so that the time for one programming operation is within the range of (z) µs.

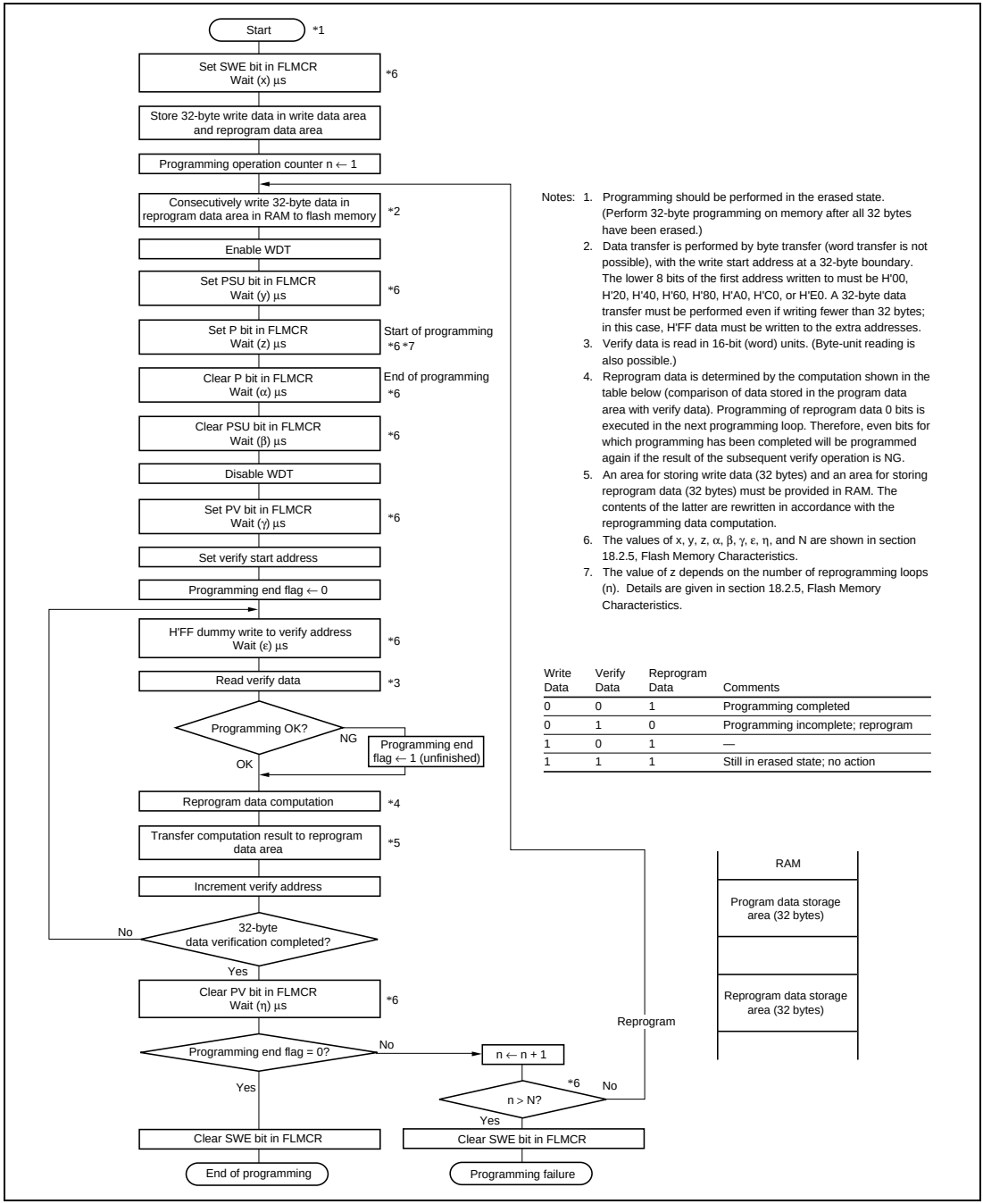
The wait time after P bit setting must be changed according to the number of reprogramming loops. For details, see section 18.2.5, Flash Memory Characteristics.

15.5.2 Program-Verify Mode

In program-verify mode, the data written in program mode is read to check whether it has been correctly written in the flash memory.

Clear the P bit in FLMCR, then wait for at least (α) µs before clearing the PSU bit to exit program mode. After exiting program mode, the watchdog timer setting is also cleared. Then the operating mode is switched to program-verify mode by setting the PV bit in FLMCR. Before reading in program-verify mode, a dummy write of H'FF data should be made to the addresses to be read. The dummy write should be executed after the elapse of (γ) µs or more. When the flash memory is read in this state (verify data is read in 16-bit units), the data at the latched address is read. Wait at least (ε) µs after the dummy write before performing this read operation. Next, the originally written data is compared with the verify data, and reprogram data is computed (see figure 15.11) and transferred to RAM. After verification of 32 bytes of data has been completed, exit program-verify mode, wait for at least (η) µs, then determine whether 32-byte programming has finished. If reprogramming is necessary, set program mode again, and repeat the program/program-verify sequence as before. However, ensure that the program/program-verify sequence is not repeated more than (N) times on the same bits.

Note: A 32-byte area to store program data and a 32-byte area to store reprogram data are required in RAM.



Notes:
1. Programming should be performed in the erased state. (Perform 32-byte programming on memory after all 32 bytes have been erased.)
2. Data transfer is performed by byte transfer (word transfer is not possible), with the write start address at a 32-byte boundary. The lower 8 bits of the first address written to must be H'00, H'20, H'40, H'60, H'80, H'A0, H'C0, or H'E0. A 32-byte data transfer must be performed even if writing fewer than 32 bytes; in this case, H'FF data must be written to the extra addresses.
3. Verify data is read in 16-bit (word) units. (Byte-unit reading is also possible.)
4. Reprogram data is determined by the computation shown in the table below (comparison of data stored in the program data area with verify data). Programming of reprogram data 0 bits is executed in the next programming loop. Therefore, even bits for which programming has been completed will be programmed again if the result of the subsequent verify operation is NG.
5. An area for storing write data (32 bytes) and an area for storing reprogram data (32 bytes) must be provided in RAM. The contents of the latter are rewritten in accordance with the reprogramming data computation.
6. The values of x, y, z, α, β, γ, ε, η, and N are shown in section 18.2.5, Flash Memory Characteristics.
7. The value of z depends on the number of reprogramming loops (n). Details are given in section 18.2.5, Flash Memory Characteristics.

Write Data	Verify Data	Reprogram Data	Comments
0	0	1	Programming completed
0	1	0	Programming incomplete; reprogram
1	0	1	—
1	1	1	Still in erased state; no action

Figure 15.11 Program/Program-Verify Flowchart

15.5.3 Erase Mode

Flash memory erasing should be performed block by block following the procedure shown in the erase/erase-verify flowchart (single-block erase) shown in figure 15.12.

For the wait time (x, y, z, α, β, γ, ε, η) after setting or clearing of each bit in the flash memory control register (FLMCR) and the maximum erase count (N), see table 18.15.

To erase the contents of flash memory, make a 1 bit setting for the flash memory area to be erased in erase block register (EBR) at least (x) µs after setting the SWE bit to 1 in FLMCR. Next, the watchdog timer (WDT) is set to prevent overerasing due to program runaway, etc. Set a value greater than (z) ms + (y + α + ß) µs as the WDT overflow period. Preparation for entering erase mode (erase setup) is performed next by setting the ESU bit in FLMCR. The operating mode is then switched to erase mode by setting the E bit in FLMCR after the elapse of at least (y) µs.

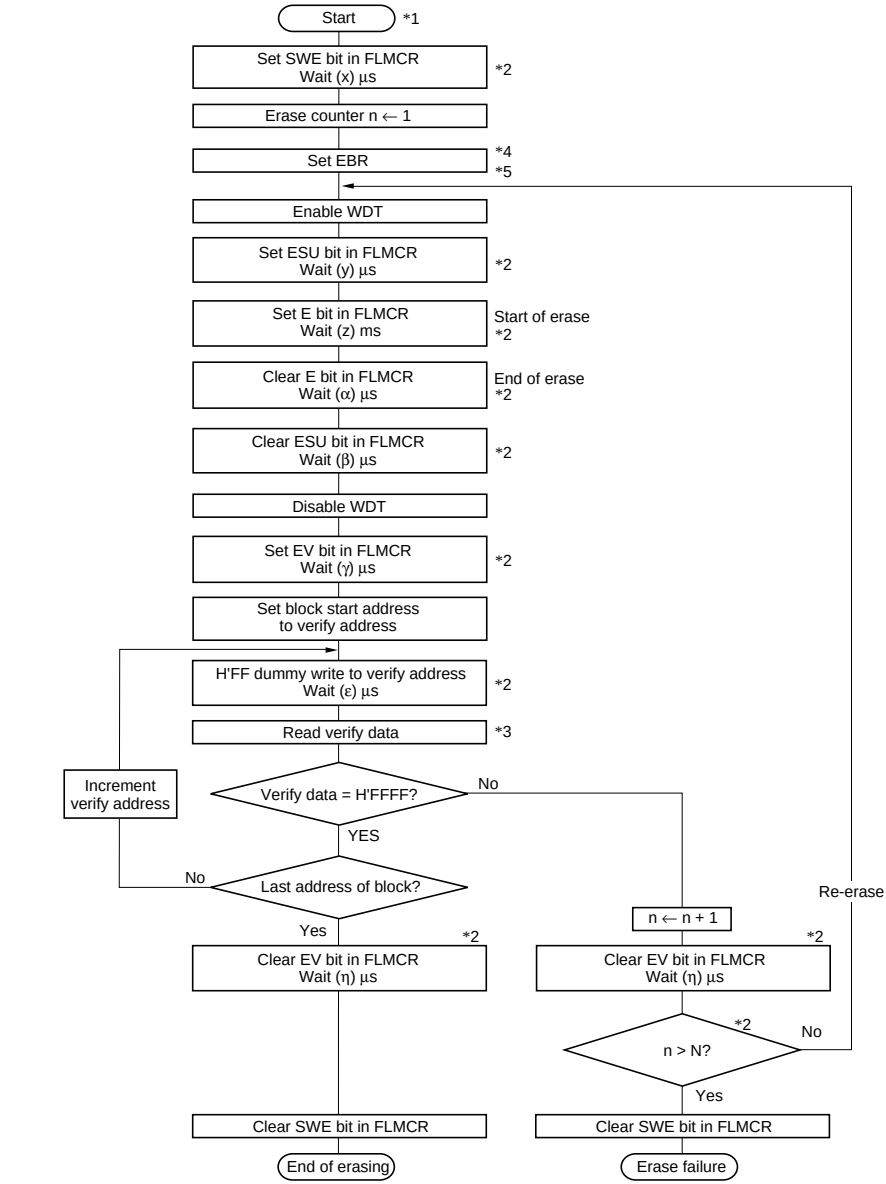
The time during which the E bit is set is the flash memory erase time. Ensure that the erase time does not exceed (z) ms.

Note: With flash memory erasing, preprogramming (setting all data in the memory to be erased to "0") is not necessary before starting the erase procedure.

15.5.4 Erase-Verify Mode

In erase-verify mode, data is read after memory has been erased to check whether it has been correctly erased.

After the elapse of the fixed erase time, clear the E bit in FLMCR, then wait for at least (α) µs before clearing the ESU bit to exit erase mode. After exiting erase mode, the watchdog timer setting is also cleared. The operating mode is then switched to erase-verify mode by setting the EV bit in FLMCR. Before reading in erase-verify mode, a dummy write of H'FF data should be made to the addresses to be read. The dummy write should be executed after the elapse of (γ) µs or more. When the flash memory is read in this state (verify data is read in 16-bit units), the data at the latched address is read. Wait at least (ε) µs after the dummy write before performing this read operation. If the read data has been erased (all "1"), a dummy write is performed to the next address, and erase-verify is performed. If the read data is unerased, set erase mode again, and repeat the erase/erase-verify sequence as before. However, do not repeat the erase/erase-verify sequence more than (N) times.



Notes: 1. Preprogramming (setting erase block data to all 0s) is not necessary.
2. The values of x, y, z, α, β, γ, ε, η, and N are shown in section 18.2.5, Flash Memory Characteristics.
3. Verify data is read in 16-bit (word) units. (Byte-unit reading is also possible.)
4. Set only one bit in EBR two or more bits must not be set simultaneously.
5. Erasing is performed in block units. To erase multiple blocks, each block must be erased in turn.

Figure 15.12 Erase/Erase-Verify Flowchart (Single-Block Erasing)

15.6 Flash Memory Protection

There are three kinds of flash memory program/erase protection: hardware protection, software protection, and error protection.

15.6.1 Hardware Protection

Hardware protection refers to a state in which programming/erasing of flash memory is forcibly disabled or aborted. Hardware protection is reset by settings in the flash memory control register (FLMCR) and erase block register (EBR). In the case of error protection, the P bit and E bit can be set, but a transition is not made to program mode or erase mode. (See table 15.8.)

Table 15.8 Hardware Protection

Item	Description	Function		
		Program	Erase	Verify*[1]
FWE pin protection	• When a low level is input to the FWE pin, FLMCR and EBR are initialized, and the program/erase-protected state is entered.*[4]	No*[2]	No*[3]	No
Reset/standby protection	• In a reset (including a WDT overflow reset) and in standby mode, FLMCR 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 (The minimum oscillation stabilization time is 20ms). In the case of a reset during operation, hold the $\overline{\text{RES}}$ pin low for at least 20 system clock cycles.*[5]	No	No*[3]	No
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 FLMCR 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]	Yes

Notes: 1. Two modes: program-verify and erase-verify.
2. The RAM area that overlapped flash memory is deleted.
3. All blocks become unerasable and specification by block is impossible.
4. For more information, see section 15.9, Notes on Flash Memory Programming/Erasing.
5. See sections 4.2.2, Reset Sequence and 15.9, Notes on Flash Memory Programming/Erasing. This LSI requires a minimum reset time during operation of 20 system clocks.

15.6.2 Software Protection

Software protection can be implemented by setting the RAMS bit in RAM control register (RAMCR) and erase block register (EBR). When software protection is in effect, setting the P or E bit in flash memory control register (FLMCR) does not cause a transition to program mode or erase mode. (See table 15.9.)

Table 15.9 Software Protection

		Function		
Item	**Description**	**Program**	**Erase**	**Verify**[1]
Emulation protection[2]	Setting the RAMS bit in RAMCR sets the program/erase-protected state for all blocks.	No[2]	No[3]	Yes
Block specification protection	Erase protection can be set for individual blocks by settings in erase block register (EBR).[4] However, program protection is disabled. Setting EBR to H'00 places all blocks in the erase-protected state.	—	No	Yes

Notes: 1. Two modes: program-verify mode and erase-verify mode.
2. Programming to the RAM area that overlaps flash memory is possible.
3. All blocks become unerasable, and specification by block is impossible.
4. Set H'00 in the EBR bits, except for erase.

15.6.3 Error Protection

In error protection, an error is detected when this LSI runaway occurs during flash memory programming/erasing*[1], or operation is not performed in accordance with the program/erase algorithm, and the program/erase operation is aborted. Aborting the program/erase operation prevents damage to the flash memory due to overprogramming or overerasing.

If the LSI malfunctions during flash memory programming/erasing, the FLER bit*[2] is set to 1 in flash memory status register (FLMSR) and the error protection state is entered. The FLMCR and EBR settings*[3] are retained, but program mode or erase mode is aborted at the point at which the error occurred. When 1 is set in the FLER bit, transition to the program mode or erase mode cannot be made even by setting the P and E bits in FLMCR. However, PV and EV bit in FLMCR setting is enabled, and a transition can be made to verify mode.

Error protection is released only by a reset via the $\overline{\text{RES}}$ pin or a WDT reset, or in the hardware standby mode.

Figure 15.13 shows the flash memory state transition diagram.

Notes: 1. This is the state in which the P or E bit in FLMCR is set to 1. In this state, NMI input is disabled. For more information, see section 15.6.4, NMI Input Disable Conditions.
2. For a detailed description of the FLER bits setting conditions, see section 15.3.4, Flash Memory Status Register (FLMSR).
3. Data can be written to FLMCR and EBR. However, when transition to the software standby mode was made in the error protection state, the registers are initialized.

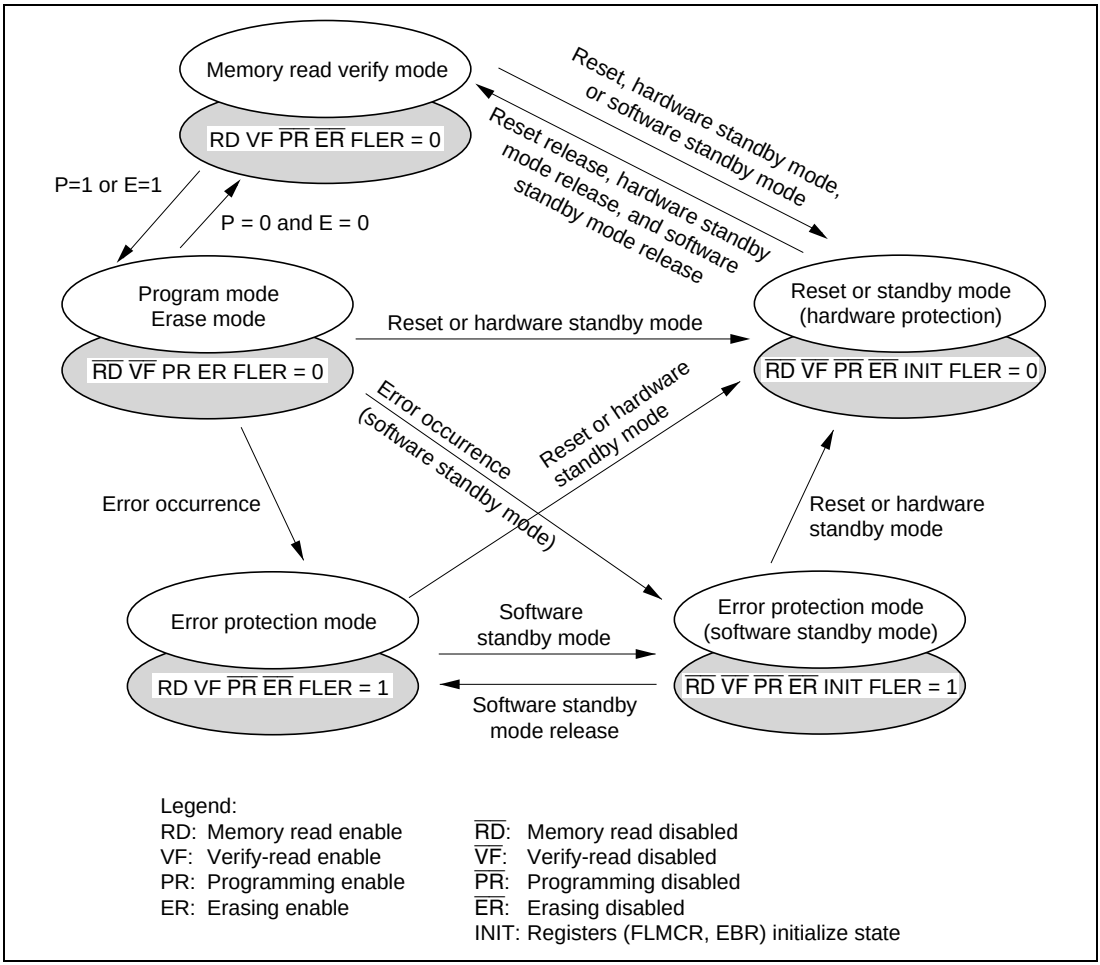


Figure 15.13 Flash Memory State Transitions
(When High Level Apply to FWE Pin in Modes 5 and 7 (On-Chip ROM Enabled))

The error protection function is disabled for errors other than the FLER bit set conditions. If considerable time elapses up to transit to this protection state, the flash memory may already be damaged. As a result, this function cannot completely protect the flash memory against damage.

Therefore, to prevent such erroneous operation, operation must be carried out correctly in according with the program/erase algorithms in the state that flash write enable (FWE) is set. In addition, the operation must be always carried out correctly by supervising microcomputer errors inside and outside the chip with the watchdog timer, etc. At transition to this protection mode, the flash memory may be erroneously programmed or erased, or its abort may result in incomplete

programming and erasing. In such cases, always forcibly return (reprogram) by boot mode. However, overprogramming and overerasing may prevent the boot mode from starting normally.

15.6.4 NMI Input Disable Conditions

While flash memory is being programmed/erased and the boot program is executing in the boot mode (however, period up to branching to on-chip RAM area)*[1], NMI input is disabled because the programming/erasing operations have priority.

This is done to avoid the following operation states:

1. Generation of an NMI input during programming/erasing violates the program/erase algorithms and normal operation can not longer be assured.
2. Vector-read cannot be carried out normally*[2] during NMI exception handling during programming/erasing and the microcomputer runs away as a result.
3. If an NMI input is generated during boot program execution, the normal boot mode sequence cannot be executed.

Therefore, this LSI has conditions that exceptionally disable NMI inputs only in the on-board programming mode. However, this does not assure normal programming/erasing and microcomputer operation.

Thus, in the FWE application state, all requests, including NMI, inside and outside the microcomputer, exception handling, and bus release must be restricted. NMI inputs are also disabled in the error protection state and the state that holds the P or E bit in FLMCR during flash memory emulation by RAM.

Notes: 1. Indicates the period up to branching to the on-chip RAM boot program area (H'FEF10 to H'FF2FF). (This branch occurs immediately after user program transfer was completed.)

Therefore, after branching to RAM area, NMI input is enabled in states other than the program/erase state. Thus, interrupt requests inside and outside the microcomputer must be disabled until initial writing by user program (writing of vector table and NMI processing program, etc.) is completed.

2. In this case, vector read is not performed normally for the following two reasons:
   a. The correct value cannot be read even by reading the flash memory during programming/erasing. (Value is undefined.)
   b. If a value has not yet been written to the NMI vector table, NMI exception handling will not be performed correctly.

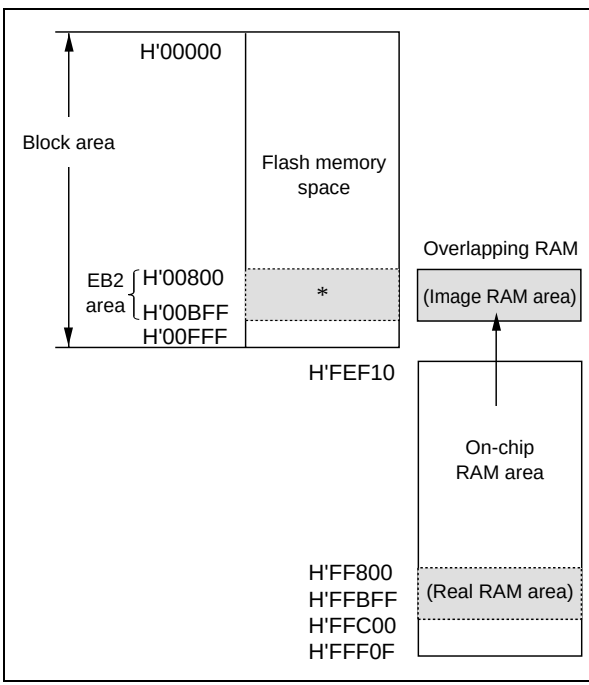
15.7 Flash Memory Emulation by RAM

Erasing and programming the flash memory takes time, which can make it difficult to tune parameters and other data in real time. In this case, overlapping part (H'FF800 to H'FFBFF) of RAM onto a small block area of flash memory can be performed to emulate real-time reprogramming of flash memory. This RAM reassignment is performed using bits 3 to 1 in the RAM control register (RAMCR).

After the RAM area change, two areas can be accessed: the overlapped flash memory area and the original RAM area (H'FF800 to H'FFBFF). For a description of the RAMCR and RAM area setting procedure, see section 15.3.3, RAM Control Register (RAMCR).

Example of real-time emulation of flash memory

An example of RAM area H'FF800 to H'FFBFF overlapping EB2 (H'00800 to H'00BFF) flash memory area is shown below.



Procedure:

1. Part (H'FF800 to H'FFBFF) of RAM overlaps the area (EB2) needed to carry out real-time reprogramming. (Bits 3 to 1 in the RAMCR are set to 1, 1, 0 and the overlap flash memory area (EB2) is selected.)
2. Real-time reprogramming is carried out using the overlapping RAM.
3. After the reprogramming data is verified, RAM overlapping is released. (RAMS bits are cleared.)
4. The data written to H'FF800 to H'FFBFF in RAM are written to flash memory space.

Note: * When part (H'FF800 to H'FFBFF) of RAM overlapped a small block area of flash memory, the overlapped flash memory area cannot be accessed. This area can be accessed by releasing overlapping.

Figure 15.14 Example of RAM Overlapping Operation

Notes on use of the RAM emulation function

(1) Notes on flash write enable (FWE) high/low

Care is necessary to prevent erroneous programming/erasing at FWE = high/low, the same as in the on-board programming mode. To prevent erroneous programming and erasing due to program runaway, etc., during FWE application, in particular, the watchdog timer should be set when the P, or E bit is set to 1 in FLMCR, even while the emulation function is being used. For more information, see section 15.9, Notes on Flash Memory Programming/Erasing.

(2) NMI input disable conditions

When the P and E bits in FLMCR are set, NMI input is disabled, the same as normal program/erase even when using the emulation function.

NMI input is cleared when the P and E bits are reset (including watchdog timer reset), in the standby mode, when a high level is not applied to FWE, and when the SWE bit in FLMCR is 0 in state in which a high level is input to FWE.

15.8 Flash Memory PROM Mode

15.8.1 PROM Mode Setting

This LSI has a PROM mode, besides an on-board programming mode, as a flash memory program/erase mode. In the PROM mode, a program can be freely written to the on-chip ROM using a PROM programmer that supports the Renesas Technology 128 kbytes flash memory on-chip microcomputer device type.

For notes on PROM mode use, see sections 15.8.9, Notes on Memory Programming and 15.9, Notes on Flash Memory Programming/Erasing.

15.8.2 Memory Map

Figure 15.15 shows the PROM mode memory map.

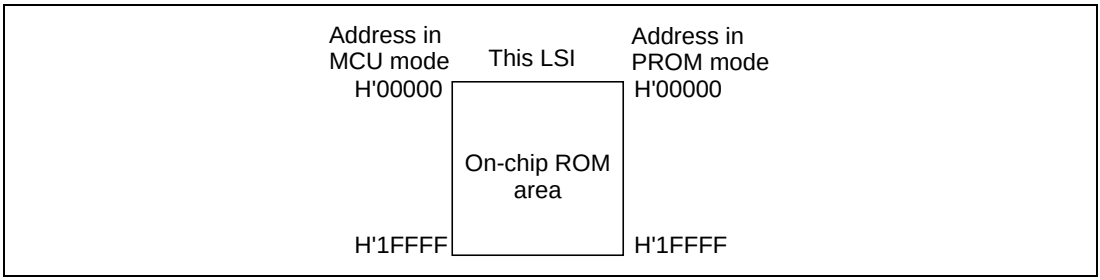


Figure 15.15 PROM Mode Memory Map

15.8.3 PROM Mode Operation

Table 15.10 shows how the different operating modes are set when using PROM mode, and table 15.11 lists the commands used in PROM mode. Details of each mode are given below.

- Memory Read Mode

  Memory read mode supports byte reads.
- Auto-Program Mode

  Auto-program mode supports programming of 128 bytes at a time. Status polling is used to confirm the end of auto-programming.
- Auto-Erase Mode

  Auto-erase mode supports automatic erasing of the entire flash memory. Status polling is used to confirm the end of auto-erasing.
- Status Read Mode

  Status polling is used for auto-programming and auto-erasing, and normal termination can be confirmed by reading the I/O 6 signal. In status read mode, error information is output if an error occurs.

Table 15.10 Settings for Each Operating Mode in PROM Mode

	Pin Names*3					
Mode	**FWE**	**$\overline{CE}$**	**$\overline{OE}$**	**$\overline{WE}$**	**D_0 to D_7**	**A_0 to A_{17}**
Read	V_{CC} or 0	L	L	H	Data output	Ain
Output disable	V_{CC} or 0	L	H	H	Hi-Z	X
Command write	V_{CC} or 0	L	H	L	Data input	Ain*2
Chip disable*1	V_{CC} or 0	H	X	X	Hi-Z	X

Legend:

L: Low level

H: High level

X: Undefined

Hi-Z: High impedance

Notes: For command writes when making a transition to auto-program or auto-erase mode, input Vcc (V) to FWE.

1. Chip disable is not a standby state; internally, it is an operation state.
2. Ain indicates that there is also address input in auto-program mode.
3. The pin names are those assigned in this LSI PROM mode.

Table 15.11 PROM Mode Commands

		1st Cycle			2nd Cycle		
Command Name	**Number of Cycles**	**Mode**	**Address**	**Data**	**Mode**	**Address**	**Data**
Memory read mode	1	Write	X	H'00	Read	RA	Dout
Auto-program mode	129	Write	X	H'40	Write	WA	Din
Auto-erase mode	2	Write	X	H'20	Write	X	H'20
Status read mode	2	Write	X	H'71	Write	X	H'71

Legend:

RA: Read address

WA: Program address

Dout: Read data

Din: Program data

Note: In auto-program mode, 129 cycles are required for command writing by a simultaneous 128-byte write.

Table 15.12 DC Characteristics in Memory Read Mode

Conditions: V_{CC} = 5.0 V ±10%, V_{SS} = 0 V, T_a = 25°C ±5°C

Item		Symbol	Min	Typ	Max	Unit	Test Conditions
Input high voltage	0_7–0_0, A_{16}–A_0	V_{IH}	2.2	—	Vcc +0.3	V	
Input low voltage	0_7–0_0, A_{16}–A_0	V_{IL}	0.3	—	0.8	V	
Schmitt trigger input voltage	$\overline{OE}$, $\overline{CE}$, $\overline{WE}$	V_T^-	1.0	—	2.5	V	
		V_T^+	2.0	—	3.5	V	
		$V_T^+ - V_T^-$	0.4	—	—	V	
Output high voltage	0_7–0_0	V_{OH}	2.4	—	—	V	I_{OH} = –200 µA
Output low voltage	0_7–0_0	V_{OL}	—	—	0.45	V	I_{OL} = 1.6 mA
Input leakage current	0_7–0_0, A_{16}–A_0	$\lvert I_{LI} \rvert$	—	—	2	µA	
V_{CC} current	Reading	I_{cc}	—	40	65	mA	
	Programming	I_{cc}	—	50	85	mA	
	Erasing	I_{cc}	—	50	85	mA	

Note: For the electrical characteristics of the flash memory version, see section 18.2.1, Absolute Maximum Ratings.

Exceeding the absolute maximum ratings may cause permanent damage to the chip.

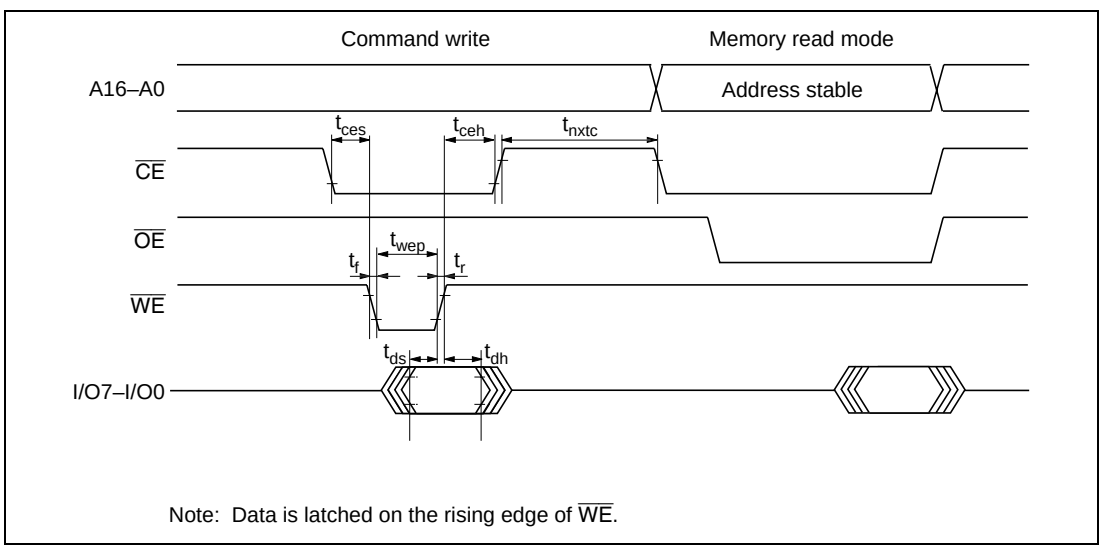
15.8.4 Memory Read Mode

AC Characteristics

Table 15.13 AC Characteristics in Memory Read Mode Transition

Conditions: V_{CC} = 5.0 V ±10%, V_{SS} = 0 V, T_a = 25°C ±5°C

Item	Symbol	Min	Max	Unit	Notes
Command write cycle	t_{nxtc}	20	—	µs	
$\overline{CE}$ hold time	t_{ceh}	0	—	ns	
$\overline{CE}$ setup time	t_{ces}	0	—	ns	
Data hold time	t_{dh}	50	—	ns	
Data setup time	t_{ds}	50	—	ns	
Write pulse width	t_{wep}	70	—	ns	
$\overline{WE}$ rise time	t_r	—	30	ns	
$\overline{WE}$ fall time	t_f	—	30	ns	



Note: Data is latched on the rising edge of $\overline{WE}$.

Figure 15.16 Timing Waveform in Memory Read Mode Transition

Table 15.14 AC Characteristics in Memory Contents Read

Conditions: V_{CC} = 5.0 V ±10%, V_{SS} = 0 V, T_a = 25°C ±5°C

Item	Symbol	Min	Max	Unit	Notes
Access time	t_{acc}	—	20	µs	
$\overline{CE}$ output delay time	t_{ce}	—	150	ns	
$\overline{OE}$ output delay time	t_{oe}	—	150	ns	
Output disable delay time	t_{df}	—	100	ns	
Data output hold time	t_{oh}	5	—	ns	

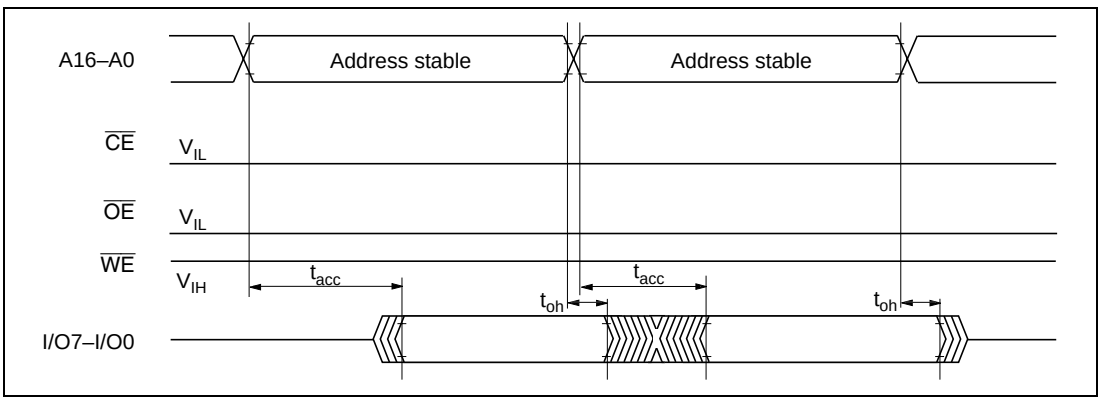


Figure 15.17 $\overline{CE}$/$\overline{OE}$ Enable State Read

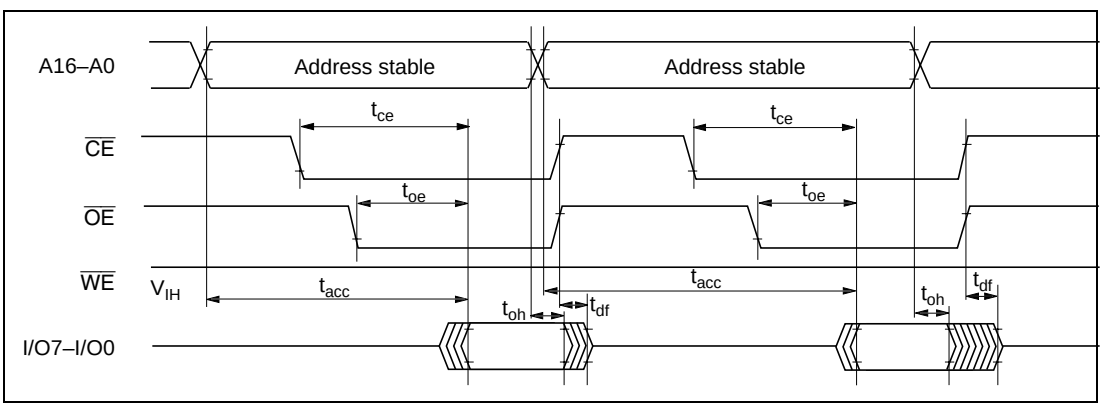
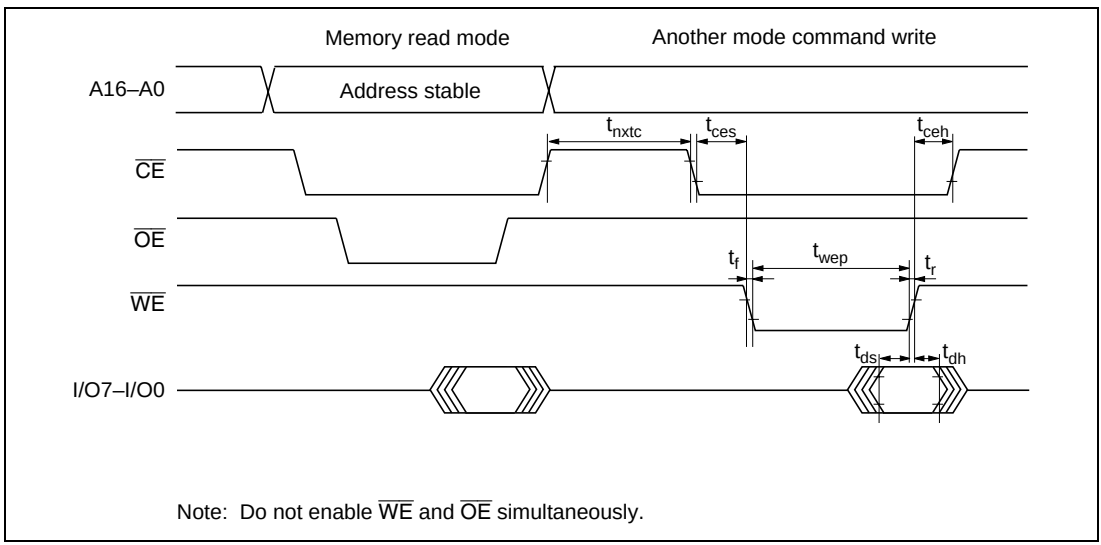


Figure 15.18 $\overline{CE}$/$\overline{OE}$ Clock Read

Table 15.15 AC Characteristics in Transition from Memory Read Mode to Another Mode

Conditions: V_{CC} = 5.0 V ±10%, V_{SS} = 0 V, T_a = 25°C ±5°C

Item	Symbol	Min	Max	Unit	Notes
Command write cycle	t_{nxtc}	20	—	µs	
$\overline{CE}$ hold time	t_{ceh}	0	—	ns	
$\overline{CE}$ setup time	t_{ces}	0	—	ns	
Data hold time	t_{dh}	50	—	ns	
Data setup time	t_{ds}	50	—	ns	
Write pulse width	t_{wep}	70	—	ns	
$\overline{WE}$ rise time	t_r	—	30	ns	
$\overline{WE}$ fall time	t_f	—	30	ns	



Note: Do not enable $\overline{WE}$ and $\overline{OE}$ simultaneously.

Figure 15.19 Transition From Memory Read Mode to Another Mode

15.8.5 Auto-Program Mode

AC Characteristics

Table 15.16 AC Characteristics in Auto-Program Mode

Conditions: V_{CC} = 5.0 V ±10%, V_{SS} = 0 V, T_a = 25°C ±5°C

Item	Symbol	Min	Max	Unit	Notes
Command write cycle	t_{nxtc}	20	—	µs	
$\overline{CE}$ hold time	t_{ceh}	0	—	ns	
$\overline{CE}$ setup time	t_{ces}	0	—	ns	
Data hold time	t_{dh}	50	—	ns	
Data setup time	t_{ds}	50	—	ns	
Write pulse width	t_{wep}	70	—	ns	
Status polling start time	t_{wsts}	1	—	ms	
Status polling access time	t_{spa}	—	150	ns	
Address setup time	t_{as}	0	—	ns	
Address hold time	t_{ah}	60	—	ns	
Memory write time	t_{write}	1	3000	ms	
$\overline{WE}$ rise time	t_r	—	30	ns	
$\overline{WE}$ fall time	t_f	—	30	ns	
Write setup time	t_{pns}	100	—	ns	
Write end setup time	t_{pnh}	100	—	ns	

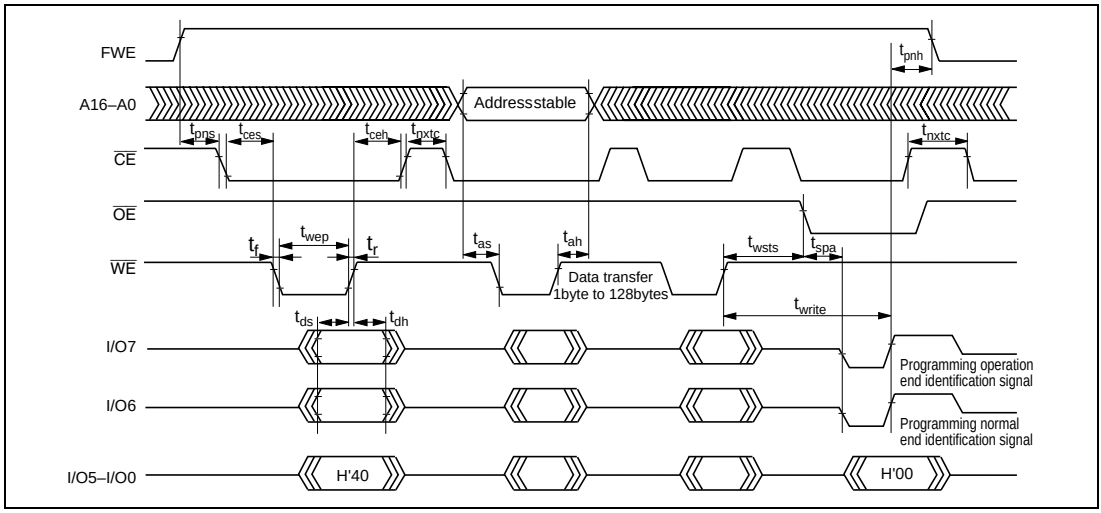


Figure 15.20 Auto-Program Mode Timing Waveforms

Cautions on Use of Auto-Program Mode

- In auto-program mode, 128 bytes are programmed simultaneously. This should be carried out by executing 128 consecutive byte transfers.
- A 128-byte data transfer is necessary even when programming fewer than 128 bytes. In this case, H'FF data must be written to the extra addresses.
- If a value other than an effective address is input, processing will switch to a memory write operation but a write error will be flagged.
- Memory address transfer is performed in the second cycle (figure 15.20). Do not perform transfer after the second cycle.
- Do not perform a command write during a programming operation.
- Perform one auto-programming operation for a 128-byte block for each address. Characteristics are not guaranteed for two or more programming operations.
- Confirm normal end of auto-programming by checking I/O 6. Alternatively, status read mode can also be used for this purpose.

15.8.6 Auto-Erase Mode

AC Characteristics

Table 15.17 AC Characteristics in Auto-Erase Mode

Conditions: V_{CC} = 5.0 V ±10%, V_{SS} = 0 V, T_a = 25°C ±5°C

Item	Symbol	Min	Max	Unit	Notes
Command write cycle	t_{nxtc}	20	—	µs	
$\overline{CE}$ hold time	t_{ceh}	0	—	ns	
$\overline{CE}$ setup time	t_{ces}	0	—	ns	
Data hold time	t_{dh}	50	—	ns	
Data setup time	t_{ds}	50	—	ns	
Write pulse width	t_{wep}	70	—	ns	
Status polling start time	t_{ests}	1	—	ms	
Status polling access time	t_{spa}	—	150	ns	
Memory erase time	t_{erase}	100	40000	ms	
$\overline{WE}$ rise time	t_r	—	30	ns	
$\overline{WE}$ fall time	t_f	—	30	ns	
Erase setup time	t_{ens}	100	—	ns	
Erase end setup time	t_{enh}	100	—	ns	

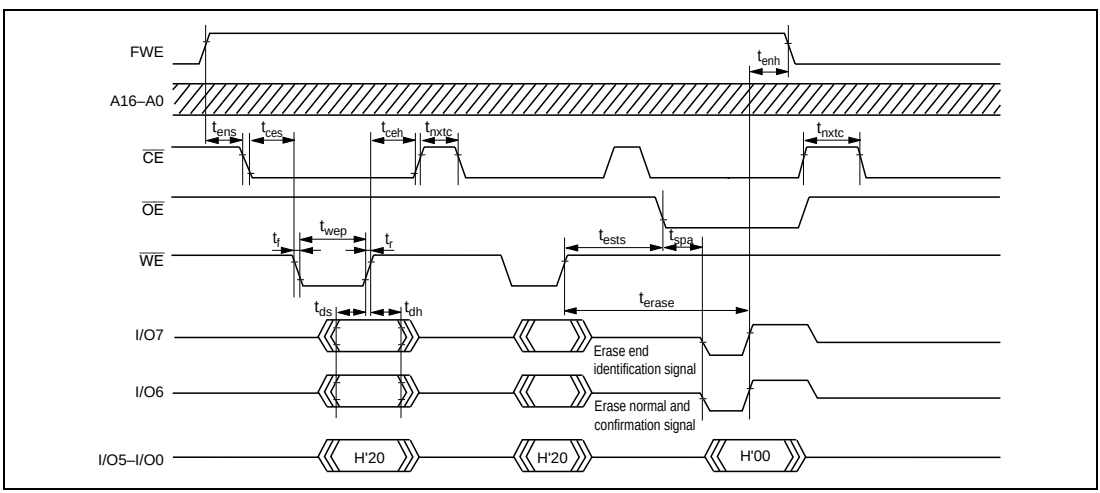


Figure 15.21 Auto-Erase Mode Timing Waveforms

Caution on Use of Erase-Program Mode

- Auto-erase mode supports only entire memory erasing.
- Do not perform a command write during auto-erasing.
- Confirm normal end of auto-erasing by checking I/O 6. Alternatively, status read mode can also be used for this purpose.

15.8.7 Status Read Mode

AC Characteristics

Table 15.18 AC Characteristics in Status Read Mode

Conditions: V_{CC} = 5.0 V ±10%, V_{SS} = 0 V, T_a = 25°C ±5°C

Item	Symbol	Min	Max	Unit	Notes
Command write cycle	t_{nxtc}	20	—	µs	
$\overline{CE}$ hold time	t_{ceh}	0	—	ns	
$\overline{CE}$ setup time	t_{ces}	0	—	ns	
Data hold time	t_{dh}	50	—	ns	
Data setup time	t_{ds}	50	—	ns	
Write pulse width	t_{wep}	70	—	ns	
$\overline{OE}$ output delay time	t_{oe}	—	150	ns	
Disable delay time	t_{df}	—	100	ns	
$\overline{CE}$ output delay time	t_{ce}	—	150	ns	
$\overline{WE}$ rise time	t_r	—	30	ns	
$\overline{WE}$ fall time	t_f	—	30	ns	

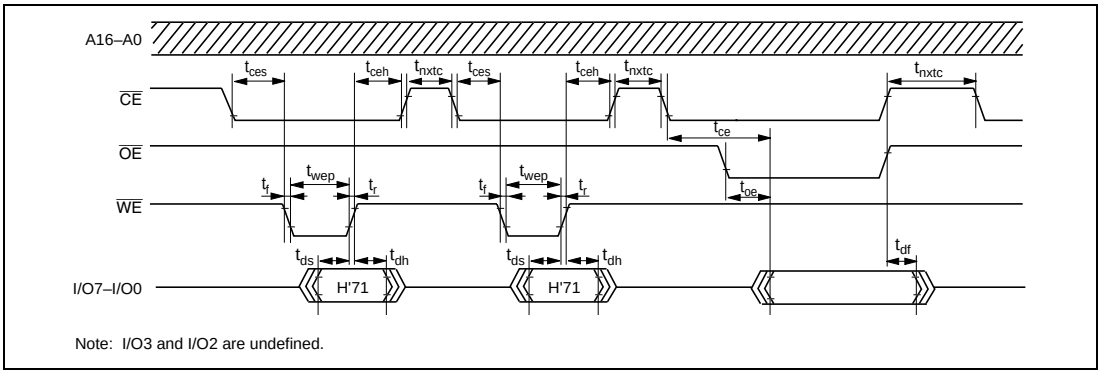


Figure 15.22 Status Read Mode Timing Waveforms

Table 15.19 Status Read Mode Return Commands

Pin Name	I/O7	I/O6	I/O5	I/O4	I/O3	I/O2	I/O1	I/O0
Attribute	Normal end identification	Command error	Program-ming error	Erase error	—	—	Program-ming or erase count exceeded	Effective address error
Initial value	0	0	0	0	0	0	0	0
Indications	Normal end: 0 Abnormal end: 1	Command error: 1 Otherwise: 0	Program-ming error: 1 Otherwise: 0	Erase error: 1 Otherwise: 0	—	—	Count exceeded: 1 Otherwise: 0	Effective address error: 1 Otherwise: 0

Notes on status read mode

After exiting auto-program mode or auto-erase mode, status read mode must be executed without dropping the power supply.

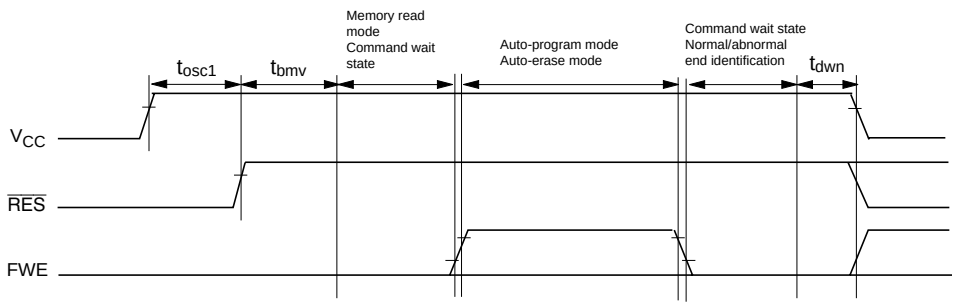
Immediately after powering on, or once powering off, the return command is undefined.

15.8.8 PROM Mode Transition Time

Commands cannot be accepted during the oscillation stabilization period or the PROM mode setup period. After the PROM mode setup time, a transition is made to memory read mode.

Table 15.20 Stipulated Transition Times to Command Wait State

Item	Symbol	Min	Max	Unit	Notes
Standby release (oscillation settling time)	t_{osc1}	20	—	ms	
PROM mode setup time	t_{bmv}	10	—	ms	
V_{CC} hold time	t_{dwn}	0	—	ms	



Note: Set the FWE input pin low level, except in the auto-program and auto-erase modes.

Figure 15.23 Oscillation Stabilization Time, Boot Program Transfer Time

15.8.9 Notes on Memory Programming

- When programming addresses which have previously been programmed, carry out auto-erasing before auto-programming (figure 15.24).
- When performing programming using PROM mode on a chip that has been programmed/erased in an on-board programming mode, auto-erasing is recommended before carrying out auto-programming.

Notes: 1. The flash memory is initially in the erased state when the device is shipped by Renesas. For other chips for which the erasure history is unknown, it is recommended that auto-erasing be executed to check and supplement the initialization (erase) level.

2. In the PROM mode, auto-programming to a 128-byte programming unit block should be performed only once.

   Do not perform additional programming to a programmed 128-byte programming unit block.

   To reprogram, perform auto-programming after auto-erasing.

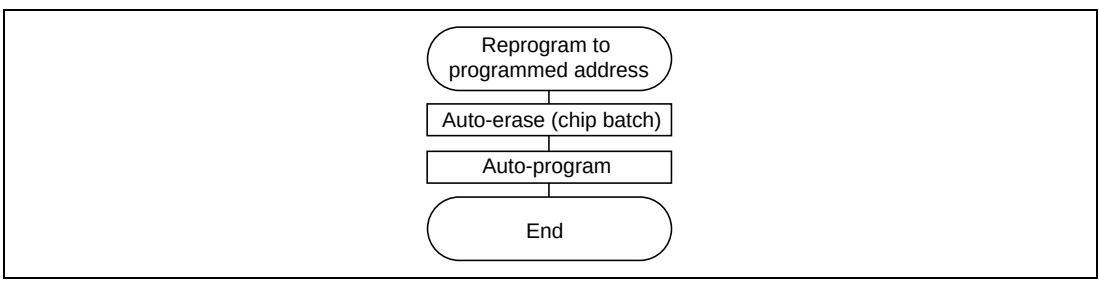


Figure 15.24 Reprogramming to Programmed Address

15.9 Notes on Flash Memory Programming/Erasing

The following describes notes when using the on-board programming mode, RAM emulation function, and PROM mode.

(1) Program/erase with the specified voltage and timing.

Applied voltages in excess of the rating can permanently damage the device.

Use a PROM writer that supports the Renesas Technology 128 kbytes flash memory on-board microcomputer device type.

Do not set the PROM writer at the HN28F101. If the PROM writer is set to the HN28F101 by mistake, a high level can be input to the FWE pin and the LSI can be destroyed.

(2) Notes on powering on/powering off (See figures 15.25 to 15.27.)

Input a high level to the FWE pin after verifying Vcc. Before turning off Vcc, set the FWE pin to a low level.

When powering on and powering off the Vcc power supply, fix the FWE pin a low level and set the flash memory to the hardware protection mode.

Be sure that the powering on and powering off timing is satisfied even when the power is turned off and back on in the event of a power interruption, etc. If this timing is not satisfied, microcomputer runaway, etc., may cause overprogramming or overerasing and the memory cells may not operate normally.

(3) Notes on FWE pin High/Low switching (See figures 15.25 to 15.27.)

Input FWE in the state microcomputer operation is verified. If the microcomputer does not satisfy the operation confirmation state, fix the FWE pin at a low level to set the protection mode.

To prevent erroneous programming/erasing of flash memory, note the following in FWE pin High/Low switching:

- Apply an input to the FWE pin after the Vcc voltage has stabilized within the rated voltage.

  If an input is applied to the FWE pin when the microcomputer Vcc voltage does not satisfy the rated voltage, flash memory may be erroneously programmed or erased because the microcomputer is in the unconfirmed state.
- Apply an input to the FWE pin when the oscillation has stabilized (after the oscillation stabilization time).

  When turning on the Vcc power, apply an input to the FWE pin after holding the $\overline{\text{RES}}$ pin at a low level during the oscillation stabilization time (t_{osc1}=20ms). Do not apply an input to the FWE pin when oscillation is stopped or unstable.
- In the boot mode, perform FWE pin High/Low switching during reset.

  In transition to the boot mode, input FWE = High level and set MD_2 to MD_0 while the $\overline{\text{RES}}$ input is low. At this time, the FWE and MD_2 to MD_0 inputs must satisfy the mode programming setup time (t_{MDS}) relative to the reset clear timing. The mode programming setup time is necessary for $\overline{\text{RES}}$ reset timing even in transition from the boot mode to another mode.

  In reset during operation, the $\overline{\text{RES}}$ pin must be held at a low level for at least 20 system clocks.
- In the user program mode, FWE = High/Low switching is possible regardless of the $\overline{\text{RES}}$ input.

  FWE input switching is also possible during program execution on flash memory.

- 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 supervised using a watchdog timer, etc.
- Input low level to the FWE pin when the SWE, ESU, PSU, EV, PV, E, and P bits in FLMCR have been cleared.

  Do not erroneously set the SWE, ESU, PSU, EV, PV, E, and P bits when FWE High/Low.

(4) Do not input a constant high level to the FWE pin.

To prevent erroneous programming/erasing in the event of program runaway, etc., input a high level to the FWE pin only when programming/erasing flash memory (including flash memory emulation by RAM). Avoid system configurations that constantly input a high level to the FWE pin. Handle program runaway, etc. by starting the watchdog timer so that flash memory 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 algorithms.

The recommended algorithms can program/erase the flash memory without applying voltage stress to the device or sacrificing the reliability of the program data.

When setting the PSU and ESU bits in FLMCR, set the watchdog timer for program runaway, etc.

(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 be accessed for purposes other than verify (verify during programming/erase).

Similarly perform flash memory program execution and data read after clearing the SWE bit even when using the RAM emulation function with a high level input to the FWE pin. However, RAM area that overlaps flash memory space can be read/programmed whether the SWE bit is set or cleared.

(7) Do not use an interrupt during flash memory programming or erasing.

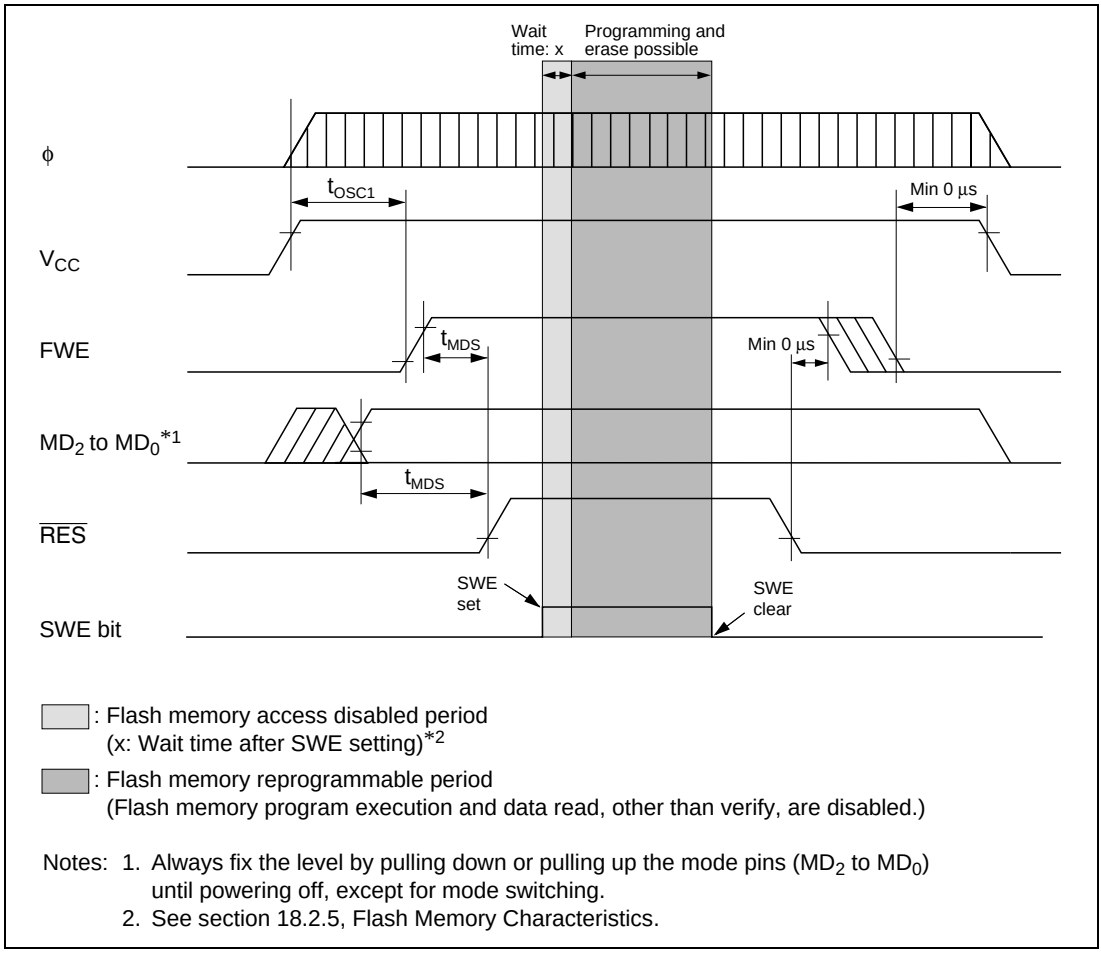
Since programming/erase operations (including emulation by RAM) have priority when a high level is input to the FWE pin, disable all interrupt requests, including NMI.

(8) Do not perform additional programming. Reprogram flash memory after erasing.

With on-board programming, program to 32-byte programming unit blocks one time only. Program to 128-byte programming unit blocks one time only even in the PROM mode. Erase all the programming unit blocks before reprogramming.
Bus release must also be disabled.

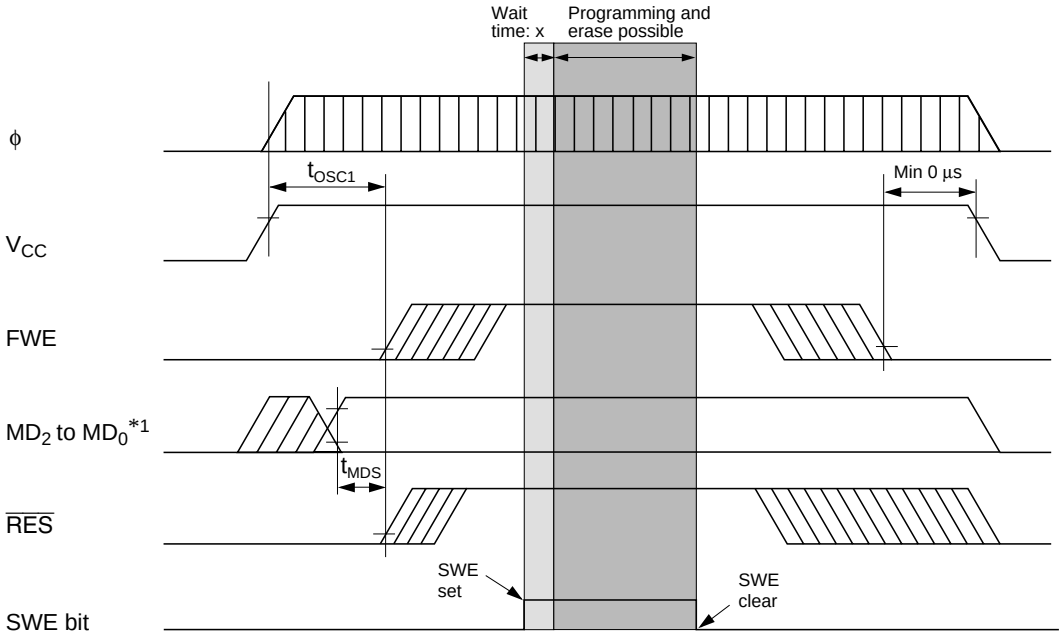
(9) Before programming, check that the chip is correctly mounted in the PROM programmer. Overcurrent damage to the device can result if the index marks on the PROM programmer socket, socket adapter, and chip are not correctly aligned.

(10) Do not touch the socket adapter or chip during programming. Touching either of these can cause contact faults and write errors.



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 18.2.5, Flash Memory Characteristics.

Figure 15.25 Powering On/Off Timing (Boot Mode)

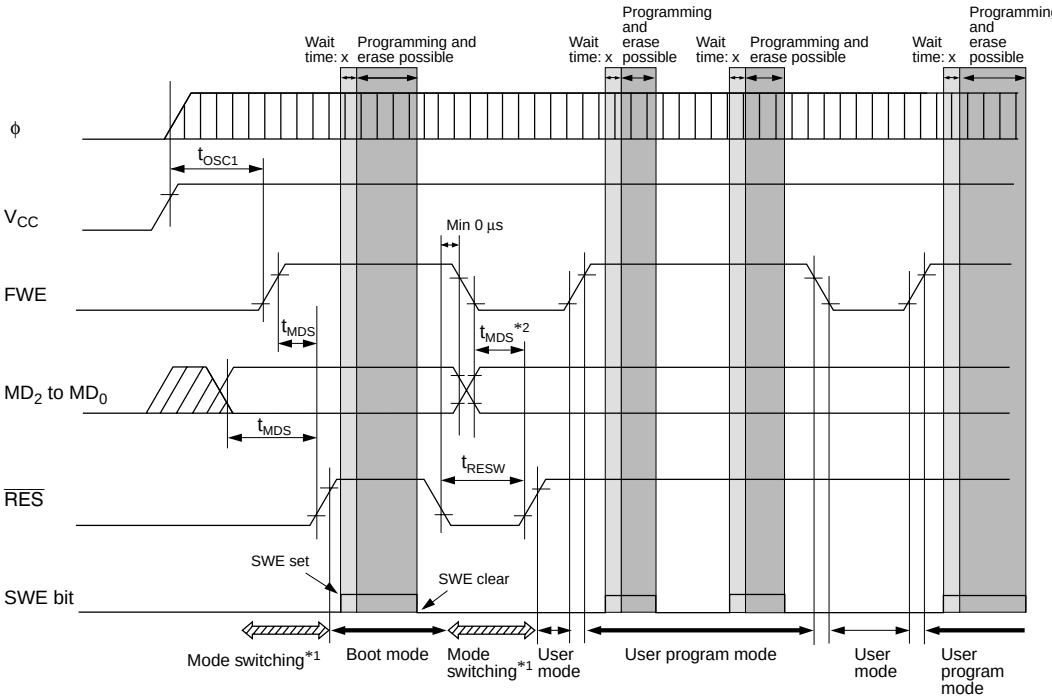


: Flash memory access disabled period
(x: Wait time after SWE setting)*2

: 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) except for mode switching.
2. See section 18.2.5, Flash Memory Characteristics.

Figure 15.26 Powering On/Off Timing (User Program Mode)



: Flash memory access disabled time
(x: Wait time after SWE setting)*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 via $\overline{RES}$ input is necessary.
During this switching period (period during which a low level is input to the $\overline{RES}$ pin), the state of the address dual port and bus control output signals ($\overline{AS}$,$\overline{RD}$,$\overline{WR}$) 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 t_{MDS} relative to the $\overline{RES}$ clear timing is necessary.
3. See section 18.2.5, Flash Memory Characteristics.

Figure 15.27 Mode Transition Timing
(Example: Boot Mode → User Mode ↔ User Program Mode)

15.10 Mask ROM Overview

15.10.1 Block Diagram

Figure 15.28 shows a block diagram of the ROM.

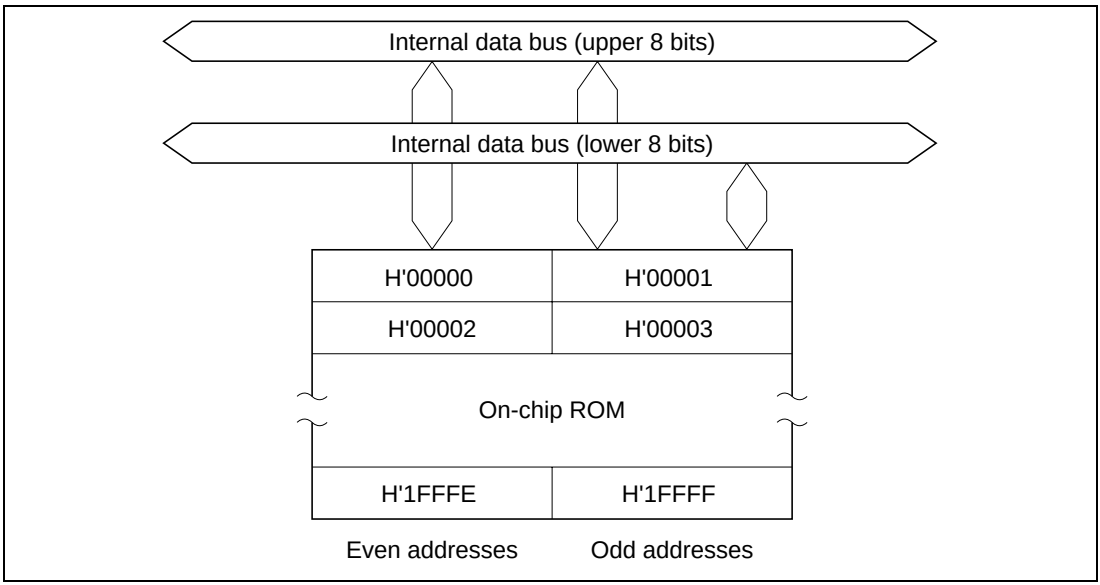


Figure 15.28 ROM Block Diagram (H8/3039)

15.11 Notes on Ordering Mask ROM Version Chip

When ordering the H8/3039 Group chips with a mask ROM, note the following.

- When ordering through an EPROM, use a 128-kbyte one.
- Fill all the unused addresses with H'FF as shown in figure15.29 to make the ROM data size 128 kbytes for all H8/3039 Group chips, which incorporate different sizes of ROM. This applies to ordering through an EPROM and through electrical data transfer.
- The flash memory versions only registers for flash memory control (FLMCR, EBR, RAMCR, and FLMSR) are not provided in the mask ROM versions. Reading the corresponding addresses in a mask ROM version will always return 1s, and writes to these addresses are disabled. This must be borne in mind when switching from the flash memory versions to a mask ROM version.

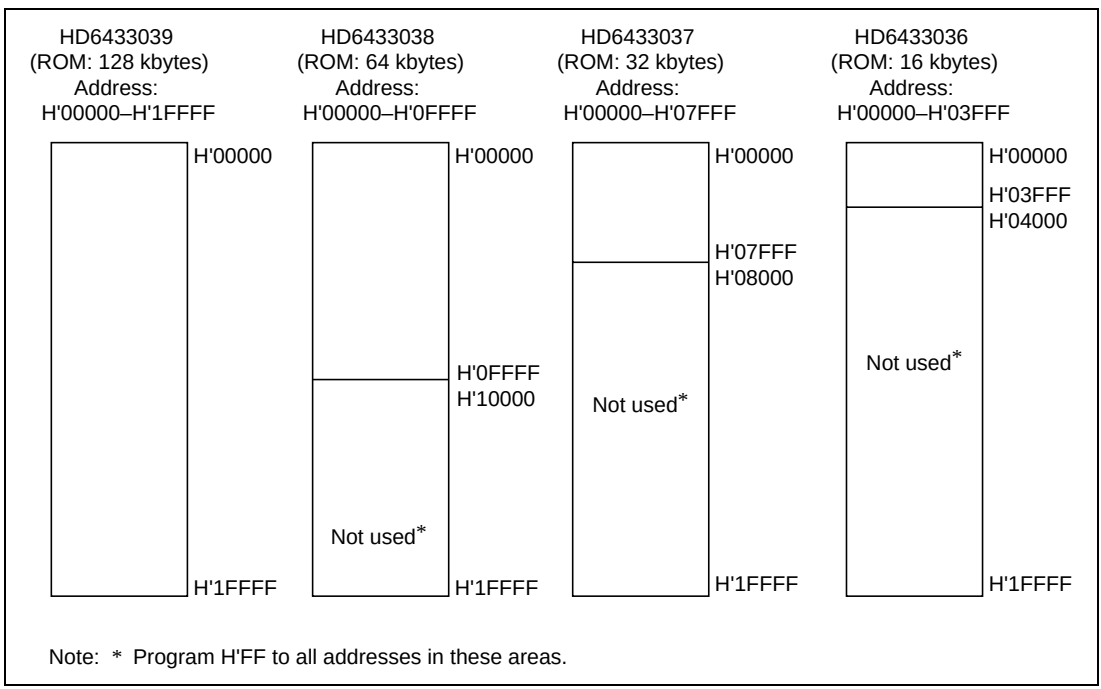


Figure 15.29 Mask ROM Addresses and Data

Section 16 Clock Pulse Generator

16.1 Overview

This LSI has a built-in clock pulse generator (CPG) that generates the system clock (ϕ) and other internal clock signals (ϕ/2 to ϕ/4096). After duty adjustment, a frequency divider divides the clock frequency to generate the system clock (ϕ). The system clock is output at the ϕ pin*[1] and furnished as a master clock to prescalers that supply clock signals to the on-chip supporting modules. Frequency division ratios of 1/1, 1/2, 1/4, and 1/8 can be selected for the frequency divider by settings in a division control register (DIVCR). Power consumption in the 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 PSTOP bit setting in the module standby control register (MSTCR). For details, see section 17.7, System Clock Output Disabling Function.

2. The division ratio of the frequency divider can be changed dynamically during operation. The clock output at the ϕ pin also changes when the division ratio is changed. The frequency output at the ϕ pin is shown below.

$$\phi = EXTAL \times n$$

EXTAL: Frequency of crystal resonator or external clock signal
n: Frequency division ratio (n = 1/1, 1/2, 1/4, or 1/8)

16.1.1 Block Diagram

Figure 16.1 shows a block diagram of the clock pulse generator.

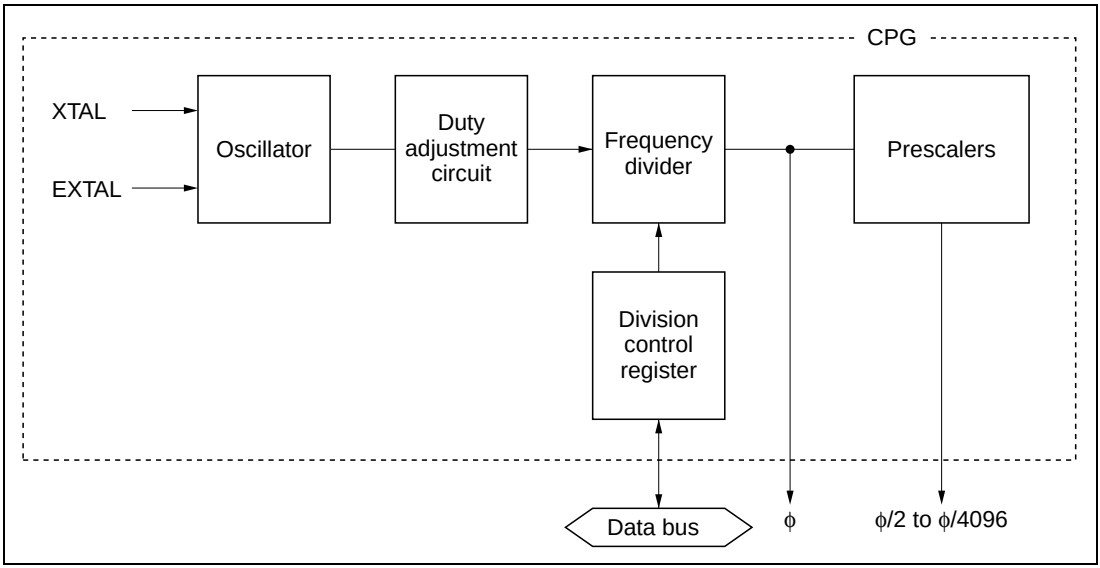


Figure 16.1 Block Diagram of Clock Pulse Generator

16.2 Oscillator Circuit

Clock pulses can be supplied by connecting a crystal resonator, or by input of an external clock signal.

16.2.1 Connecting a Crystal Resonator

Circuit Configuration

A crystal resonator can be connected as in the example in figure 16.2. The damping resistance Rd should be selected according to table 16.1. An AT-cut parallel-resonance crystal should be used.

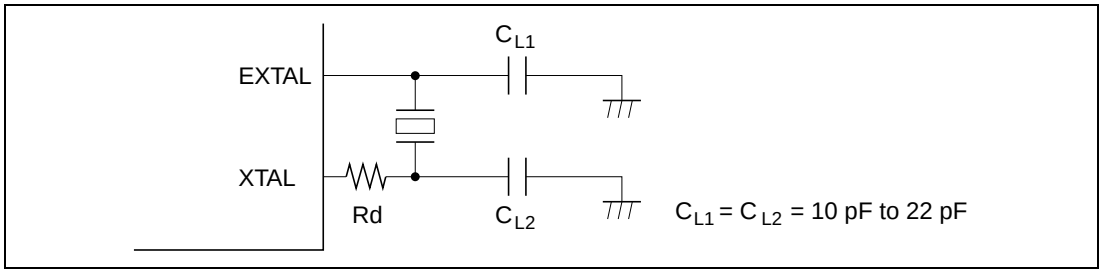


Figure 16.2 Connection of Crystal Resonator (Example)

Table 16.1 Damping Resistance Value (Example)

Frequency (MHz)	2	4	8	10	12	16	18
Rd (Ω)	1 k	500	200	0	0	0	0

Crystal Resonator

Figure 16.3 shows an equivalent circuit of the crystal resonator. The crystal resonator should have the characteristics listed in table 16.2.

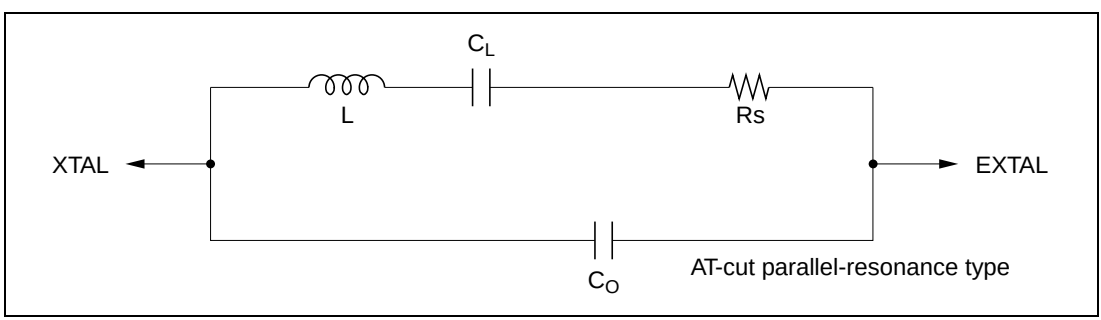


Figure 16.3 Crystal Resonator Equivalent Circuit

Table 16.2 Crystal Resonator Parameters

Frequency (MHz)	2	4	8	10	12	16	18
Rs max (Ω)	500	120	80	70	60	50	40
Co max (pF)	7	7	7	7	7	7	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 from interfering with correct oscillation. See figure 16.4.

When the board is designed, the crystal resonator and its load capacitors should be placed as close as possible to the XTAL and EXTAL pins.

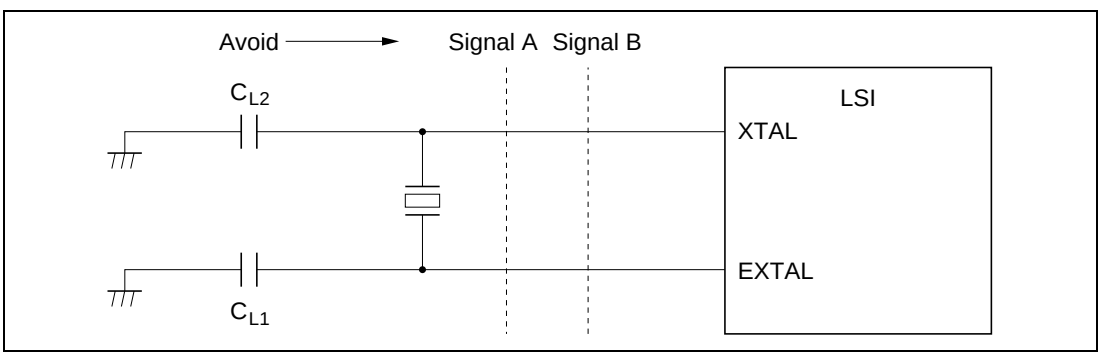


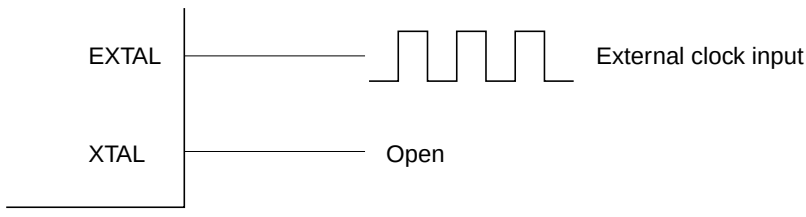
Figure 16.4 Example of Incorrect Board Design

16.2.2 External Clock Input

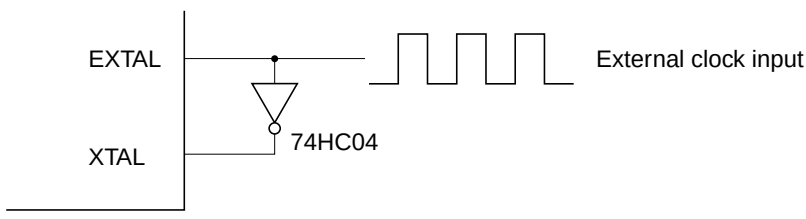
Circuit Configuration

An external clock signal can be input as shown in the examples in figure 16.5. In example b, the clock should be held high in standby mode.

If the XTAL pin is left open, the stray capacitance should not exceed 10 pF.



a. XTAL pin left open



b. Complementary clock input at XTAL pin

Figure 16.5 External Clock Input (Examples)

External Clock

The external clock frequency should be equal to the system clock frequency (ϕ). Table 16.3 and figure 16.6 indicate the clock timing.

Table 16.3 Clock Timing

		V_{cc} = 2.7 V to 5.5 V		V_{cc} = 5.0 V ±10%				
Item	**Symbol**	**Min**	**Max**	**Min**	**Max**	**Unit**	**Test Conditions**	
External clock rise time	t_{EXr}	—	10	—	5	ns	Figure 16.6	
External clock fall time	t_{EXf}	—	10	—	5	ns		
External clock input duty (a/t_{cyc})	—	30	70	30	70	%	ϕ ≥ 5 MHz	Figure 16.6
		40	60	40	60	%	ϕ < 5 MHz	
ϕ clock width duty (b/t_{cyc})	—	40	60	40	60	%		

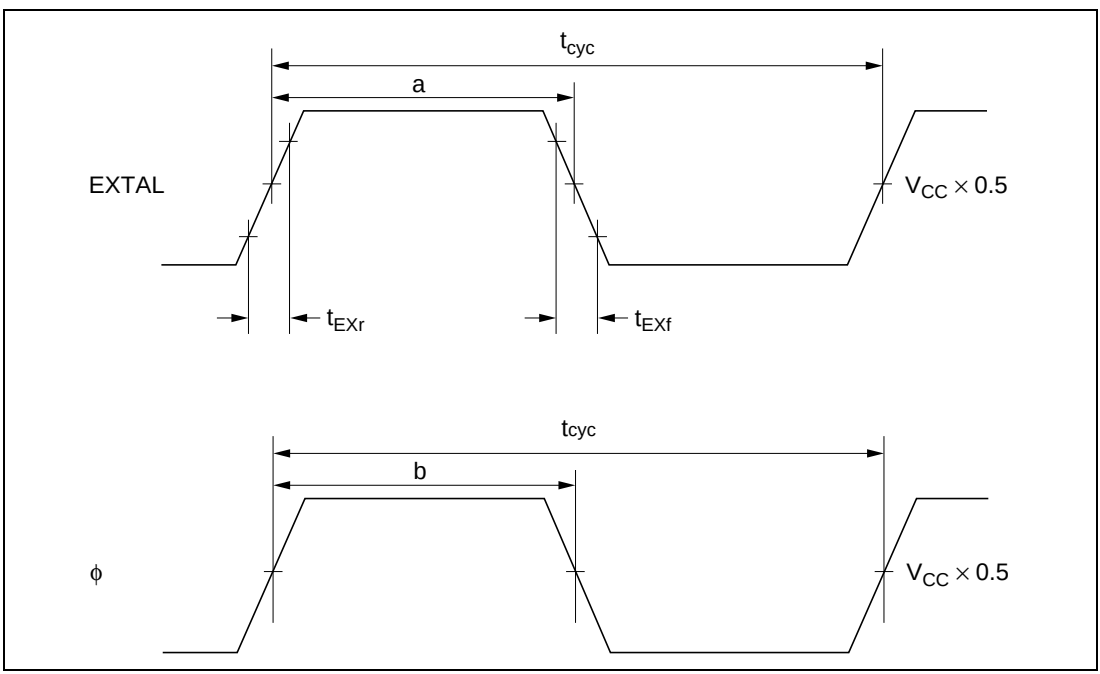


Figure 16.6 External Clock Input Timing

Table 16.4 and figure 16.7 show the timing for the external clock output stabilization delay time. The oscillator and duty correction circuit have the function of regulating the waveform of the external clock input to the EXTAL pin. When the specified clock signal is input to the EXTAL pin, internal clock signal output is confirmed after the elapse of the external clock output stabilization delay time (t_{DEXT}). As clock signal output is not confirmed during the t_{DEXT} period, the reset signal should be driven low and the reset state maintained during this time.

Table 16.4 External Clock Output Stabilization Delay Time

Conditions: V_{CC} = 2.7 V to 5.5 V, AV_{CC} = 2.7 V to 5.5 V, V_{SS} = AV_{SS} = 0 V

Item	Symbol	Min	Max	Unit	Notes
External clock output stabilization delay time	t_{DEXT}*	500	—	µs	Figure 16.7

Note: * t_{DEXT} includes a 10 t_{cyc} $\overline{RES}$ pulse width (t_{RESW}).

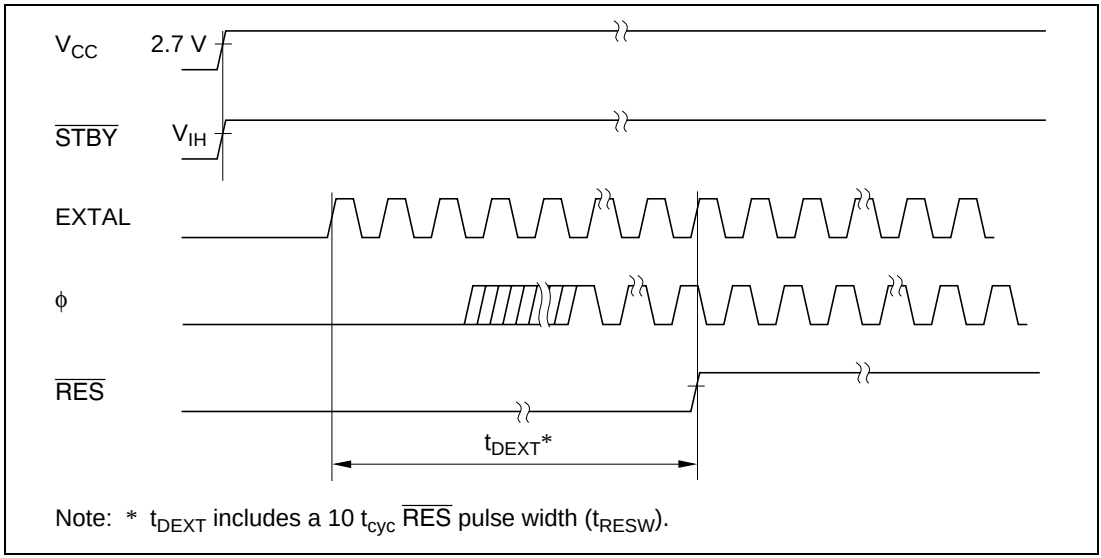


Figure 16.7 External Clock Output Stabilization Delay Time

16.3 Duty Adjustment Circuit

When the oscillator frequency is 5 MHz or higher, the duty adjustment circuit adjusts the duty cycle of the clock signal from the oscillator to generate the system clock (ϕ).

16.4 Prescalers

The prescalers divide the system clock (ϕ) to generate internal clocks (ϕ/2 to ϕ/4096).

16.5 Frequency Divider

The frequency divider divides the duty-adjusted clock signal to generate the system clock (ϕ). The frequency division ratio can be changed dynamically by modifying the value in DIVCR, as described below. Power consumption in the chip is reduced in almost direct proportion to the frequency division ratio. The system clock generated by the frequency divider can be output at the ϕ pin.

16.5.1 Register Configuration

Table 16.5 summarizes the frequency division register.

Table 16.5 Frequency Division Register

Address*	Name	Abbreviation	R/W	Initial Value
H'FF5D	Division control register	DIVCR	R/W	H'FC

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

16.5.2 Division Control Register (DIVCR)

DIVCR is an 8-bit readable/writable register that selects the division ratio of the frequency divider.

Bit	7	6	5	4	3	2	1	0
	—	—	—	—	—	—	DIV1	DIV0
Initial value	1	1	1	1	1	1	0	0
Read/Write	—	—	—	—	—	—	R/W	R/W

Reserved bits

Divide bits 1 and 0
These bits select the frequency division ratio

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

Bits 7 to 2—Reserved: These bits cannot be modified and are always read as 1.

Bits 1 and 0—Divide (DIV1 and DIV0): These bits select the frequency division ratio, as follows.

Bit 1 DIV1	Bit 0 DIV0	Frequency Division Ratio	
0	0	1/1	(Initial value)
0	1	1/2	
1	0	1/4	
1	1	1/8	

16.5.3 Usage Notes

The DIVCR setting changes the ϕ frequency, so note the following points.

- Select a frequency division ratio that stays within the assured operation range specified for the clock cycle time t_{cyc} in the AC electrical characteristics. Note that ϕ_{MIN} = 1 MHz. Avoid settings that give system clock frequencies less than 1 MHz.

All on-chip module operations are based on ϕ. Note that the timing of timer operations, serial communication, and other time-dependent processing differs before and after any change in the division ratio. The waiting time for exit from software standby mode also changes when the division ratio is changed. For details, see section 17.4.3, Selection of Oscillator Waiting Time after Exit from Software Standby Mode

Section 17 Power-Down State

17.1 Overview

This LSI has a power-down state that greatly reduces power consumption by halting CPU functions, and a module standby function that reduces power consumption by selectively halting on-chip modules. 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 the power-down state. The modules that can be halted are the ITU, SCI0, SCI1, and A/D converter.

Table 17.1 indicates the methods of entering and exiting these power-down modes and the status of the CPU and on-chip supporting modules in each mode.

Table 17.1 Power-Down State and Module Standby Function

Mode	Entering Conditions	State: Clock	CPU	CPU Registers	ITU	SCI0	SCI1	A/D	Supporting Modules	RAM	φ clock output	I/O Ports	Exiting Methods
Sleep mode	SLEEP instruction executed while SSBY = 0 in SYSCR	Active	Halted	Held	Active	Active	Active	Active	Active	Held	φoutput	Held	• Interrupt • $\overline{\text{RES}}$ • $\overline{\text{STBY}}$
Software standby mode	SLEEP instruction executed while SSBY = 1 in SYSCR	Halted	Halted	Held	Halted and reset	Halted and reset	Halted and reset	Halted and reset	Halted and reset	Held	High output	Held	• NMI • $\overline{\text{IRQ}_0}$ to $\overline{\text{IRQ}_1}$ • $\overline{\text{RES}}$ • $\overline{\text{STBY}}$
Hardware standby mode	Low input at $\overline{\text{STBY}}$ pin	Halted	Halted	Undetermined	Halted and reset	Halted and reset	Halted and reset	Halted and reset	Halted and reset	Held*2	High impedance	High impedance	• $\overline{\text{STBY}}$ • $\overline{\text{RES}}$
Module standby function	Corresponding bit set to 1 in MSTCR	Active	Active	—	Halted*1 and reset	Halted*1 and reset	Halted*1 and reset	Halted*1 and reset	Active	—	High impedance*1	—	• $\overline{\text{STBY}}$ • $\overline{\text{RES}}$ • Clear MSTCR bit to 0*3

Legend:
SYSCR: System control register
SSBY: Software standby bit
MSTCR: Module standby control register

Notes: 1. State in which the corresponding MSTCR bit was set to 1. For details see section 17.2.2, Module Standby Control Register (MSTCR).
2. The RAME bit must be cleared to 0 in SYSCR before the transition from the program execution state to hardware standby mode.
3. When a MSTCR bit is set to 1, the registers of the corresponding on-chip supporting module are initialized. To restart the module, first clear the MSTCR bit to 0, then set up the module registers again.

17.2 Register Configuration

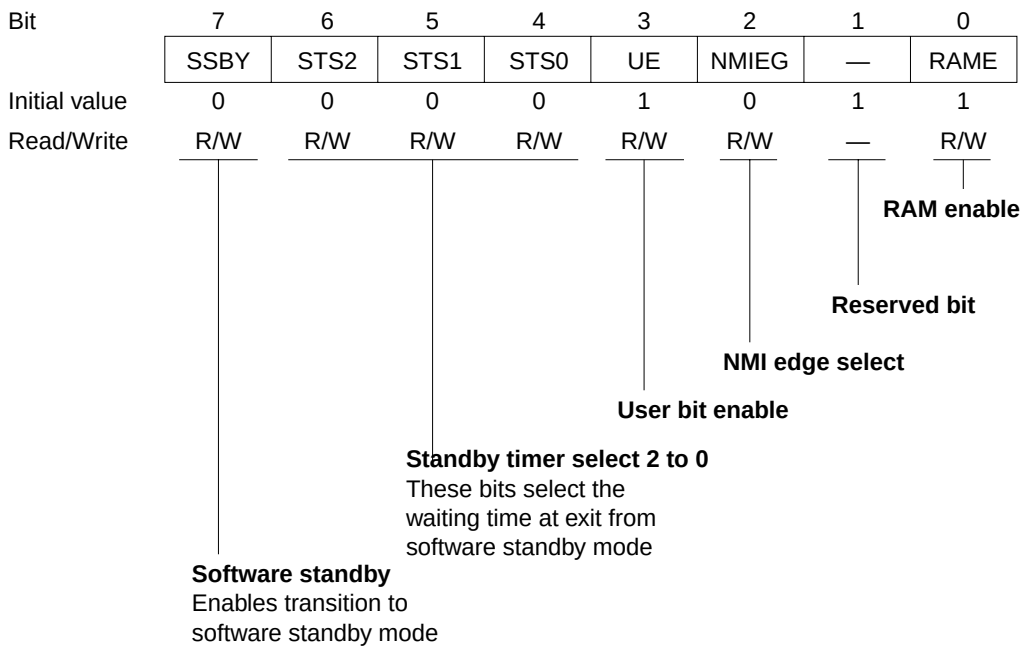
This LSI has a system control register (SYSCR) that controls the power-down state, and a module standby control register (MSTCR) that controls the module standby function. Table 17.2 summarizes this register.

Table 17.2 Register Configuration

Address*	Name	Abbreviation	R/W	Initial Value
H'FFF2	System control register	SYSCR	R/W	H'0B
H'FF5E	Module standby control register	MSTCR	R/W	H'40

Note: * Lower 16 bits of the address.

17.2.1 System Control Register (SYSCR)



SYSCR is an 8-bit readable/writable register. Bit 7 (SSBY) and bits 6 to 4 (STS2 to STS0) control the power-down state. For information on the other SYSCR bits, see section 3.3, System Control Register (SYSCR).

Bit 7—Software Standby (SSBY): Enables transition to software standby mode. When software standby mode is exited by an external interrupt, this bit remains set to 1 after the return to normal operation. To clear this bit, write 0.

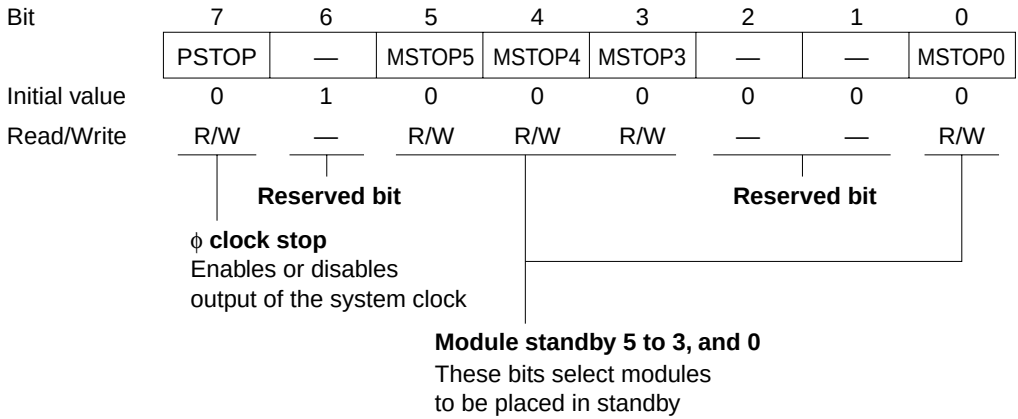
Bit 7 SSBY	Description	
0	SLEEP instruction causes transition to sleep mode	(Initial value)
1	SLEEP instruction causes transition to software standby mode	

Bits 6 to 4—Standby Timer Select (STS2 to STS0): These bits select the length of time the CPU and on-chip supporting modules wait for the clock to settle when software standby mode is exited by an external interrupt. If the clock is generated by a crystal resonator, set these bits according to the clock frequency so that the waiting time (for the clock to stabilize) will be at least 7 ms. See table 17.3. If an external clock is used, any setting is permitted.

Bit 6 STS2	Bit 5 STS1	Bit 4 STS0	Description	
0	0	0	Waiting time = 8192 states	(Initial value)
		1	Waiting time = 16384 states	
	1	0	Waiting time = 32768 states	
		1	Waiting time = 65536 states	
1	0	0	Waiting time = 131072 states	
	0	1	Waiting time = 1024 states	
	1	—	Illegal setting	

17.2.2 Module Standby Control Register (MSTCR)

MSTCR is an 8-bit readable/writable register that controls output of the system clock (ϕ). It also controls the module standby function, which places individual on-chip supporting modules in the standby state. Module standby can be designated for the ITU, SCI0, SCI1, and A/D converter modules.



MSTCR is initialized to H'40 by a reset and in hardware standby mode. It is not initialized in software standby mode.

Bit 7—ϕ Clock Stop (PSTOP): Enables or disables output of the system clock (ϕ).

Bit 7 PSTOP	Description	
0	System clock output is enabled	(Initial value)
1	System clock output is disabled	

Bit 6—Reserved: This bit cannot be modified and is 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	(Initial value)
1	ITU is in standby state	

Bit 4—Module Standby 4 (MSTOP4): Selects whether to place SCI0 in standby.

Bit 4 MSTOP4	Description	
0	SCI0 operates normally	(Initial value)
1	SCI0 is in standby state	

Bit 3—Module Standby 3 (MSTOP3): Selects whether to place SCI1 in standby.

Bit 3 MSTOP3	Description	
0	SCI1 operates normally	(Initial value)
1	SCI1 is in standby state	

Bits 2 to 1—Reserved: Bits 2 to 1 are reserved.

Bit 0—Module Standby 0 (MSTOP0): Selects whether to place the A/D converter in standby.

Bit 0 MSTOP0	Description	
0	A/D converter operates normally	(Initial value)
1	A/D converter is in standby state	

17.3 Sleep Mode

17.3.1 Transition to Sleep Mode

When the SSBY bit is cleared to 0 in the system control register (SYSCR), execution of the SLEEP instruction causes a transition from the program execution state to sleep mode. Immediately after executing the SLEEP instruction the CPU halts, but the contents of its internal registers are retained. The on-chip supporting modules do not halt in sleep mode. On-chip supporting modules which have been placed in standby by the module standby function, however, remain halted.

17.3.2 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 interrupt exception handling state. Sleep mode is not exited by an interrupt source in an on-chip supporting module if the interrupt is disabled in the on-chip supporting module. Sleep mode is not exited by an interrupt other then NMI if the interrupt is masked by interrupt priority settings (IPR) and the settings of the I and UI bits in CCR.

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 standby mode.

17.4 Software Standby Mode

17.4.1 Transition to Software Standby Mode

To enter software standby mode, execute the SLEEP instruction while the SSBY bit is set to 1 in SYSCR.

In software standby mode, current dissipation is reduced to an extremely low level because the CPU, clock, and on-chip supporting modules all halt. The on-chip supporting modules are reset and halted. As long as the specified voltage is supplied, however, CPU register contents and on-chip RAM data are retained. The settings of the I/O ports are also held.

17.4.2 Exit from Software Standby Mode

Software standby mode can be exited by input of an external interrupt at the NMI, $\overline{IRQ_0}$, $\overline{IRQ_1}$, or by input at the $\overline{RES}$ or $\overline{STBY}$ pin.

Exit by Interrupt: When an NMI, IRQ_0, or IRQ_1 interrupt request signal is received, the clock oscillator begins operating. After the oscillator settling time selected by bits STS2 to STS0 in SYSCR, stable clock signals are supplied to the entire chip, software standby mode ends, and interrupt exception handling begins. Software standby mode is not exited if the interrupt enable bits of interrupts IRQ_0, and IRQ_1 are cleared to 0, or if these interrupts are masked in the CPU.

Exit by $\overline{RES}$ Input: When the $\overline{RES}$ input goes low, the clock oscillator starts and clock pulses are supplied immediately to the entire chip. The $\overline{RES}$ signal must be held low long enough for the clock oscillator to stabilize. When $\overline{RES}$ goes high, the CPU starts reset exception handling.

Exit by $\overline{STBY}$ Input: Low input at the $\overline{STBY}$ pin causes a transition to hardware standby mode.

17.4.3 Selection of Oscillator Waiting Time after Exit from Software Standby Mode

Bits STS2 to STS0 in SYSCR, and its DIV1 and DIV0 in DIVCR should be set as follows.

Crystal Resonator

Set STS2 to STS0, and DIV1 and DIV0 so that the waiting time (for the clock to stabilize) is at least 7 ms. Table 17.3 indicates the waiting times that are selected by STS2 to STS0, and DIV1 and DIV0 settings at various system clock frequencies.

External Clock

Any value may be set.

Table 17.3 Clock Frequency and Waiting Time for Clock to Settle

DIV1	DIV0	STS2	STS1	STS0	Waiting Time	18 MHz	16 MHz	12 MHz	10 MHz	8 MHz	6 MHz	4 MHz	2 MHz	1 MHz	Unit
0	0	0	0	0	8192 states	0.46	0.51	0.65	0.8	1.0	1.3	2.0	4.1	8.2	ms
		0	0	1	16384 states	0.91	1.0	1.3	1.6	2.0	2.7	4.1	8.2	16.4	
		0	1	0	32768 states	1.8	2.0	2.7	3.3	4.1	5.5	8.2	16.4	32.8	
		0	1	1	65536 states	3.6	4.1	5.5	6.6	8.2	10.9	16.4	32.8	65.5	
		1	0	0	131072 states	7.3	8.2	10.9	13.1	16.4	21.8	32.8	65.5	131.1	
		1	0	1	1024 states	0.057	0.064	0.085	0.10	0.13	0.17	0.26	0.51	1.0	
		1	1	—	Illegal setting										
0	1	0	0	0	8192 states	0.91	1.02	1.4	1.6	2.0	2.7	4.1	8.2	16.4	ms
		0	0	1	16384 states	1.8	2.0	2.7	3.3	4.1	5.5	8.2	16.4	32.8	
		0	1	0	32768 states	3.6	4.1	5.5	6.6	8.2	10.9	16.4	32.8	65.5	
		0	1	1	65536 states	7.3	8.2	10.9	13.1	16.4	21.8	32.8	65.5	131.1	
		1	0	0	131072 states	14.6	16.4	21.8	26.2	32.8	43.7	65.5	131.1	262.1	
		1	0	1	1024 states	0.11	0.13	0.17	0.20	0.26	0.34	0.51	1.0	2.0	
		1	1	—	Illegal setting										
1	0	0	0	0	8192 states	1.8	2.0	2.7	3.3	4.1	5.5	8.2	16.4	32.8	ms
		0	0	1	16384 states	3.6	4.1	5.5	6.6	8.2	10.9	16.4	32.8	65.5	
		0	1	0	32768 states	7.3	8.2	10.9	13.1	16.4	21.8	32.8	65.5	131.1	
		0	1	1	65536 states	14.6	16.4	21.8	26.2	32.8	43.7	65.5	131.1	262.1	
		1	0	0	131072 states	29.1	32.8	43.7	52.4	65.5	87.4	131.1	262.1	524.3	
		1	0	1	1024 states	0.23	0.26	0.34	0.41	0.51	0.68	1.02	2.0	4.1	
		1	1	—	Illegal setting										
1	1	0	0	0	8192 states	3.6	4.1	5.5	6.6	8.2	10.9	16.4	32.8	65.5	ms
		0	0	1	16384 states	7.3	8.2	10.9	13.1	16.4	21.8	32.8	65.5	131.1	
		0	1	0	32768 states	14.6	16.4	21.8	26.2	32.8	43.7	65.5	131.1	262.1	
		0	1	1	65536 states	29.1	32.8	43.7	52.4	65.5	87.4	131.1	262.1	524.3	
		1	0	0	131072 states	58.3	65.5	87.4	104.9	131.1	174.8	262.1	524.3	1048.6	
		1	0	1	1024 states	0.46	0.51	0.68	0.82	1.0	1.4	2.0	4.1	8.2	
		1	1	—	Illegal setting										

: Recommended setting

17.4.4 Sample Application of Software Standby Mode

Figure 17.1 shows an example in which software standby mode is entered at the fall of NMI and exited at the rise of NMI.

With the NMI edge select bit (NMIEG) cleared to 0 in SYSCR (selecting the falling edge), an NMI interrupt occurs. Next the NMIEG bit is set to 1 (selecting the rising edge) and the SSBY bit is set to 1; then the SLEEP instruction is executed to enter software standby mode.

Software standby mode is exited at the next rising edge of the NMI signal.

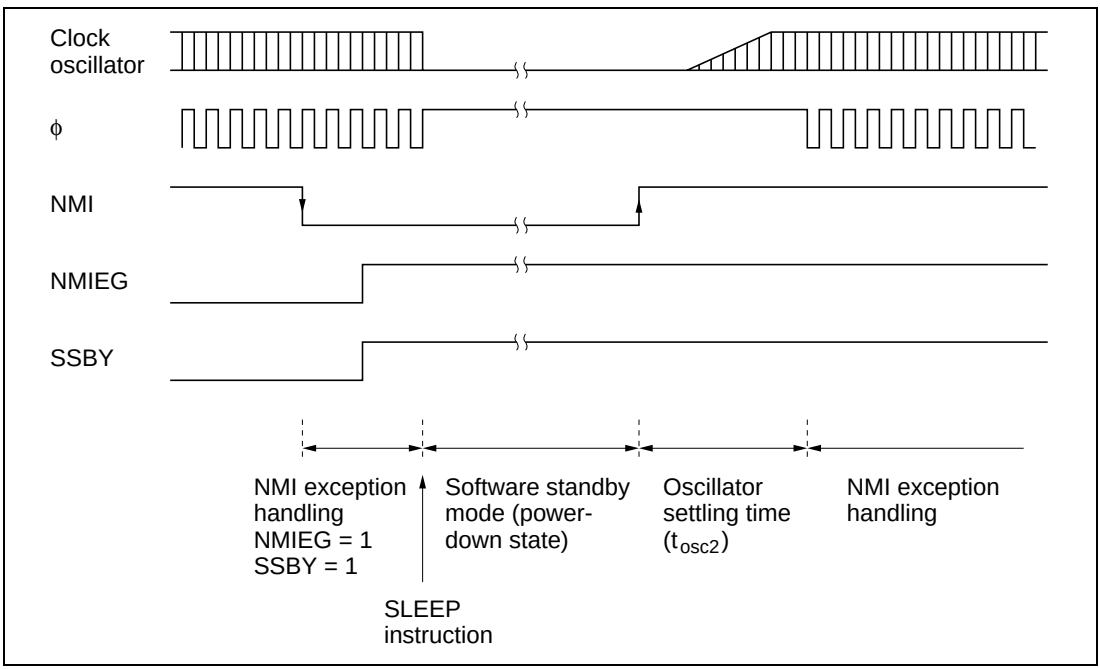


Figure 17.1 NMI Timing for Software Standby Mode (Example)

17.4.5 Usage Note

The I/O ports retain their existing states in software standby mode. If a port is in the high output state, its output current is not reduced.

17.5 Hardware Standby Mode

17.5.1 Transition to Hardware Standby Mode

Regardless of its current state, the chip enters hardware standby mode whenever the $\overline{\text{STBY}}$ pin goes low. Hardware standby mode reduces power consumption drastically by halting all functions of the CPU and on-chip supporting modules. All modules are reset except the on-chip RAM. As long as the specified voltage is supplied, on-chip RAM data is 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 data.

The inputs at the mode pins (MD_2 to MD_0) should not be changed during hardware standby mode.

17.5.2 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 low, when $\overline{\text{STBY}}$ goes high, the clock oscillator starts running. $\overline{\text{RES}}$ should be held low long enough for the clock oscillator to settle. When $\overline{\text{RES}}$ goes high, reset exception handling begins, followed by a transition to the program execution state.

17.5.3 Timing for Hardware Standby Mode

Figure 17.2 shows the timing relationships for hardware standby mode. To enter hardware standby mode, first drive $\overline{\text{RES}}$ low, then drive $\overline{\text{STBY}}$ low. To exit hardware standby mode, first drive $\overline{\text{STBY}}$ high, wait for the clock to settle, then bring $\overline{\text{RES}}$ from low to high.

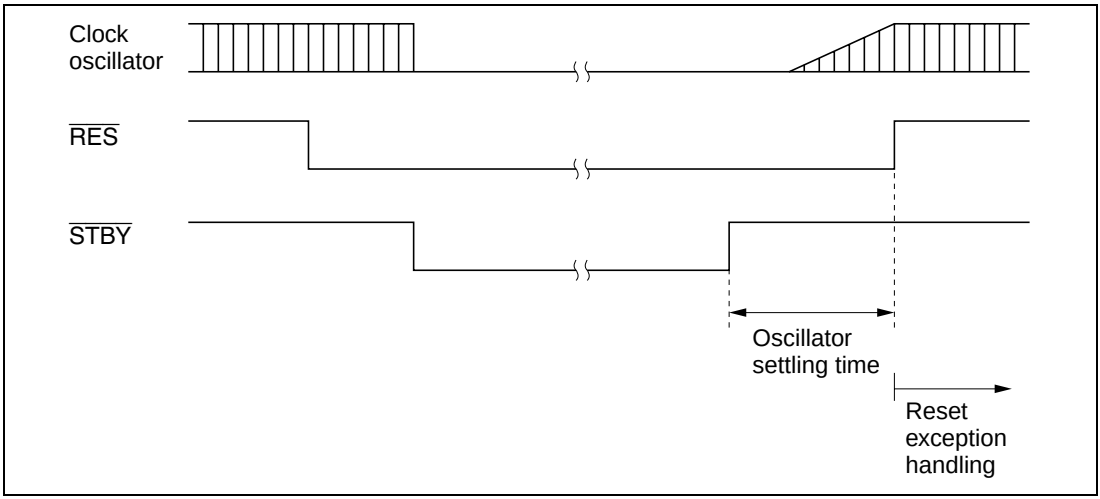


Figure 17.2 Hardware Standby Mode Timing

17.6 Module Standby Function

17.6.1 Module Standby Timing

The module standby function can halt several of the on-chip supporting modules (the ITU, SCI0, SCI1, and A/D converter) independently of the power-down state. This standby function is controlled by bits MSTOP5 to MSTOP3 and MSTOP0 in MSTCR. When one of these bits is set to 1, the corresponding on-chip supporting module is placed in standby and halts at the beginning of the next bus cycle after the MSTCR write cycle.

17.6.2 Read/Write in Module Standby

When an on-chip supporting module is in module standby, read/write access to its registers is disabled. Read access always results in H'FF data. Write access is ignored.

17.6.3 Usage Notes

When using the module standby function, note the following points.

Cancellation of Interrupt Handling: When an on-chip supporting module is placed in standby by the module standby function, its registers are initialized, including registers with interrupt request flags. Consequently, if an interrupt occurs just before the MSTOP bit is set to 1, the interrupt will not be recognized. The interrupt source will not be held pending.

Pin States: Pins used by an on-chip supporting module lose their module functions when the module is placed in module standby. What happens after that depends on the particular pin. For details, see section 7, I/O Ports. Pins that change from the input to the output state require special care. For example, if SCI1 is placed in module standby, the receive data pin loses its receive data function and becomes a generic I/O pin. If its data direction bit is set to 1, the pin becomes a data output pin, and its output may collide with external serial data. Data collisions should be prevented 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 standby function, all its registers are initialized. To restart the module, after its MSTOP bit is cleared to 0, its registers must be set up again. It is not possible to write to the registers while the MSTOP bit is set to 1.

17.7 System Clock Output Disabling Function

Output of the system clock (ϕ) can be controlled by the PSTOP bit in MSTCR. When the PSTOP bit is set to 1, output of the system clock halts and the ϕ pin is placed in the high-impedance state. Figure 17.3 shows the timing of the stopping and starting of system clock output. When the PSTOP bit is cleared to 0, output of the system clock is enabled. Table 17.4 indicates the state of the ϕ pin in various operating states.

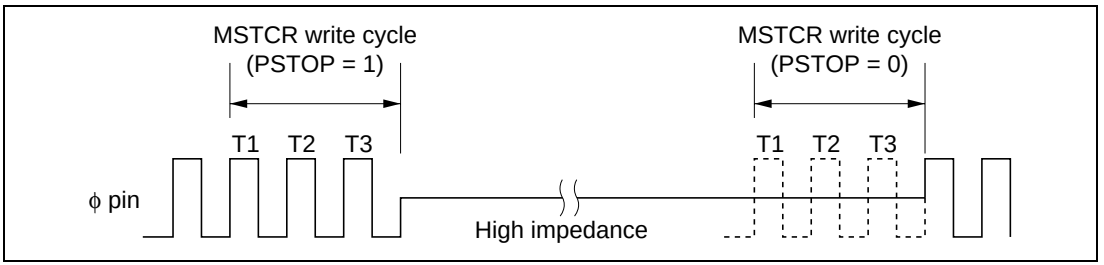


Figure 17.3 Starting and Stopping of System Clock Output

Table 17.4 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

Section 18 Electrical Characteristics

18.1 Electrical Characteristics of Mask ROM Version

18.1.1 Absolute Maximum Ratings

Table 18.1 lists the absolute maximum ratings.

Table 18.1 Absolute Maximum Ratings

Item	Symbol	Value	Unit
Power supply voltage	V_{CC}	–0.3 to +7.0	V
Input voltage (except port 7)*	V_{in}	–0.3 to V_{CC} +0.3	V
Input voltage (port 7)	V_{in}	–0.3 to AV_{CC} +0.3	V
Analog power supply voltage	AV_{CC}	–0.3 to +7.0	V
Analog input voltage	V_{AN}	–0.3 to AV_{CC} +0.3	V
Operating temperature	T_{opr}	Regular specifications: –20 to +75	°C
		Wide-range specifications: –40 to +85	°C
Storage temperature	T_{stg}	–55 to +125	°C

Caution: Permanent damage to the chip may result if absolute maximum ratings are exceeded.

Note: * 12 V must not be applied to any pin, as this will cause permanent damage to the chip.

18.1.2 DC Characteristics

Table 18.2 lists the DC characteristics. Table 18.3 lists the permissible output currents.

Table 18.2 DC Characteristics (1)

Conditions: V_{CC} = 5.0 V ±10%, AV_{CC} = 5.0 V ±10%, V_{SS} = AV_{SS} = 0 V*[1], T_a = –20°C to +75°C (regular specifications), T_a = –40°C to +85°C (wide-range specifications)

Item		Symbol	Min	Typ	Max	Unit	Test Conditions
Schmitt trigger input voltages	Port A,	V_T^-	1.0	—	—	V	
	$P8_0$ to $P8_1$,	V_T^+	—	—	$V_{CC} \times 0.7$	V	
	PB_0 to PB_3	$V_T^+ - V_T^-$	0.4	—	—	V	
Input high voltage	$\overline{RES}$, $\overline{STBY}$, NMI, MD_2, MD_1, 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, 2, 3, 5, 6, 9, PB_4, PB_5, PB_7		2.0	—	V_{CC} +0.3	V	
Input low voltage	$\overline{RES}$, $\overline{STBY}$, MD_2, MD_1, MD_0	V_{IL}	–0.3	—	0.5	V	
	NMI, EXTAL, ports 1, 2, 3, 5, 6, 7, 9, PB_4, PB_5, PB_7		–0.3	—	0.8	V	
Output high voltage	All output pins (except $\overline{RESO}$)	V_{OH}	V_{CC} –0.5	—	—	V	I_{OH} = –200 µA
			3.5	—	—	V	I_{OH} = –1 mA
Output low voltage	All output pins (except $\overline{RESO}$)	V_{OL}	—	—	0.4	V	I_{OL} = 1.6 mA
	Ports 1, 2, 5, B		—	—	1.0	V	I_{OL} = 10 mA
	$\overline{RESO}$		—	—	0.4	V	I_{OL} = 2.6 mA

Item		Symbol	Min	Typ	Max	Unit	Test Conditions
Input leakage current	$\overline{STBY}$, NMI, $\overline{RES}$, MD_2, MD_1, MD_0	$\|I_{in}\|$	—	—	1.0	µA	V_{in} = 0.5 to V_{CC} –0.5 V
	Port 7		—	—	1.0	µA	V_{in} = 0.5 to AV_{CC} –0.5 V
Three-state leakage current (off state)	Ports 1, 2, 3, 5, 6, 8 to B	$\|I_{TSI}\|$	—	—	1.0	µA	V_{in} = 0.5 to V_{CC} –0.5 V
	$\overline{RESO}$		—	—	10.0	µA	
Input pull-up MOS current	Ports 2, 5	$-I_p$	50	—	300	µA	V_{in} = 0 V
Input capacitance	NMI, $\overline{RES}$	C_{in}	—	—	50	pF	V_{in} = 0 V, f = 1 MHz, T_a = 25°C
	All input pins except NMI and $\overline{RES}$		—	—	20		
Current dissipation*2	Normal operation	I_{CC}	—	50	70	mA	f = 18 MHz
	Sleep mode		—	35	50		f = 18 MHz
	Standby mode*3		—	0.01	5.0	µA	$T_a \leq 50°C$
			—	—	20.0		50°C < T_a
Analog power supply current	During A/D conversion	AI_{CC}	—	1.7	2.8	mA	
	Idle		—	0.02	10.0	µA	
RAM standby voltage		V_{RAM}	2.0	—	—	V	

Notes: 1. If the A/D converter is not used, do not leave the AV_{CC} and AV_{SS} pins open. Connect AV_{CC} 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 with all 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 V.

Table 18.2 DC Characteristics (2)

Conditions: V_{CC} = 2.7 V to 5.5 V, AV_{CC} = 2.7 V to 5.5 V, V_{SS} = AV_{SS} = 0 V*[1], T_a = –20°C to +75°C (regular specifications), T_a = –40°C to +85°C (wide-range specifications)

Item		Symbol	Min	Typ	Max	Unit	Test Conditions
Schmitt trigger input voltages	Port A, $P8_0$ to $P8_1$, PB_0 to PB_3	V_T^-	$V_{CC} \times 0.2$	—	—	V	
		V_T^+	—	—	$V_{CC} \times 0.7$	V	
		$V_T^+ - V_T^-$	$V_{CC} \times 0.04$	—	—	V	
Input high voltage	$\overline{RES}$, $\overline{STBY}$, NMI, MD_2, MD_1, 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, 2, 3, 5, 6, 9, PB_4, PB_5, PB_7		$V_{CC} \times 0.7$	—	V_{CC} +0.3	V	
Input low voltage	$\overline{RES}$, $\overline{STBY}$, MD_2, MD_1, MD_0	V_{IL}	–0.3	—	$V_{CC} \times 0.1$	V	
	NMI, EXTAL, ports 1, 2, 3, 5, 6, 7, 9, PB_4, PB_5, PB_7		–0.3	—	$V_{CC} \times 0.2$	V	V_{CC} < 4.0 V
					0.8	V	V_{CC} = 4.0 V to 5.5 V
Output high voltage	All output pins (except $\overline{RESO}$)	V_{OH}	V_{CC} –0.5	—	—	V	I_{OH} = –200 µA
			V_{CC} –1.0	—	—	V	I_{OH} = –1 mA
Output low voltage	All output pins (except $\overline{RESO}$)	V_{OL}	—	—	0.4	V	I_{OL} = 1.0 mA
	Ports 1, 2, 5, B		—	—	1.0	V	$V_{CC} \le 4$ V, I_{OL} = 5 mA, 4 V < $V_{CC} \le$ 5.5 V, I_{OL} = 10 mA
	$\overline{RESO}$		—	—	0.4	V	I_{OL} = 1.6 mA
Input leakage current	$\overline{STBY}$, NMI, $\overline{RES}$, MD_2, MD_1, MD_0	$\lvert I_{in} \rvert$	—	—	1.0	µA	V_{in} = 0.5 V to V_{CC} –0.5 V
	Port 7		—	—	1.0	µA	V_{in} = 0.5 V to AV_{CC} –0.5 V

Item		Symbol	Min	Typ	Max	Unit	Test Conditions
Three-state leakage current (off state)	Ports 1, 2, 3, 5, 6, 8, 9, A, B	$\|I_{TSI}\|$	—	—	1.0	µA	V_{in} = 0.5 V to V_{CC} –0.5 V
	$\overline{RESO}$		—	—	10.0	µA	
Input pull-up MOS current	Ports 2, 5	$-I_p$	10	—	300	µA	V_{CC} = 2.7 V to 5.5 V, V_{in} = 0 V
Input capacitance	NMI, $\overline{RES}$	C_{in}	—	—	50	pF	V_{in} = 0 V, f = 1 MHz , Ta = 25°C
	All input pins except NMI and $\overline{RES}$		—	—	20		
Current dissipation*2	Normal operation	I_{CC}*4	—	12 (3.0 V)	33.8 (5.5 V)	mA	f = 8 MHz
	Sleep mode		—	8 (3.0 V)	25.0 (5.5 V)	mA	f = 8 MHz
	Standby mode*3		—	0.01	5.0	µA	$T_a \leq 50°C$
			—	—	20.0	µA	$50°C < T_a$
Analog power supply current	During A/D conversion	AI_{CC}	—	1.3	2.5	mA	AV_{CC} = 3.0 V
			—	1.7	2.8	mA	AV_{CC} = 5.0 V
	Idle		—	0.02	10.0	µA	
RAM standby voltage		V_{RAM}	2.0	—	—	V	

Notes: 1. If the A/D converter is not used, do not leave the AV_{CC} and AV_{SS} pins open. Connect AV_{CC} 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 with all output pins unloaded and the on-chip pull-up MOS transistors in the off state.
3. The values are for $V_{RAM} \leq V_{CC} < 2.7$ V, V_{IH} min = $V_{CC} \times 0.9$, and V_{IL} max = 0.3 V.
4. I_{CC} depends on V_{CC} and f as follows:
   I_{CC} max = 3.0 (mA) + 0.7 (mA/MHz · V) × V_{CC} × f [normal mode]
   I_{CC} max = 3.0 (mA) + 0.5 (mA/MHz · V) × V_{CC} × f [sleep mode]

Table 18.2 DC Characteristics (3)

Conditions: V_{CC} = 3.0 V to 5.5 V, AV_{CC} = 3.0 V to 5.5 V, V_{SS} = AV_{SS} = 0 V*[1], T_a = –20°C to +75°C (regular specifications), T_a = –40°C to +85°C (wide-range specifications)

Item		Symbol	Min	Typ	Max	Unit	Test Conditions
Schmitt trigger input voltages	Port A, $P8_0$ to $P8_1$, PB_0 to PB_3	V_T^-	$V_{CC} \times 0.2$	—	—	V	
		V_T^+	—	—	$V_{CC} \times 0.7$	V	
		$V_T^+ - V_T^-$	$V_{CC} \times 0.04$	—	—	V	
Input high voltage	$\overline{RES}$, $\overline{STBY}$, NMI, MD_2, MD_1, 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, 2, 3, 5, 6, 9, PB_4, PB_5, PB_7		$V_{CC} \times 0.7$	—	V_{CC} +0.3	V	
Input low voltage	$\overline{RES}$, $\overline{STBY}$, MD_2, MD_1, MD_0	V_{IL}	–0.3	—	$V_{CC} \times 0.1$	V	
	NMI, EXTAL, ports 1, 2, 3, 5, 6, 7, 9, PB_4, PB_5, PB_7		–0.3	—	$V_{CC} \times 0.2$	V	V_{CC} < 4.0 V
					0.8	V	V_{CC} = 4.0 V to 5.5 V
Output high voltage	All output pins (except $\overline{RESO}$)	V_{OH}	V_{CC} –0.5	—	—	V	I_{OH} = –200 µA
			V_{CC} –1.0	—	—	V	I_{OH} = –1 mA
Output low voltage	All output pins (except $\overline{RESO}$)	V_{OL}	—	—	0.4	V	I_{OL} = 1.0 mA
	Ports 1, 2, 5, B		—	—	1.0	V	$V_{CC} \leq 4$ V, I_{OL} = 5 mA, 4 V < $V_{CC} \leq$ 5.5 V, I_{OL} = 10 mA
	$\overline{RESO}$		—	—	0.4	V	I_{OL} = 1.6 mA
Input leakage current	$\overline{STBY}$, NMI, $\overline{RES}$, MD_2, MD_1, MD_0	$\lvert I_{in} \rvert$	—	—	1.0	µA	V_{in} = 0.5 V to V_{CC} –0.5 V
	Port 7		—	—	1.0	µA	V_{in} = 0.5 V to AV_{CC} –0.5 V

Item		Symbol	Min	Typ	Max	Unit	Test Conditions
Three-state leakage current (off state)	Ports 1, 2, 3, 5, 6, 8 to B	$\|I_{TSI}\|$	—	—	1.0	µA	V_{in} = 0.5 V to V_{CC} –0.5 V
	$\overline{RESO}$		—	—	10.0	µA	
Input pull-up MOS current	Ports 2, 5	$-I_p$	10	—	300	µA	V_{CC} = 3.0 V to 5.5 V, V_{in} = 0 V
Input capacitance	NMI, $\overline{RES}$	C_{in}	—	—	50	pF	V_{in} = 0 V, f = 1 MHz, T_a = 25°C
	All input pins except NMI and $\overline{RES}$		—	—	20		
Current dissipation*2	Normal operation	I_{CC}*4	—	15 (3.0 V)	41.5 (5.5 V)	mA	f = 10 MHz
	Sleep mode		—	10 (3.0 V)	30.5 (5.5 V)	mA	f = 10 MHz
	Standby mode*3		—	0.01	5.0	µA	$T_a \leq$ 50°C
			—	—	20.0	µA	50°C < T_a
Analog power supply current	During A/D conversion	AI_{CC}	—	1.3	2.5	mA	AV_{CC} = 3.0 V
			—	1.7	—	mA	AV_{CC} = 5.0 V
	Idle		—	0.02	10.0	µA	
RAM standby voltage		V_{RAM}	2.0	—	—	V	

Notes: 1. If the A/D converter is not used, do not leave the AV_{CC} and AV_{SS} pins open. Connect AV_{CC} 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 with all 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. I_{CC} depends on V_{CC} and f as follows:
   I_{CC} max = 3.0 (mA) + 0.7 (mA/MHz · V) × V_{CC} × f [normal mode]
   I_{CC} max = 3.0 (mA) + 0.5 (mA/MHz · V) × V_{CC} × f [sleep mode]

Table 18.3 Permissible Output Currents

Conditions: V_{CC} = 2.7 V to 5.5 V, AV_{CC} = 2.7 V to 5.5 V, V_{SS} = AV_{SS} = 0 V, T_a = –20°C to +75°C (regular specifications), T_a = –40°C to +85°C (wide-range specifications)

Item		Symbol	Min	Typ	Max	Unit
Permissible output low current (per pin)	Ports 1, 2, 5 and B	I_{OL}	—	—	10	mA
	Other output pins		—	—	2.0	mA
Permissible output low current (total)	Total of 27 pins including ports 1, 2, 5 and B	ΣI_{OL}	—	—	80	mA
	Total of 23 pins, including ports 8, 9, A and B		—	—	75*[2]/65*[1]	mA
	Total of all output pins, including the above		—	—	120	mA
Permissible output high current (per pin)	All output pins	I_{OH}	—	—	2.0	mA
Permissible output high current (total)	Total of all output pins	ΣI_{OH}	—	—	40	mA

Notes: To protect chip reliability, do not exceed the output current values in table 18.3.

When driving a Darlington pair or LED, always insert a current-limiting resistor in the output line, as shown in figures 18.1 and 18.2.

1. The value is for conditions: V_{CC} = 2.7 V to 5.5 V, AV_{CC} = 2.7 V to 5.5 V
2. The value is for conditions: V_{CC} = 4.5 V to 5.5 V, AV_{CC} = 4.5 V to 5.5 V

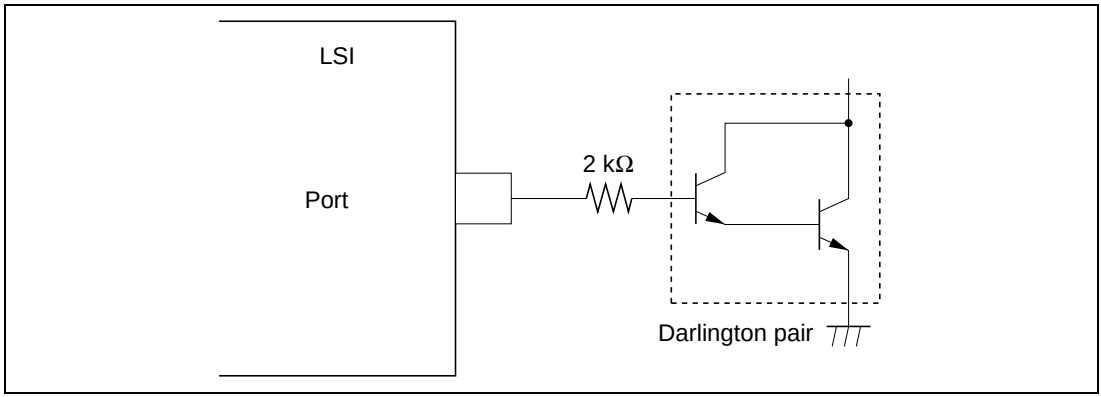


Figure 18.1 Darlington Pair Drive Circuit (Example)

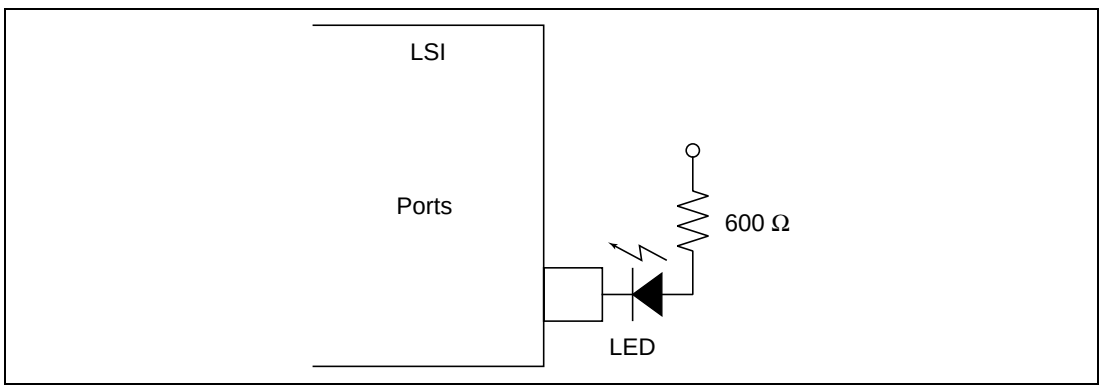


Figure 18.2 LED Drive Circuit (Example)

18.1.3 AC Characteristics

Bus timing parameters are listed in table 18.4. Control signal timing parameters are listed in table 18.5. Timing parameters of the on-chip supporting modules are listed in table 18.6.

Table 18.4 Bus Timing

Condition A: V_{CC} = 2.7 V to 5.5 V, AV_{CC} = 2.7 V to 5.5 V, V_{SS} = AV_{SS} = 0 V, ϕ = 2 MHz to 8 MHz, T_a = –20°C to +75°C (regular specifications), T_a = –40°C to +85°C (wide-range specifications)

Condition B: V_{CC} = 3.0 V to 5.5 V, AV_{CC} = 3.0 V to 5.5 V, V_{SS} = AV_{SS} = 0 V, ϕ = 2 MHz to 10 MHz, T_a = –20°C to +75°C (regular specifications), T_a = –40°C to +85°C (wide-range specifications)

Condition C: V_{CC} = 5.0 V ±10%, AV_{CC} = 5.0 V ±10%, V_{SS} = AV_{SS} = 0 V, ϕ = 2 MHz to 18 MHz, T_a = –20°C to +75°C (regular specifications), T_a = –40°C to +85°C (wide-range specifications)

		Condition A		Condition B		Condition C			
		8 MHz		10 MHz		18 MHz			
Item	Symbol	Min	Max	Min	Max	Min	Max	Unit	Test Conditions
Clock cycle time	t_{cyc}	125	500	100	500	55.5	500	ns	Figure 18.7, Figure 18.8
Clock low pulse width	t_{CL}	40	—	30	—	17	—		
Clock high pulse width	t_{CH}	40	—	30	—	17	—		
Clock rise time	t_{Cr}	—	20	—	15	—	10		
Clock fall time	t_{Cf}	—	20	—	15	—	10		
Address delay time	t_{AD}	—	60	—	50	—	25		
Address hold time	t_{AH}	25	—	20	—	10	—		
Address strobe delay time	t_{ASD}	—	60	—	40	—	25		
Write strobe delay time	t_{WSD}	—	60	—	50	—	25		
Strobe delay time	t_{SD}	—	60	—	50	—	25		
Write data strobe pulse width 1	t_{WSW1}*	85	—	60	—	32	—		
Write data strobe pulse width 2	t_{WSW2}*	150	—	110	—	62	—		
Address setup time 1	t_{AS1}	20	—	15	—	10	—		
Address setup time 2	t_{AS2}	80	—	65	—	38	—		
Read data setup time	t_{RDS}	50	—	35	—	15	—		
Read data hold time	t_{RDH}	0	—	0	—	0	—		

Item	Symbol	Condition A 8 MHz Min	Condition A 8 MHz Max	Condition B 10 MHz Min	Condition B 10 MHz Max	Condition C 18 MHz Min	Condition C 18 MHz Max	Unit	Test Conditions
Write data delay time	t_{WDD}	—	75	—	75	—	55	ns	Figure 18.7, Figure 18.8
Write data setup time 1	t_{WDS1}	60	—	40	—	10	—		
Write data setup time 2	t_{WDS2}	5	—	–10	—	–10	—		
Write data hold time	t_{WDH}	25	—	20	—	20	—		
Read data access time 1	t_{ACC1}*	—	120	—	100	—	50		
Read data access time 2	t_{ACC2}*	—	240	—	200	—	105		
Read data access time 3	t_{ACC3}*	—	70	—	50	—	20		
Read data access time 4	t_{ACC4}*	—	180	—	150	—	80		
Precharge time	t_{PCH}*	85	—	60	—	40	—		
Wait setup time	t_{WTS}	40	—	40	—	25	—		Figure 18.9
Wait hold time	t_{WTH}	10	—	10	—	5	—		

Note: * For Condition A, the following times depend on the clock cycle time as shown below.

$t_{ACC1} = 1.5 \times t_{cyc} - 68$ (ns)
$t_{ACC2} = 2.5 \times t_{cyc} - 73$ (ns)
$t_{ACC3} = 1.0 \times t_{cyc} - 55$ (ns)
$t_{ACC4} = 2.0 \times t_{cyc} - 70$ (ns)
$t_{WSW1} = 1.0 \times t_{cyc} - 40$ (ns)
$t_{WSW2} = 1.5 \times t_{cyc} - 38$ (ns)
$t_{PCH} = 1.0 \times t_{cyc} - 40$ (ns)

For Condition B, the following times depend on the clock cycle time as shown below.

$t_{ACC1} = 1.5 \times t_{cyc} - 50$ (ns)
$t_{ACC2} = 2.5 \times t_{cyc} - 50$ (ns)
$t_{ACC3} = 1.0 \times t_{cyc} - 50$ (ns)
$t_{ACC4} = 2.0 \times t_{cyc} - 50$ (ns)
$t_{WSW1} = 1.0 \times t_{cyc} - 40$ (ns)
$t_{WSW2} = 1.5 \times t_{cyc} - 40$ (ns)
$t_{PCH} = 1.0 \times t_{cyc} - 40$ (ns)

For Condition C, the following times depend on the clock cycle time as shown below.

$t_{ACC1} = 1.5 \times t_{cyc} - 34$ (ns)
$t_{ACC2} = 2.5 \times t_{cyc} - 34$ (ns)
$t_{ACC3} = 1.0 \times t_{cyc} - 36$ (ns)
$t_{ACC4} = 2.0 \times t_{cyc} - 31$ (ns)
$t_{WSW1} = 1.0 \times t_{cyc} - 24$ (ns)
$t_{WSW2} = 1.5 \times t_{cyc} - 22$ (ns)
$t_{PCH} = 1.0 \times t_{cyc} - 21$ (ns)

Table 18.5 Control Signal Timing

Condition A: V_{CC} = 2.7 V to 5.5 V, AV_{CC} = 2.7 V to 5.5 V, V_{SS} = AV_{SS} = 0 V, ϕ = 2 MHz to 8 MHz, T_a = –20°C to +75°C (regular specifications), T_a = –40°C to +85°C (wide-range specifications)

Condition B: V_{CC} = 3.0 V to 5.5 V, AV_{CC} = 3.0 V to 5.5 V, V_{SS} = AV_{SS} = 0 V, ϕ = 2 MHz to 10 MHz, T_a = –20°C to +75°C (regular specifications), T_a = –40°C to +85°C (wide-range specifications)

Condition C: V_{CC} = 5.0 V ±10%, AV_{CC} = 5.0 V ±10%, V_{SS} = AV_{SS} = 0 V, ϕ = 2 MHz to 18 MHz, T_a = –20°C to +75°C (regular specifications), T_a = –40°C to +85°C (wide-range specifications)

		Condition A		Condition B		Condition C			
		8 MHz		10 MHz		18 MHz			
Item	**Symbol**	**Min**	**Max**	**Min**	**Max**	**Min**	**Max**	**Unit**	**Test Conditions**
$\overline{RES}$ setup time	t_{RESS}	200	—	200	—	200	—	ns	Figure 18.10
$\overline{RES}$ pulse width	t_{RESW}	10	—	10	—	10	—	tcyc	
Mode programming setup time (MD_0, MD_1, MD_2)	t_{MDS}	200	—	200	—	200	—	ns	
$\overline{RESO}$ output delay time	t_{RESD}	—	100	—	100	—	100	ns	Figure 18.11
$\overline{RESO}$ output pulse width	t_{RESOW}	132	—	132	—	132	—	tcyc	
NMI setup time (NMI, $\overline{IRQ_0}$, $\overline{IRQ_1}$, $\overline{IRQ_4}$, $\overline{IRQ_5}$)	t_{NMIS}	200	—	200	—	150	—	ns	Figure 18.12
NMI hold time (NMI, $\overline{IRQ_0}$, $\overline{IRQ_1}$, $\overline{IRQ_4}$, $\overline{IRQ_5}$)	t_{NMIH}	10	—	10	—	10	—		
Interrupt pulse width (NMI, $\overline{IRQ_1}$, $\overline{IRQ_0}$ when exiting software standby mode)	t_{NMIW}	200	—	200	—	200	—		
Clock oscillator settling time at reset (crystal)	t_{OSC1}	20	—	20	—	20	—	ms	Figure 18.13
Clock oscillator settling time in software standby (crystal)	t_{OSC2}	8	—	8	—	7	—	ms	Figure 17.1

Table 18.6 Timing of On-Chip Supporting Modules

Condition A: V_{CC} = 2.7 V to 5.5 V, AV_{CC} = 2.7 V to 5.5 V, V_{SS} = AV_{SS} = 0 V, ϕ = 2 MHz to 8 MHz, T_a = –20°C to +75°C (regular specifications), T_a = –40°C to +85°C (wide-range specifications)

Condition B: V_{CC} = 3.0 V to 5.5 V, AV_{CC} = 3.0 V to 5.5 V, V_{SS} = AV_{SS} = 0 V, ϕ = 2 MHz to 10 MHz, T_a = –20°C to +75°C (regular specifications), T_a = –40°C to +85°C (wide-range specifications)

Condition C: V_{CC} = 5.0 V ±10%, AV_{CC} = 5.0 V ±10%, V_{SS} = AV_{SS} = 0 V, ϕ = 2 MHz to 18 MHz, T_a = –20°C to +75°C (regular specifications), T_a = –40°C to +85°C (wide-range specifications)

				Condition A		Condition B		Condition C			
				8 MHz		10 MHz		18 MHz			
Item			**Symbol**	**Min**	**Max**	**Min**	**Max**	**Min**	**Max**	**Unit**	**Test Conditions**
ITU	Timer output delay time		t_{TOCD}	—	100	—	100	—	100	ns	Figure 18.15
	Timer input setup time		t_{TICS}	50	—	50	—	50	—		
	Timer clock input setup time		t_{TCKS}	50	—	50	—	50	—		Figure 18.16
	Timer clock pulse width	Single edge	t_{TCKWH}	1.5	—	1.5	—	1.5	—	t_{cyc}	
		Both edges	t_{TCKWL}	2.5	—	2.5	—	2.5	—		
SCI	Input clock cycle	Asynchronous	t_{Scyc}	4	—	4	—	4	—		Figure 18.17
		Synchronous		6	—	6	—	6	—		
	Input clock rise time		t_{SCKr}	—	1.5	—	1.5	—	1.5		
	Input clock fall time		t_{SCKf}	—	1.5	—	1.5	—	1.5		
	Input clock pulse width		t_{SCKW}	0.4	0.6	0.4	0.6	0.4	0.6	t_{Scyc}	

Item		Symbol	Condition A 8 MHz Min	Condition A 8 MHz Max	Condition B 10 MHz Min	Condition B 10 MHz Max	Condition C 18 MHz Min	Condition C 18 MHz Max	Unit	Test Conditions
SCI	Transmit data delay time	t_{TXD}	—	100	—	100	—	100	ns	Figure 18.18
	Receive data setup time (synchronous)	t_{RXS}	100	—	100	—	100	—		
	Receive data hold time (synchronous clock input)	t_{RXH}	100	—	100	—	100	—		
	Receive data hold time (synchronous clock output)		0	—	0	—	0	—		
Ports and TPC	Output data delay time	t_{PWD}	—	100	—	100	—	100	ns	Figure 18.14
	Input data setup time	t_{PRS}	50	—	50	—	50	—		
	Input data hold time	t_{PRH}	50	—	50	—	50	—		

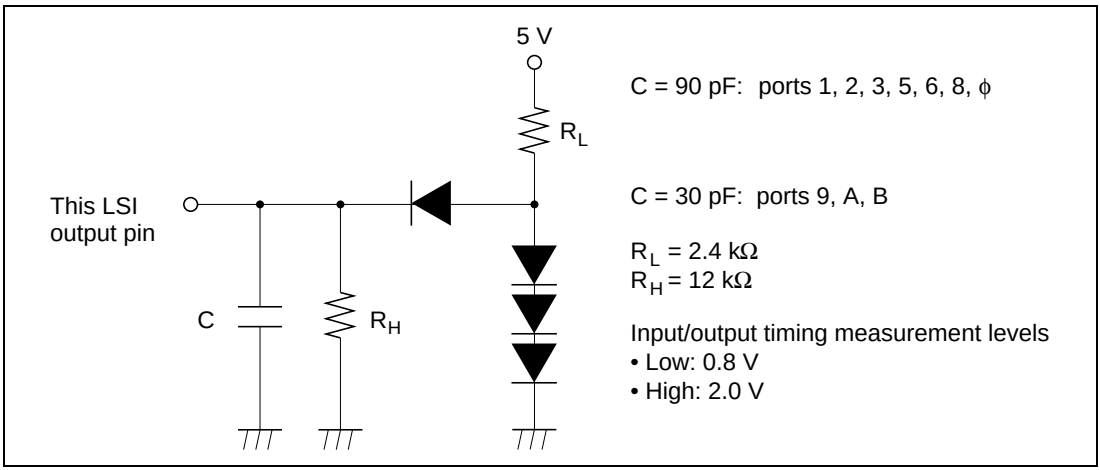


Figure 18.3 Output Load Circuit

18.1.4 A/D Conversion Characteristics

Table 18.7 lists the A/D conversion characteristics.

Table 18.7 A/D Converter Characteristics

Condition A: V_{CC} = 2.7 V to 5.5 V, AV_{CC} = 2.7 V to 5.5 V, V_{SS} = AV_{SS} = 0 V, ϕ = 2 MHz to 8 MHz, T_a = –20°C to +75°C (regular specifications), T_a = –40°C to +85°C (wide-range specifications)

Condition B: V_{CC} = 3.0 V to 5.5 V, AV_{CC} = 3.0 V to 5.5 V, V_{SS} = AV_{SS} = 0 V, ϕ = 2 MHz to 10 MHz, T_a = –20°C to +75°C (regular specifications), T_a = –40°C to +85°C (wide-range specifications)

Condition C: V_{CC} = 5.0 V ±10%, AV_{CC} = 5.0 V ±10%, V_{SS} = AV_{SS} = 0 V, ϕ = 2 MHz to 18 MHz, T_a = –20°C to +75°C (regular specifications), T_a = –40°C to +85°C (wide-range specifications)

	Condition A			Condition B			Condition C			
	8 MHz			10 MHz			18 MHz			
Item	**Min**	**Typ**	**Max**	**Min**	**Typ**	**Max**	**Min**	**Typ**	**Max**	**Unit**
Resolution	10	10	10	10	10	10	10	10	10	bits
Conversion time	—	—	16.8	—	—	13.4	—	—	7.5	µs
Analog input capacitance	—	—	20	—	—	20	—	—	20	pF
Permissible signal-source impedance	—	—	10*[1]	—	—	10*[1]	—	—	10*[4]	kΩ
	—	—	5*[2]	—	—	5*[3]	—	—	5*[5]	
Nonlinearity error	—	—	±7.5	—	—	±7.5	—	—	±3.5	LSB
Offset error	—	—	±7.5	—	—	±7.5	—	—	±3.5	LSB
Full-scale error	—	—	±7.5	—	—	±7.5	—	—	±3.5	LSB
Quantization error	—	—	±0.5	—	—	±0.5	—	—	±0.5	LSB
Absolute accuracy	—	—	±8.0	—	—	±8.0	—	—	±4.0	LSB

Notes: 1. The value is for 4.0 V ≤ AV_{CC} ≤ 5.5 V.
2. The value is for 2.7 V ≤ AV_{CC} < 4.0 V.
3. The value is for 3.0 V ≤ AV_{CC} < 4.0 V.
4. The value is for ϕ ≤ 12 MHz.
5. The value is for ϕ > 12 MHz.

18.2 Electrical Characteristics of Flash Memory Version

18.2.1 Absolute Maximum Ratings

Table 18.8 lists the absolute maximum ratings.

Table 18.8 Absolute Maximum Ratings

Item	Symbol	Value	Unit
Power supply voltage	V_{CC}	–0.3 to +7.0	V
Input voltage (except port 7)*[1]	V_{in}	–0.3 to V_{CC} +0.3	V
Input voltage (port 7)	V_{in}	–0.3 to AV_{CC} +0.3	V
Analog power supply voltage	AV_{CC}	–0.3 to +7.0	V
Analog input voltage	V_{AN}	–0.3 to AV_{CC} +0.3	V
Operating temperature	T_{opr}	Regular specifications: –20 to +75*[2]	°C
		Wide-range specifications: –40 to +85*[2]	°C
Storage temperature	T_{stg}	–55 to +125	°C

Caution: Permanent damage to the chip may result if absolute maximum ratings are exceeded.

Notes: 1. 12 V must not be applied to any pin, as this will cause permanent damage to the chip.

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

18.2.2 DC Characteristics

Table 18.9 lists the DC characteristics. Table 18.10 lists the permissible output currents.

Table 18.9 DC Characteristics (1)

Conditions: V_{CC} = 5.0 V ±10%, AV_{CC} = 5.0 V ±10%, V_{SS} = AV_{SS} = 0 V*[1], T_a = –20°C to +75°C (regular specifications), T_a = –40°C to +85°C (wide-range specifications)
(Programming/Erasing Conditions: T_a = 0°C to +75°C (regular specifications), T_a = 0°C to +85°C (wide-range specifications))

Item		Symbol	Min	Typ	Max	Unit	Test Conditions
Schmitt trigger input voltages	Port A, $P8_0$ to $P8_1$, PB_0 to PB_3	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, MD_1, MD_0, FWE	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, 2, 3, 5, 6, 9, PB_4, PB_5, PB_7		2.0	—	V_{CC} +0.3	V	
Input low voltage	$\overline{RES}$, $\overline{STBY}$, MD_2, MD_1, MD_0, FWE	V_{IL}	–0.3	—	0.5	V	
	NMI, EXTAL, ports 1, 2, 3, 5, 6, 7, 9, PB_4, PB_5, PB_7		–0.3	—	0.8	V	
Output high voltage	All output pins	V_{OH}	V_{CC} –0.5	—	—	V	I_{OH} = –200 µA
			3.5	—	—	V	I_{OH} = –1 mA
Output low voltage	All output pins	V_{OL}	—	—	0.4	V	I_{OL} = 1.6 mA
	Ports 1, 2, 5, B		—	—	1.0	V	I_{OL} = 10 mA

Item		Symbol	Min	Typ	Max	Unit	Test Conditions
Input leakage current	$\overline{STBY}$, NMI, $\overline{RES}$, MD_2, MD_1, MD_0	$\lvert I_{in} \rvert$	—	—	1.0	µA	V_{in} = 0.5 V to V_{CC} –0.5 V
	Port 7		—	—	1.0	µA	V_{in} = 0.5 V to AV_{CC} –0.5 V
	FWE		—	—	10		
Three-state leakage current (off state)	Ports 1, 2, 3, 5, 6, 8, 9, A, B	$\lvert I_{TSI} \rvert$	—	—	1.0	µA	V_{in} = 0.5 V to V_{CC} –0.5 V
Input pull-up current	Ports 2, 5	$-I_p$	50	—	300	µA	V_{in} = 0 V
Input capacitance	NMI, $\overline{RES}$	C_{in}	—	—	50	pF	V_{in} = 0 V, f = 1 MHz, T_a = 25°C
	All input pins except NMI and $\overline{RES}$		—	—	20		
Current dissipation *2 *4	Normal operation	I_{CC}	—	50	70	mA	f = 18 MHz
	Sleep mode		—	35	50		f = 18 MHz
	Standby mode*3		—	0.01	5.0	µA	$T_a \leq$ 50°C
			—	—	20.0		50°C < T_a
Analog power supply current	During A/D conversion	AI_{CC}	—	1.7	2.8	mA	
	Idle		—	0.02	10.0	µA	
RAM standby voltage		V_{RAM}	2.0	—	—	V	

Notes: 1. If the A/D converter is not used, do not leave the AV_{CC} and AV_{SS} pins open. Connect AV_{CC} 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 with all 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 V.

4. Power supply current value when programming/erasing in flash memory (T_a = 0°C to +75°C (regular specifications), T_a = 0°C to +85°C (wide-range specifications)) is 20 mA (max) higher than the power supply current value in normal operation.

Table 18.9 DC Characteristics (2)

Conditions: V_{CC} = 3.0 V to 5.5 V, AV_{CC} = 3.0 V to 5.5 V, V_{SS} = AV_{SS} = 0 V*[1], T_a = –20°C to +75°C (regular specifications), T_a = –40°C to +85°C (wide-range specifications)
(Programming/Erasing Conditions: V_{CC} = 3.0 V to 3.6 V, T_a = 0°C to +75°C (regular specifications), T_a = 0°C to +85°C (wide-range specifications))

Item		Symbol	Min	Typ	Max	Unit	Test Conditions
Schmitt trigger input voltages	Port A, $P8_0$ to $P8_1$, PB_0 to PB_3	V_T^-	$V_{CC} \times 0.2$	—	—	V	
		V_T^+	—	—	$V_{CC} \times 0.7$	V	
		$V_T^+ - V_T^-$	$V_{CC} \times 0.04$	—	—	V	
Input high voltage	$\overline{RES}$, $\overline{STBY}$, NMI, MD_2 to MD_0, FWE	V_{IH}	$V_{CC} \times 0.9$	—	V_{CC} +0.3	V	
	EXTAL		$V_{CC} \times 0.7$	—	V_{CC} +0.3	V	
	Port 7		$V_{CC} \times 0.7$	—	AV_{CC} +0.3	V	
	Ports 1 to 3, 5, 6, 9, PB_4, PB_5, PB_7		$V_{CC} \times 0.7$	—	V_{CC} +0.3	V	
Input low voltage	$\overline{RES}$, $\overline{STBY}$, FWE, MD_2 to MD_0, FWE	V_{IL}	–0.3	—	$V_{CC} \times 0.1$	V	
	NMI, EXTAL, ports 1 to 3, 5 to 7, 9, PB_4, PB_5, PB_7		–0.3	—	$V_{CC} \times 0.2$	V	V_{CC} < 4.0 V
					0.8		V_{CC} = 4.0 V to 5.5V
Output high voltage	All output pins	V_{OH}	V_{CC} –0.5	—	—	V	I_{OH} = –200 µA
			V_{CC} –1.0	—	—	V	I_{OH} = –1 mA
Output low voltage	All output pins	V_{OL}	—	—	0.4	V	I_{OL} = 1.0 mA
	Ports 1, 2, 5, B		—	—	1.0	V	$V_{CC} \leq 4$ V, I_{OL} = 5 mA, 4V < $V_{CC} \leq$ 5.5 V, I_{OL} = 10 mA

Item		Symbol	Min	Typ	Max	Unit	Test Conditions
Input leakage current	$\overline{STBY}$, NMI, $\overline{RES}$, MD_2, MD_1, MD_0	$\|I_{in}\|$	—	—	1.0	µA	V_{in} = 0.5 V to V_{CC} –0.5 V
	Port 7		—	—	1.0	µA	V_{in} = 0.5 V to AV_{CC} –0.5 V
	FWE		—	—	10	µA	V_{in} = 0.5 V to V_{CC} –0.5 V
Three-state leakage current (off state)	Ports 1, 2, 3, 5, 6, 8 to B	$\|I_{TSI}\|$	—	—	1.0	µA	V_{in} = 0.5 V to V_{CC} –0.5 V
Input pull-up current	Ports 2 and 5	$-I_p$	10	—	300	µA	V_{CC} = 3.0 V to 5.5 V, V_{in} = 0 V
Input capacitance	NMI, $\overline{RES}$	C_{in}	—	—	50	pF	V_{in} = 0 V, f = 1 MHz, T_a = 25°C
	All input pins except NMI and $\overline{RES}$		—	—	20	pF	
Current dissipation *2 *5	Normal operation	I_{CC} *4	—	15 (3.0 V)	41.5 (5.5 V)	mA	f = 10 MHz
	Sleep mode		—	10 (3.0 V)	30.5 (5.5 V)	mA	f = 10 MHz
	Standby mode*3		—	0.01	5.0	µA	$T_a \leq$ 50°C
			—	—	20.0		50°C < T_a
Analog power supply current	During A/D conversion	AI_{CC}	—	1.3	2.5	mA	AV_{CC} = 3.0 V
			—	1.7	2.8		AV_{CC} = 5.0 V
	Idel		—	0.02	10.0	µA	
RAM standby voltage		V_{RAM}	2.0	—	—	V	

Notes: 1. If the A/D converter is not used, do not leave the AV_{CC} and AV_{SS} pins open. Connect AV_{CC} to V_{CC}, and connect AV_{SS} to V_{SS}.

2. Current dissipation values are for V_{IH} min = VCC –0.5 V and V_{IL} max = 0.5 V with all 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 = VCC × 0.9, and V_{IL} max = 0.3 V.

4. I_{CC} depends on V_{CC} and f as follows:
   I_{CC} max = 3.0 (mA) + 0.7 (mA/MHz · V) × V_{CC} × f [normal mode]
   I_{CC} max = 3.0 (mA) + 0.5 (mA/MHz · V) × V_{CC} × f [sleep mode]

5. The current dissipation value when programming/erasing flash memory (T_a = 0°C to +75°C (regular specifications), T_a = 0°C to +85°C (wide-range specifications)) is 20 mA (max) higher than the current dissipation value in normal operation.

Table 18.10 Permissible Output Currents

Conditions: V_{CC} = 3.0 V to 5.5 V, AV_{CC} = 3.0 V to 5.5 V, V_{SS} = AV_{SS} = 0 V, T_a = –20°C to +75°C (regular specifications), T_a = –40°C to +85°C (wide-range specifications)

Item		Symbol	Min	Typ	Max	Unit
Permissible output low current (per pin)	Ports 1, 2, 5 and B	I_{OL}	—	—	10	mA
	Other output pins		—	—	2.0	mA
Permissible output low current (total)	Total of 27 pins including ports 1, 2, 5 and B	ΣI_{OL}	—	—	80	mA
	Total of 23 pins, including ports 8, 9, A and B		—	—	75*[2]/ 65*[1]	mA
	Total of all output pins, including the above		—	—	120	mA
Permissible output high current (per pin)	All output pins	I_{OH}	—	—	2.0	mA
Permissible output high current (total)	Total of all output pins	ΣI_{OH}	—	—	40	mA

Notes: To protect chip reliability, do not exceed the output current values in table 18.10.

When driving a Darlington pair or LED, always insert a current-limiting resistor in the output line, as shown in figures 18.4 and 18.5.

1. Conditions: V_{CC} = 3.0 V to 5.5 V, AV_{CC} = 3.0 V to 5.5 V
2. Conditions: V_{CC} = 4.5 V to 5.5 V, AV_{CC} = 4.5 V to 5.5 V

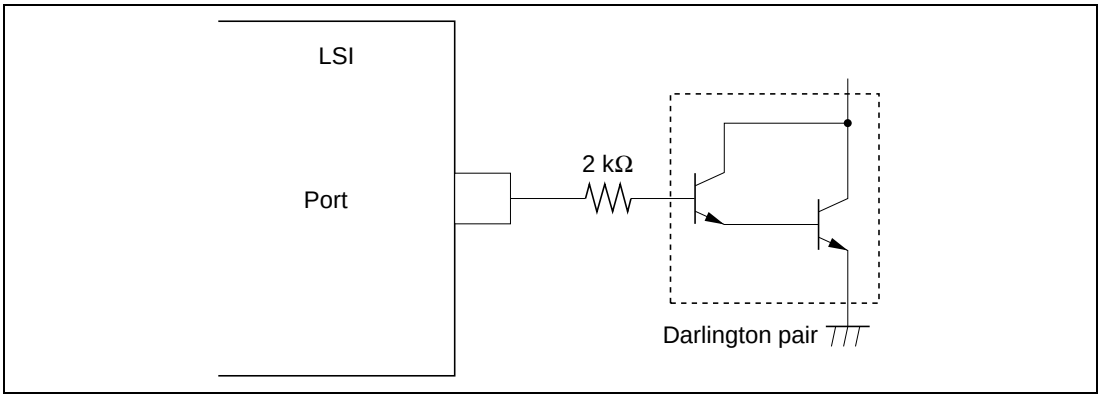


Figure 18.4 Darlington Pair Drive Circuit (Example)

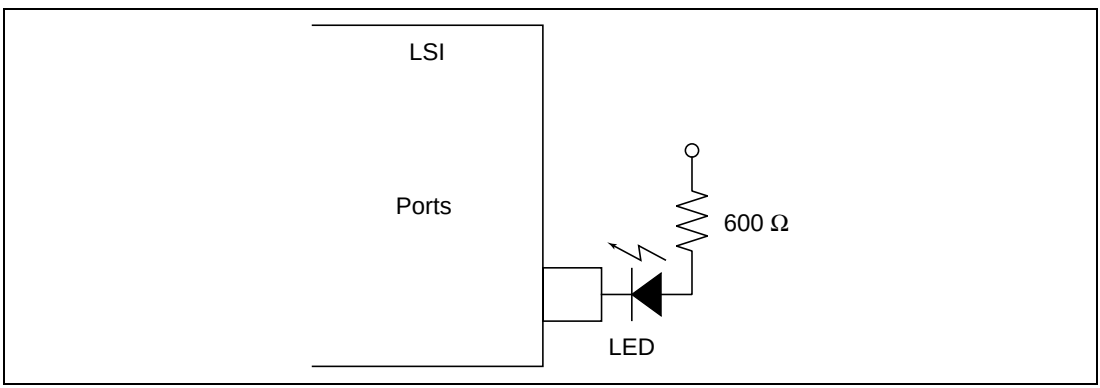


Figure 18.5 LED Drive Circuit (Example)

18.2.3 AC Characteristics

Bus timing parameters are listed in table 18.11. Control signal timing parameters are listed in table 18.12. Timing parameters of the on-chip supporting modules are listed in table 18.13.

Table 18.11 Bus Timing

Condition A: V_{CC} = 3.0 V to 5.5 V, AV_{CC} = 3.0 V to 5.5 V, V_{SS} = AV_{SS} = 0 V, ϕ = 2 to 10 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_{SS} = AV_{SS} = 0 V, ϕ = 2 MHz to 18 MHz, T_a = –20°C to +75°C (regular specifications), T_a = –40°C to +85°C (wide-range specifications)

		Condition A		Condition B			
		10 MHz		18 MHz			
Item	Symbol	Min	Max	Min	Max	Unit	Test Conditions
Clock cycle time	t_{cyc}	100	500	55.5	500	ns	Figure 18.7, Figure 18.8
Clock low pulse width	t_{CL}	30	—	17	—		
Clock high pulse width	t_{CH}	30	—	17	—		
Clock rise time	t_{Cr}	—	15	—	10		
Clock fall time	t_{Cf}	—	15	—	10		
Address delay time	t_{AD}	—	50	—	25		
Address hold time	t_{AH}	20	—	10	—		
Address strobe delay time	t_{ASD}	—	40	—	25		
Write strobe delay time	t_{WSD}	—	50	—	25		
Strobe delay time	t_{SD}	—	50	—	25		
Write data strobe pulse width 1	t_{WSW1}*	60	—	32	—		
Write data strobe pulse width 2	t_{WSW2}*	110	—	62	—		
Address setup time 1	t_{AS1}	15	—	10	—		
Address setup time 2	t_{AS2}	65	—	38	—		
Read data setup time	t_{RDS}	35	—	15	—		
Read data hold time	t_{RDH}	0	—	0	—		

Item	Symbol	Condition A 10 MHz Min	Condition A 10 MHz Max	Condition B 18 MHz Min	Condition B 18 MHz Max	Unit	Test Conditions
Write data delay time	t_{WDD}	—	75	—	55	ns	Figure 18.7, Figure 18.8
Write data setup time 1	t_{WDS1}	40	—	10	—		
Write data setup time 2	t_{WDS2}	–10	—	–10	—		
Write data hold time	t_{WDH}	20	—	20	—		
Read data access time 1	t_{ACC1}*	—	100	—	50		
Read data access time 2	t_{ACC2}*	—	200	—	105		
Read data access time 3	t_{ACC3}*	—	50	—	20		
Read data access time 4	t_{ACC4}*	—	150	—	80		
Precharge time	t_{PCH}*	60	—	40	—		
Wait setup time	t_{WTS}	40	—	25	—	ns	Figure 18.9
Wait hold time	t_{WTH}	10	—	5	—		

Note: * For condition A, the following times depend on the clock cycle time as shown below.

$t_{ACC1} = 1.5 \times t_{cyc} - 50$ (ns) $t_{WSW1} = 1.0 \times t_{cyc} - 40$ (ns)
$t_{ACC2} = 2.5 \times t_{cyc} - 50$ (ns) $t_{WSW2} = 1.5 \times t_{cyc} - 40$ (ns)
$t_{ACC3} = 1.0 \times t_{cyc} - 50$ (ns) $t_{PCH} = 1.0 \times t_{cyc} - 40$ (ns)
$t_{ACC4} = 2.0 \times t_{cyc} - 50$ (ns)

For condition B, the following times depend on the clock cycle time as shown below.

$t_{ACC1} = 1.5 \times t_{cyc} - 34$ (ns) $t_{WSW1} = 1.0 \times t_{cyc} - 24$ (ns)
$t_{ACC2} = 2.5 \times t_{cyc} - 34$ (ns) $t_{WSW2} = 1.5 \times t_{cyc} - 22$ (ns)
$t_{ACC3} = 1.0 \times t_{cyc} - 36$ (ns) $t_{PCH} = 1.0 \times t_{cyc} - 21$ (ns)
$t_{ACC4} = 2.0 \times t_{cyc} - 31$ (ns)

Table 18.12 Control Signal Timing

Condition A: V_{CC} = 3.0 V to 5.5 V, AV_{CC} = 3.0 V to 5.5 V, V_{SS} = AV_{SS} = 0 V, ϕ = 2 to 10 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_{SS} = AV_{SS} = 0 V, ϕ = 2 MHz to 18 MHz, T_a = –20°C to +75°C (regular specifications), T_a = –40°C to +85°C (wide-range specifications)

		Condition A		Condition B			
		10 MHz		18 MHz			
Item	**Symbol**	**Min**	**Max**	**Min**	**Max**	**Unit**	**Test Conditions**
$\overline{RES}$ setup time	t_{RESS}	200	—	200	—	ns	Figure 18.10
$\overline{RES}$ pulse width	t_{RESW}	20	—	20	—	tcyc	
Mode programming setup time	t_{MDS}	200	—	200	—	ns	
NMI setup time (NMI, $\overline{IRQ_0}$, $\overline{IRQ_1}$, $\overline{IRQ_4}$, $\overline{IRQ_5}$)	t_{NMIS}	200	—	150	—	ns	Figure 18.12
NMI hold time (NMI, $\overline{IRQ_0}$, $\overline{IRQ_1}$, $\overline{IRQ_4}$, $\overline{IRQ_5}$)	t_{NMIH}	10	—	10	—		
Interrupt pulse width (NMI, $\overline{IRQ_1}$, $\overline{IRQ_0}$ when exiting software standby mode)	t_{NMIW}	200	—	200	—		
Clock oscillator settling time at reset (crystal)	t_{OSC1}	20	—	20	—	ms	Figure 18.13
Clock oscillator settling time in software standby (crystal)	t_{OSC2}	8	—	7	—	ms	Figure 17.1

Table 18.13 Timing of On-Chip Supporting Modules

Condition A: V_{CC} = 3.0 V to 5.5 V, AV_{CC} = 3.0 V to 5.5 V, V_{SS} = AV_{SS} = 0 V, ϕ = 2 MHz to 10 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_{SS} = AV_{SS} = 0 V, ϕ = 2 MHz to 18 MHz, T_a = –20°C to +75°C (regular specifications), T_a = –40°C to +85°C (wide-range specifications)

				Condition A		Condition B			
				10 MHz		18 MHz			
Item			Symbol	Min	Max	Min	Max	Unit	Test Conditions
ITU	Timer output delay time		t_{TOCD}	—	100	—	100	ns	Figure 18.15
	Timer input setup time		t_{TICS}	50	—	50	—		
	Timer clock input setup time		t_{TCKS}	50	—	50	—		Figure 18.16
	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	—		Figure 18.17
		Synchronous		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	Figure 18.18
	Receive data setup time (synchronous)		t_{RXS}	100	—	100	—		
	Receive data hold time (synchronous clock input)		t_{RXH}	100	—	100	—		
	Receive data hold time (synchronous clock output)			0	—	0	—		

Item		Symbol	Condition A 10 MHz Min	Condition A 10 MHz Max	Condition B 18 MHz Min	Condition B 18 MHz Max	Unit	Test Conditions
Ports and TPC	Output data delay time	t_{PWD}	—	100	—	100	ns	Figure 18.14
	Input data setup time	t_{PRS}	50	—	50	—		
	Input data hold time	t_{PRH}	50	—	50	—		

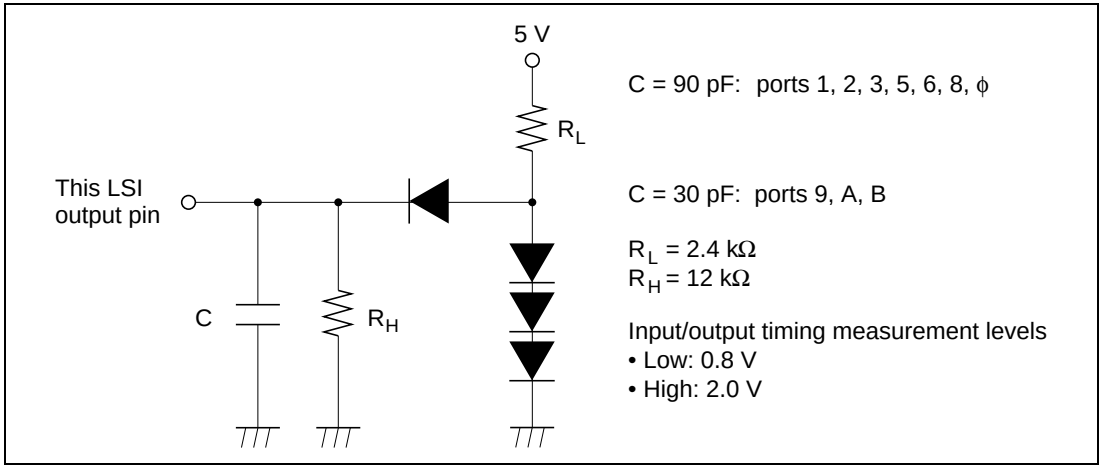


Figure 18.6 Output Load Circuit

18.2.4 A/D Conversion Characteristics

Table 18.14 lists the A/D conversion characteristics.

Table 18.14 A/D Converter Characteristics

Condition A: V_{CC} = 3.0 V to 5.5 V, AV_{CC} = 3.0 V to 5.5 V, V_{SS} = AV_{SS} = 0 V, ϕ = 2 MHz to 10 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_{SS} = AV_{SS} = 0 V, ϕ = 2 MHz to 18 MHz, T_a = –20°C to +75°C (regular specifications), T_a = –40°C to +85°C (wide-range specifications)

	Condition A			Condition B			
	10 MHz			18 MHz			
Item	**Min**	**Typ**	**Max**	**Min**	**Typ**	**Max**	**Unit**
Resolution	10	10	10	10	10	10	bits
Conversion time	—	—	13.4	—	—	7.5	µs
Analog input capacitance	—	—	20	—	—	20	pF
Permissible signal-source impedance	—	—	5*[1]	—	—	10*[2]	kΩ
				—	—	5*[3]	
Nonlinearity error	—	—	±7.5	—	—	±3.5	LSB
Offset error	—	—	±7.5	—	—	±3.5	LSB
Full-scale error	—	—	±7.5	—	—	±3.5	LSB
Quantization error	—	—	±0.5	—	—	±0.5	LSB
Absolute accuracy	—	—	±8.0	—	—	±4.0	LSB

Notes: 1. The value is for ϕ = 10 MHz.
2. The value is for $\phi \leq$ 12 MHz.
3. The value is for $\phi >$ 12 MHz.

18.2.5 Flash Memory Characteristics

Table 18.15 shows the flash memory characteristics.

Table 18.15 Flash Memory Characteristics (1)

Conditions: V_{CC} = 4.5 V to 5.5 V, AV_{CC} = 4.5 V to 5.5 V, V_{SS} = AV_{SS} = 0 V
T_a = 0°C to +75°C (program/erase operating temperature range: regular specifications), T_a = 0°C to +85°C (program/erase operating temperature range: wide-range specifications)

Item		Symbol	Min	Typ	Max	Unit	Test condition
Programming time*1 *2 *4		t_P	—	10	200	ms/32 bytes	
Erase time*1 *3 *5		t_E	—	100	300	ms/block	
Reprogramming count		N_{WEC}	—	—	100	Times	
Programming	Wait time after SWE bit setting*1	x	10	—	—	µs	
	Wait time after PSU bit setting*1	y	50	—	—	µs	
	Wait time after P bit setting*1 *4	z	150	—	500	µs	
	Wait time after P bit clear*1	α	10	—	—	µs	
	Wait time after PSU bit clear*1	β	10	—	—	µs	
	Wait time after PV bit setting*1	γ	4	—	—	µs	
	Wait time after H'FF dummy write*1	ε	2	—	—	µs	
	Wait time after PV bit clear*1	η	4	—	—	µs	
	Maximum programming count*1 *4	N	—	—	403	Times	
Erase	Wait time after SWE bit setting*1	x	10	—	—	µs	
	Wait time after ESU bit setting*1	y	200	—	—	µs	
	Wait time after E bit setting*1 *5	z	5	—	10	ms	
	Wait time after E bit clear*1	α	10	—	—	µs	
	Wait time after ESU bit clear*1	β	10	—	—	µs	
	Wait time after EV bit setting*1	γ	20	—	—	µs	
	Wait time after H'FF dummy write*1	ε	2	—	—	µs	
	Wait time after EV bit clear*1	η	5	—	—	µs	
	Maximum erase count*1 *5	N	30	—	60	Times	

Notes: 1. Set the times according to the program/erase algorithms.
2. Programming time per 32 bytes (Shows the total time the flash memory control register (FLMCR) is set. It does not include the programming verification time.)

3. Block erase time (Shows the period the E bit in FLMCR is set. It does not include the erase verification time.)
4. To specify the maximum programming time (t_P(max)) in the 32-byte programming flowchart, set the max value (403) for the maximum programming count (N).

   The wait time after P bit setting (z) should be changed as follows according to the programming counter value.

   Programming counter value of 1 to 4: z = 150 µs
   Programming counter value of 5 to 403: z = 500 µs

5. For the maximum erase time (t_E(max)), the following relationship applies between the wait time after E bit setting (z) and the maximum erase count (N):

   t_E(max) = Wait time after E bit setting (z) × maximum erase count (N)

   To set the maximum erase time, the values of z and N should be set so as to satisfy the above formula.

   Examples: When z = 5 [ms]: N = 60 times
   When z = 10 [ms]: N = 30 times

Table 18.15 Flash Memory Characteristics (2)

Conditions: V_{CC} = 3.0 V to 3.6 V, AV_{CC} = 3.0 V to 3.6 V, V_{SS} = AV_{SS} = 0 V
T_a = 0°C to +75°C (Programming/erasing operating temperature range: regular specification) T_a = 0°C to +85°C (Programming/erasing operating temperature range: wide-range specification)

Item		Symbol	Min	Typ	Max	Unit	Test condition
Programming time*[1] *[2] *[4]		t_P	—	10	200	ms/32 bytes	
Erase time*[1] *[3] *[5]		t_E	—	100	300	ms/block	
Reprogramming count		N_{WEC}	—	—	100	Times	
Programming	Wait time after SWE bit setting*[1]	x	10	—	—	µs	
	Wait time after PSU bit setting*[1]	y	50	—	—	µs	
	Wait time after P bit setting*[1] *[4]	z	150	—	500	µs	
	Wait time after P bit clear*[1]	α	10	—	—	µs	
	Wait time after PSU bit clear*[1]	β	10	—	—	µs	
	Wait time after PV bit setting*[1]	γ	4	—	—	µs	
	Wait time after H'FF dummy write*[1]	ε	2	—	—	µs	
	Wait time after PV bit clear*[1]	η	4	—	—	µs	
	Maximum programming count*[1] *[4]	N	—	—	403	Times	
Erase	Wait time after SWE bit setting*[1]	x	10	—	—	µs	
	Wait time after ESU bit setting*[1]	y	200	—	—	µs	
	Wait time after E bit setting*[1] *[5]	z	5	—	10	ms	
	Wait time after E bit clear*[1]	α	10	—	—	µs	
	Wait time after ESU bit clear*[1]	β	10	—	—	µs	
	Wait time after EV bit setting*[1]	γ	20	—	—	µs	
	Wait time after H'FF dummy write*[1]	ε	2	—	—	µs	
	Wait time after EV bit clear*[1]	η	5	—	—	µs	
	Maximum erase count*[1] *[5]	N	30	—	60	Times	

Notes: 1. Make each time setting in accordance with the program/program-verify flowchart or erase/erase-verify flowchart.
2. Programming time per 32 bytes (Shows the total period for which the P-bit in the flash memory control register (FLMCR) is set. It does not include the programming verification time.)
3. Block erase time (Shows the total period for which the E-bit in FLMCR is set. It does not include the erase verification time.)

4. To specify the maximum programming time (t_P(max)) in the 32-byte programming flowchart, set the maximum value (403) for the maximum programming count (N).

   The wait time after P bit setting (z) should be changed as follows according to the programming counter value.

   Programming counter value of 1 to 4: z = 150 µs
   Programming counter value of 5 to 403: z = 500 µs

5. For the maximum erase time (t_E(max)), the following relationship applies between the wait time after E bit setting (z) and the maximum erase count (N):

   t_E(max) = Wait time after E bit setting (z) × maximum erase count (N)

   To set the maximum erase time, the values of z and N should be set so as to satisfy the above formula.

   Examples: When z = 5 [ms], N = 60 times
   When z = 10 [ms], N = 30 times

18.3 Operational Timing

This section shows timing diagrams.

18.3.1 Bus Timing

Bus timing is shown as follows:

- Basic bus cycle: two-state access
  Figure 18.7 shows the timing of the external two-state access cycle.
- Basic bus cycle: three-state access
  Figure 18.8 shows the timing of the external three-state access cycle.
- Basic bus cycle: three-state access with one wait state
  Figure 18.9 shows the timing of the external three-state access cycle with one wait state inserted.

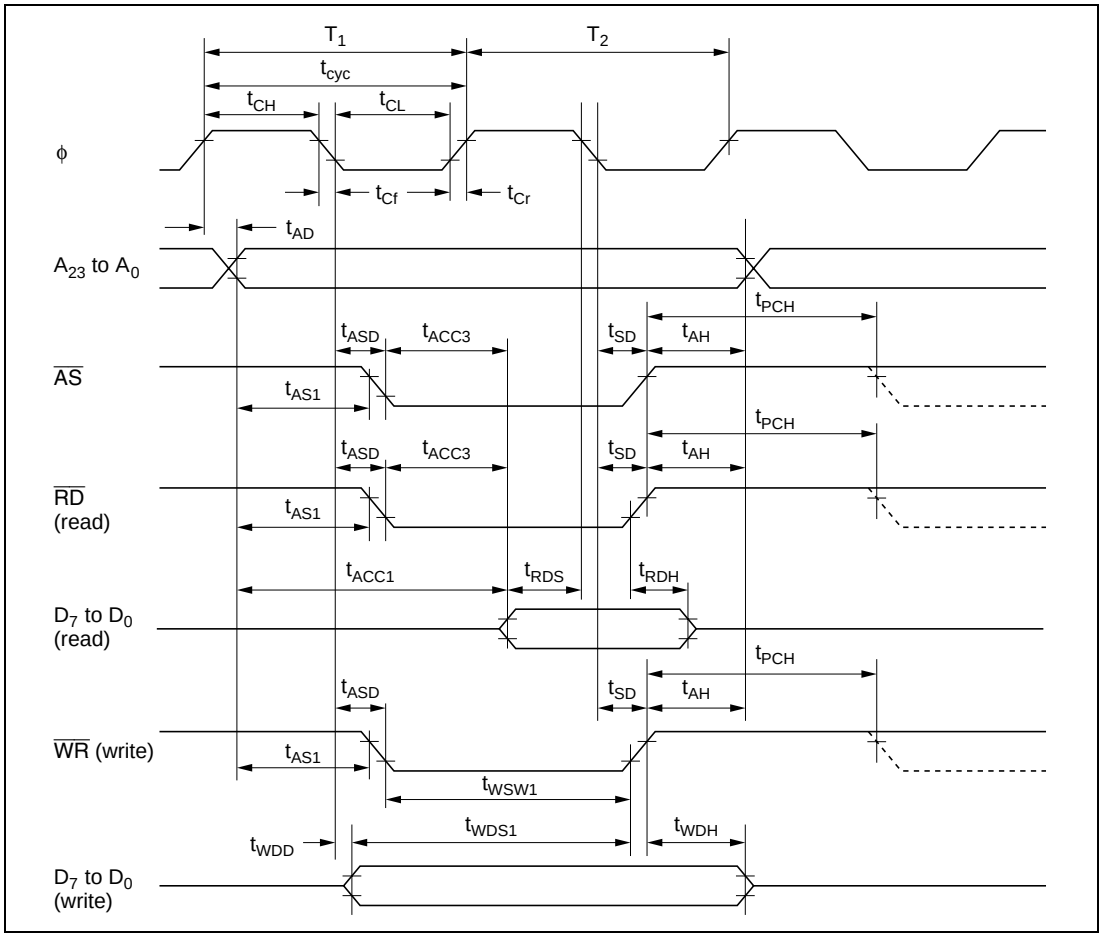


Figure 18.7 Basic Bus Cycle: Two-State Access

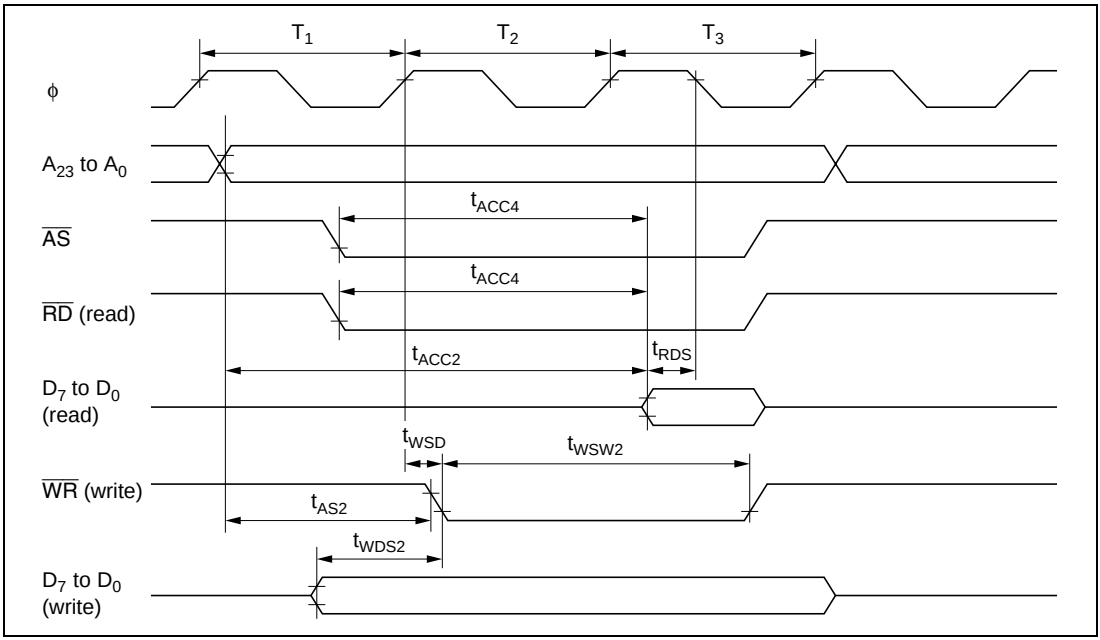


Figure 18.8 Basic Bus Cycle: Three-State Access

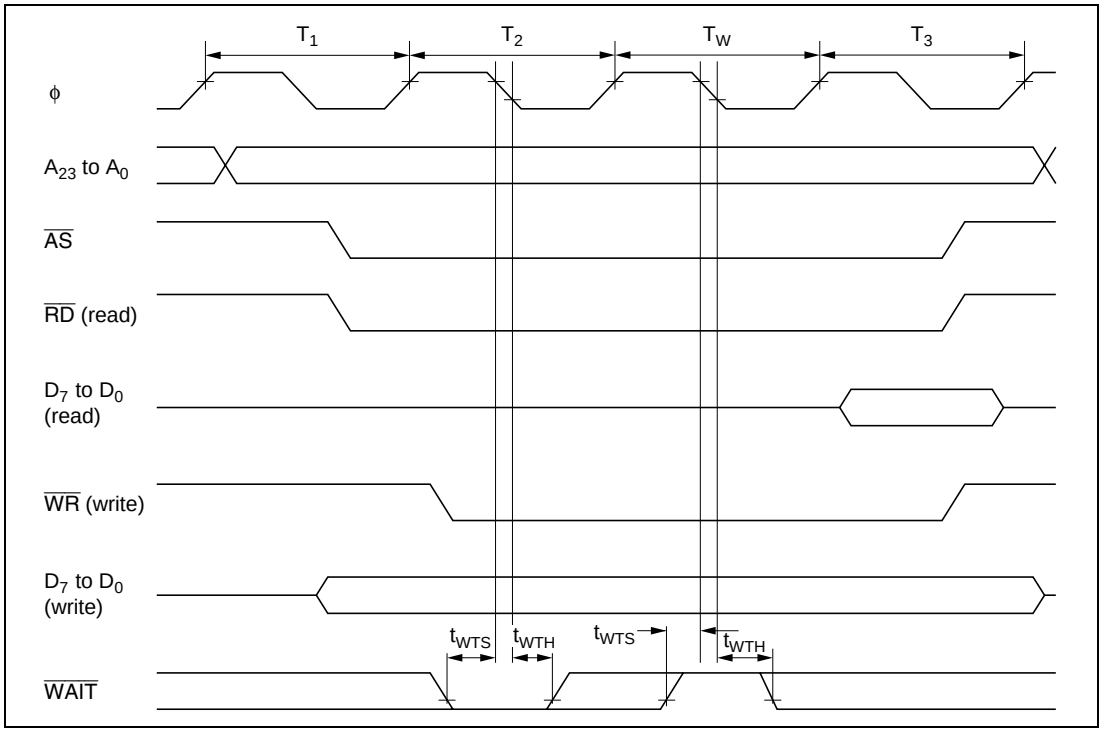
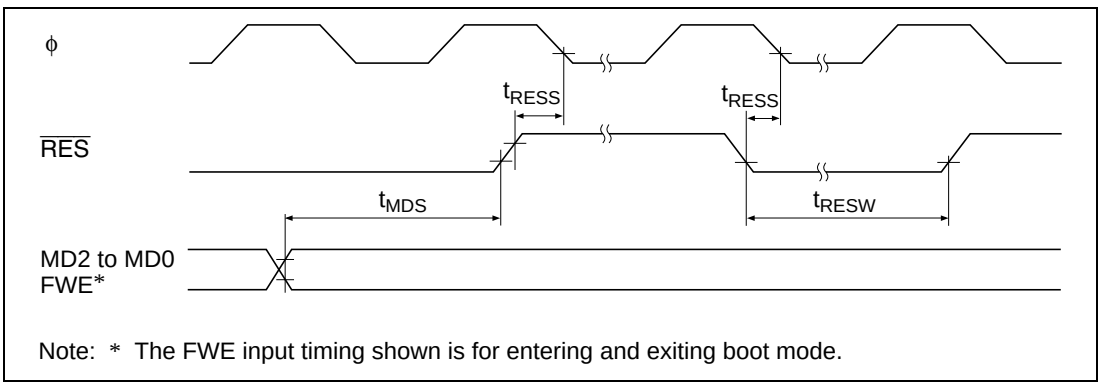


Figure 18.9 Basic Bus Cycle: Three-State Access with One Wait State

18.3.2 Control Signal Timing

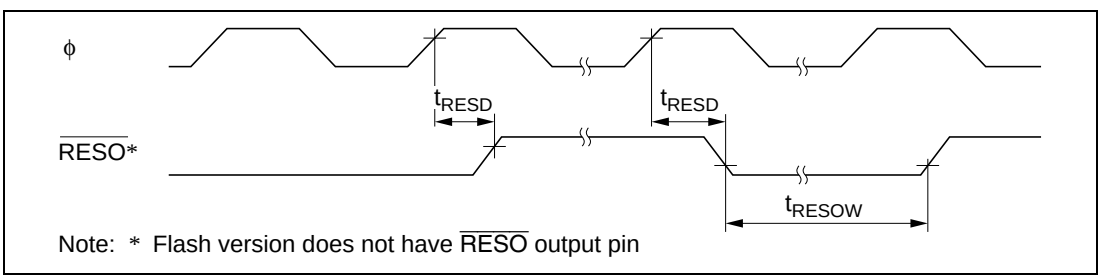
Control signal timing is shown as follows:

- Reset input timing
  Figure 18.10 shows the reset input timing.
- Reset output timing
  Figure 18.11 shows the reset output timing.
- Interrupt input timing
  Figure 18.12 shows the interrupt input timing for NMI and $\overline{IRQ_5}$, $\overline{IRQ_4}$, $\overline{IRQ_1}$, and $\overline{IRQ_0}$.



Note: * The FWE input timing shown is for entering and exiting boot mode.

Figure 18.10 Reset Input Timing



Note: * Flash version does not have $\overline{RESO}$ output pin

Figure 18.11 Reset Output Timing

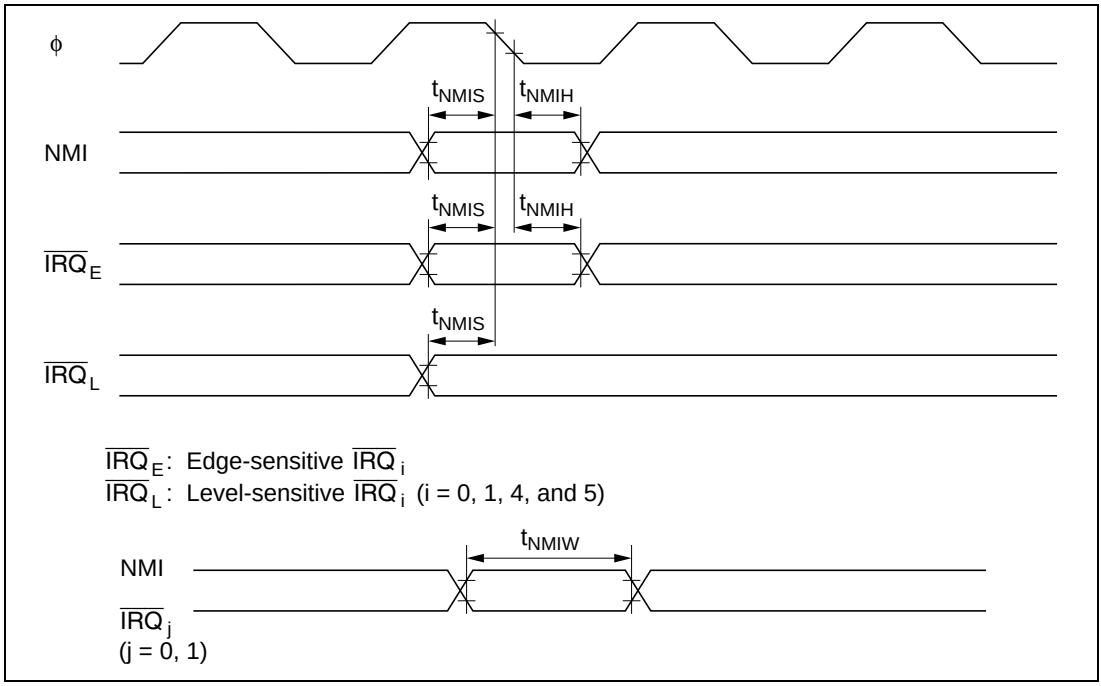


Figure 18.12 Interrupt Input Timing

18.3.3 Clock Timing

Clock timing is shown below.

- Oscillator settling timing
  Figure 18.13 shows the oscillator settling timing.

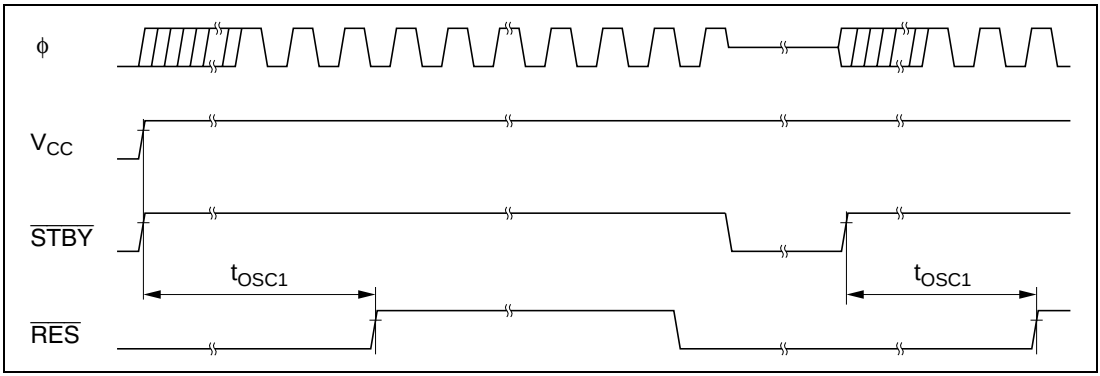


Figure 18.13 Oscillator Settling Timing

18.3.4 TPC and I/O Port Timing

TPC and I/O port timing is shown below.

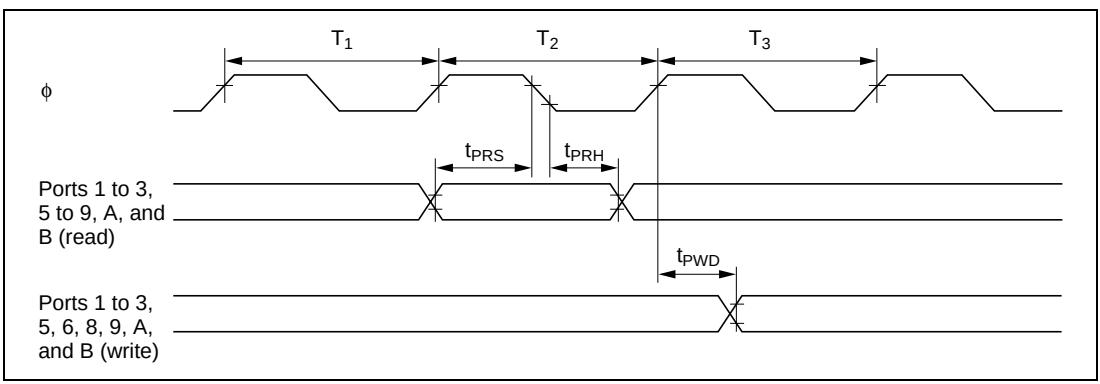


Figure 18.14 TPC and I/O Port Input/Output Timing

18.3.5 ITU Timing

ITU timing is shown as follows:

- ITU input/output timing
  Figure 18.15 shows the ITU input/output timing.
- ITU external clock input timing
  Figure 18.16 shows the ITU external clock input timing.

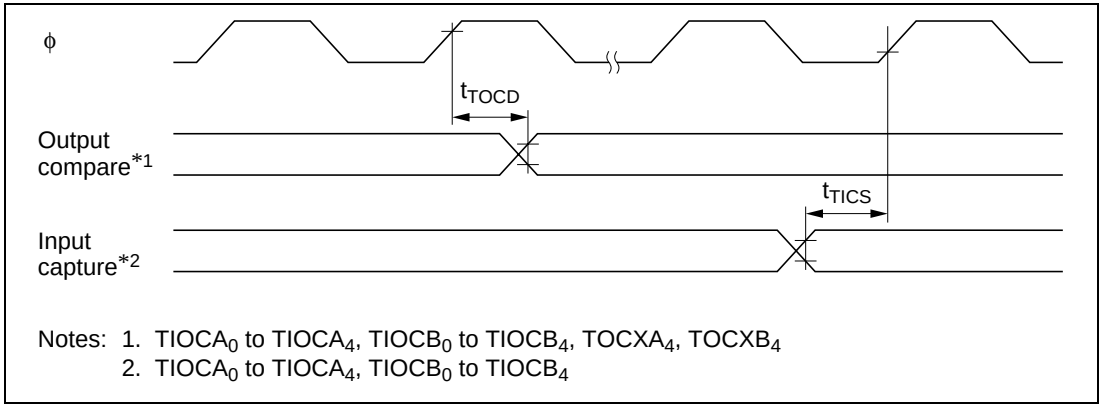


Figure 18.15 ITU Input/Output Timing

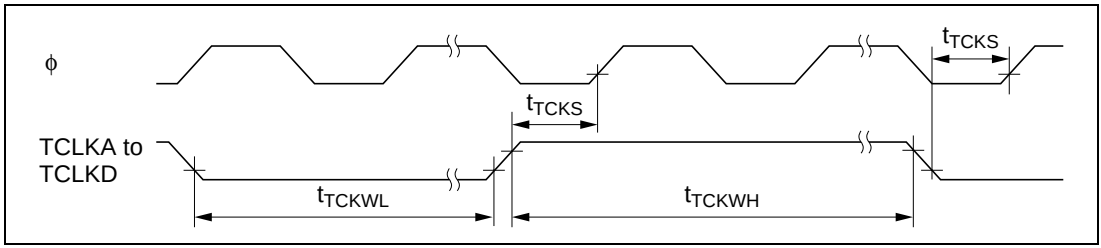


Figure 18.16 ITU External Clock Input Timing

18.3.6 SCI Input/Output Timing

SCI timing is shown as follows:

- SCI input clock timing
  Figure 18.17 shows the SCI input clock timing.
- SCI input/output timing (synchronous mode)
  Figure 18.18 shows the SCI input/output timing in synchronous mode.

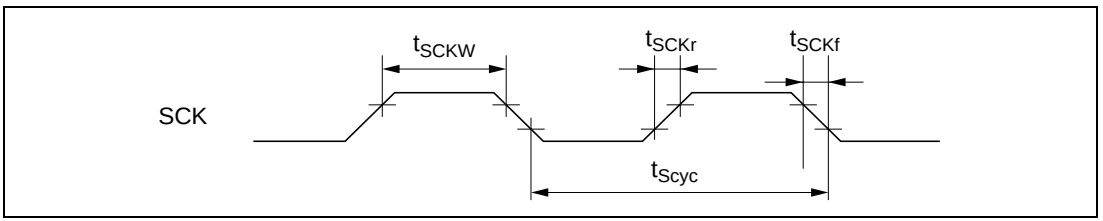


Figure 18.17 SCK Input Clock Timing

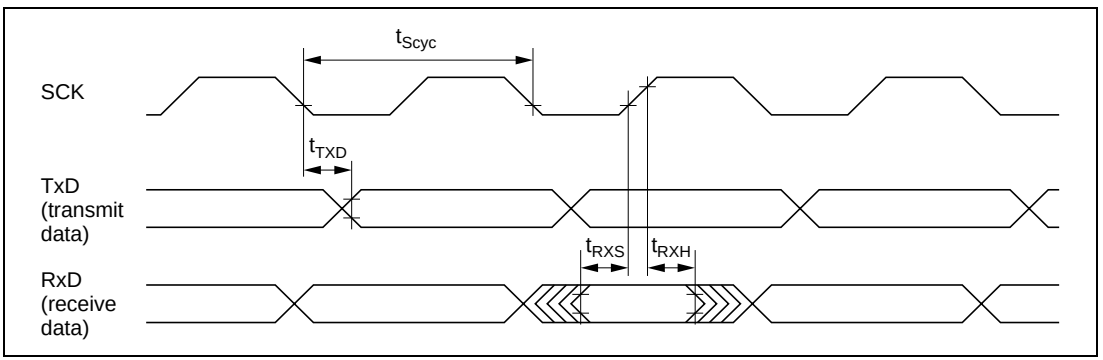


Figure 18.18 SCI Input/Output Timing in Synchronous Mode

Appendix A Instruction Set

A.1 Instruction List

Operand Notation

Symbol	Description
Rd	General destination register*
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 transition from 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-bit registers (R0 to R7 and E0 to E7).

Condition Code Notation

Symbol	Description
↕	Changed according to execution result
*	Undetermined (no guaranteed value)
0	Cleared to 0
1	Set to 1
—	Not affected by execution of the instruction
Δ	Varies depending on conditions, described in notes

Table A.1 Instruction Set

1. Data transfer instructions

Mnemonic	Operand Size	Operation	Addressing Mode and Instruction Length (bytes)									Condition Code						No. of States*1	
			#xx	Rn	@ERn	@(d, ERn)	@–ERn/@ERn+	@aa	@(d, PC)	@@aa	Implied	I	H	N	Z	V	C	Normal	Advanced
MOV.B #xx:8, Rd	B	#xx:8 → Rd8	2									—	—	↕	↕	0	—	2	
MOV.B Rs, Rd	B	Rs8 → Rd8		2								—	—	↕	↕	0	—	2	
MOV.B @ERs, Rd	B	@ERs → Rd8			2							—	—	↕	↕	0	—	4	
MOV.B @(d:16, ERs), Rd	B	@(d:16, ERs) → Rd8				4						—	—	↕	↕	0	—	6	
MOV.B @(d:24, ERs), Rd	B	@(d:24, ERs) → Rd8				8						—	—	↕	↕	0	—	10	
MOV.B @ERs+, Rd	B	@ERs → RD8 ERs32+1 → ERs32					2					—	—	↕	↕	0	—	6	
MOV.B @aa:8, Rd	B	@aa:8 → Rd8						2				—	—	↕	↕	0	—	4	
MOV.B @aa:16, Rd	B	@aa:16 → Rd8						4				—	—	↕	↕	0	—	6	
MOV.B @aa:24, Rd	B	@aa:24 → Rd8						6				—	—	↕	↕	0	—	8	
MOV.B Rs, @ERd	B	Rs8 → @ERd			2							—	—	↕	↕	0	—	4	
MOV.B Rs, @(d:16, ERd)	B	Rd8 → @(d:16, ERd)				4						—	—	↕	↕	0	—	6	
MOV.B Rs, @(d:24, ERd)	B	Rd8 → @(d:24, ERd)				8						—	—	↕	↕	0	—	10	
MOV.B Rs, @–ERd	B	ERd32–1 → ERd32 Rs8 → @ERd					2					—	—	↕	↕	0	—	6	
MOV.B Rs, @aa:8	B	Rs8 → @aa:8						2				—	—	↕	↕	0	—	4	
MOV.B Rs, @aa:16	B	Rs8 → @aa:16						4				—	—	↕	↕	0	—	6	
MOV.B Rs, @aa:24	B	Rs8 → @aa:24						6				—	—	↕	↕	0	—	8	
MOV.W #xx:16, Rd	W	#xx:16 → Rd16	4									—	—	↕	↕	0	—	4	
MOV.W Rs, Rd	W	Rs16 → Rd16		2								—	—	↕	↕	0	—	2	
MOV.W @ERs, Rd	W	@ERs → Rd16			2							—	—	↕	↕	0	—	4	
MOV.W @(d:16, ERs), Rd	W	@(d:16, ERs) → Rd16				4						—	—	↕	↕	0	—	6	
MOV.W @(d:24, ERs), Rd	W	@(d:24, ERs) → Rd16				8						—	—	↕	↕	0	—	10	
MOV.W @ERs+, Rd	W	@ERs → Rd16 ERs32+2 → @ERd32					2					—	—	↕	↕	0	—	6	
MOV.W @aa:16, Rd	W	@aa:16 → Rd16						4				—	—	↕	↕	0	—	6	

Mnemonic	Operand Size	Operation	Addressing Mode and Instruction Length (bytes)									Condition Code						No. of States*1	
			#xx	Rn	@ERn	@(d, ERn)	@–ERn/@ERn+	@aa	@(d, PC)	@@aa	Implied	I	H	N	Z	V	C	Normal	Advanced
MOV.W @aa:24, Rd	W	@aa:24 → Rd16						6				—	—	↕	↕	0	—	8	
MOV.W Rs, @ERd	W	Rs16 → @ERd			2							—	—	↕	↕	0	—	4	
MOV.W Rs, @(d:16, ERd)	W	Rs16 → @(d:16, ERd)				4						—	—	↕	↕	0	—	6	
MOV.W Rs, @(d:24, ERd)	W	Rs16 → @(d:24, ERd)				8						—	—	↕	↕	0	—	8	
MOV.W Rs, @–ERd	W	ERd32–2 → ERd32 Rs16 → @ERd					2					—	—	↕	↕	0	—	6	
MOV.W Rs, @aa:16	W	Rs16 → @aa:16						4				—	—	↕	↕	0	—	6	
MOV.W Rs, @aa:24	W	Rs16 → @aa:24						6				—	—	↕	↕	0	—	8	
MOV.L #xx:32, Rd	L	#xx:32 → Rd32	6									—	—	↕	↕	0	—	6	
MOV.L ERs, ERd	L	ERs32 → ERd32		2								—	—	↕	↕	0	—	2	
MOV.L @ERs, ERd	L	@ERs → ERd32			4							—	—	↕	↕	0	—	8	
MOV.L @(d:16, ERs), ERd	L	@(d:16, ERs) → ERd32				6						—	—	↕	↕	0	—	10	
MOV.L @(d:24, ERs), ERd	L	@(d:24, ERs) → ERd32				10						—	—	↕	↕	0	—	14	
MOV.L @ERs+, ERd	L	@ERs → ERd32 ERs32+4 → ERs32					4					—	—	↕	↕	0	—	10	
MOV.L @aa:16, ERd	L	@aa:16 → ERd32						6				—	—	↕	↕	0	—	10	
MOV.L @aa:24, ERd	L	@aa:24 → ERd32						8				—	—	↕	↕	0	—	12	
MOV.L ERs, @ERd	L	ERs32 → @ERd			4							—	—	↕	↕	0	—	8	
MOV.L ERs, @(d:16, ERd)	L	ERs32 → @(d:16, ERd)				6						—	—	↕	↕	0	—	10	
MOV.L ERs, @(d:24, ERd)	L	ERs32 → @(d:24, ERd)				10						—	—	↕	↕	0	—	14	
MOV.L ERs, @–ERd	L	ERd32–4 → ERd32 ERs32 → @ERd					4					—	—	↕	↕	0	—	10	
MOV.L ERs, @aa:16	L	ERs32 → @aa:16						6				—	—	↕	↕	0	—	10	
MOV.L ERs, @aa:24	L	ERs32 → @aa:24						8				—	—	↕	↕	0	—	12	
POP.W Rn	W	@SP → Rn16 SP+2 → SP									2	—	—	↕	↕	0	—	6	
POP.L ERn	L	@SP → ERn32 SP+4 → SP									4	—	—	↕	↕	0	—	10	

Mnemonic	Operand Size	Operation	Addressing Mode and Instruction Length (bytes)									Condition Code						No. of States*1	
			#xx	Rn	@ERn	@(d, ERn)	@–ERn/@ERn+	@aa	@(d, PC)	@@aa	Implied	I	H	N	Z	V	C	Normal	Advanced
PUSH.W Rn	W	SP–2 → SP Rn16 → @SP									2	—	—	↕	↕	0	—	6	
PUSH.L ERn	L	SP–4 → SP ERn32 → @SP									4	—	—	↕	↕	0	—	10	
MOVFPE @aa:16, Rd	B	Cannot be used in the H8/3039 Group						4				Cannot be used in the H8/3039 Group							
MOVTPE Rs, @aa:16	B	Cannot be used in the H8/3039 Group						4				Cannot be used in the H8/3039 Group							

2. Arithmetic instructions

Mnemonic	Operand Size	Operation	Addressing Mode and Instruction Length (bytes)									Condition Code						No. of States*1	
			#xx	Rn	@ERn	@(d, ERn)	@–ERn/@ERn+	@aa	@(d, PC)	@@aa	Implied	I	H	N	Z	V	C	Normal	Advanced
ADD.B #xx:8, Rd	B	Rd8+#xx:8 → Rd8	2									—	↕	↕	↕	↕	↕	2	
ADD.B Rs, Rd	B	Rd8+Rs8 → Rd8		2								—	↕	↕	↕	↕	↕	2	
ADD.W #xx:16, Rd	W	Rd16+#xx:16 → Rd16	4									—	(1)	↕	↕	↕	↕	4	
ADD.W Rs, Rd	W	Rd16+Rs16 → Rd16		2								—	(1)	↕	↕	↕	↕	2	
ADD.L #xx:32, ERd	L	ERd32+#xx:32 → ERd32	6									—	(2)	↕	↕	↕	↕	6	
ADD.L ERs, ERd	L	ERd32+ERs32 → ERd32		2								—	(2)	↕	↕	↕	↕	2	
ADDX.B #xx:8, Rd	B	Rd8+#xx:8 +C → Rd8	2									—	↕	↕	(3)	↕	↕	2	
ADDX.B Rs, Rd	B	Rd8+Rs8 +C → Rd8		2								—	↕	↕	(3)	↕	↕	2	
ADDS.L #1, ERd	L	ERd32+1 → ERd32		2								—	—	—	—	—	—	2	
ADDS.L #2, ERd	L	ERd32+2 → ERd32		2								—	—	—	—	—	—	2	
ADDS.L #4, ERd	L	ERd32+4 → ERd32		2								—	—	—	—	—	—	2	
INC.B Rd	B	Rd8+1 → Rd8		2								—	—	↕	↕	↕	—	2	
INC.W #1, Rd	W	Rd16+1 → Rd16		2								—	—	↕	↕	↕	—	2	
INC.W #2, Rd	W	Rd16+2 → Rd16		2								—	—	↕	↕	↕	—	2	

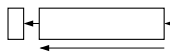
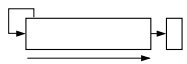
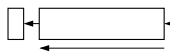
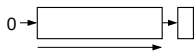
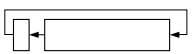
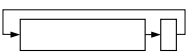
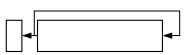
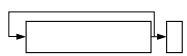
Mnemonic	Operand Size	Operation	Addressing Mode and Instruction Length (bytes)									Condition Code						No. of States*1	
			#xx	Rn	@ERn	@(d, ERn)	@–ERn/@ERn+	@aa	@(d, PC)	@@aa	Implied	I	H	N	Z	V	C	Normal	Advanced
INC.L #1, ERd	L	ERd32+1 → ERd32		2								—	—	↕	↕	↕	—	2	
INC.L #2, ERd	L	ERd32+2 → ERd32		2								—	—	↕	↕	↕	—	2	
DAA Rd	B	Rd8 decimal adjust → Rd8		2								—	*	↕	↕	*	—	2	
SUB.B Rs, Rd	B	Rd8–Rs8 → Rd8		2								—	↕	↕	↕	↕	↕	2	
SUB.W #xx:16, Rd	W	Rd16–#xx:16 → Rd16	4									—	(1)	↕	↕	↕	↕	4	
SUB.W Rs, Rd	W	Rd16–Rs16 → Rd16		2								—	(1)	↕	↕	↕	↕	2	
SUB.L #xx:32, ERd	L	ERd32–#xx:32 → ERd32	6									—	(2)	↕	↕	↕	↕	6	
SUB.L ERs, ERd	L	ERd32–ERs32 → ERd32		2								—	(2)	↕	↕	↕	↕	2	
SUBX.B #xx:8, Rd	B	Rd8–#xx:8–C → Rd8	2									—	↕	↕	(3)	↕	↕	2	
SUBX.B Rs, Rd	B	Rd8–Rs8–C → Rd8		2								—	↕	↕	(3)	↕	↕	2	
SUBS.L #1, ERd	L	ERd32–1 → ERd32		2								—	—	—	—	—	—	2	
SUBS.L #2, ERd	L	ERd32–2 → ERd32		2								—	—	—	—	—	—	2	
SUBS.L #4, ERd	L	ERd32–4 → ERd32		2								—	—	—	—	—	—	2	
DEC.B Rd	B	Rd8–1 → Rd8		2								—	—	↕	↕	↕	—	2	
DEC.W #1, Rd	W	Rd16–1 → Rd16		2								—	—	↕	↕	↕	—	2	
DEC.W #2, Rd	W	Rd16–2 → Rd16		2								—	—	↕	↕	↕	—	2	
DEC.L #1, ERd	L	ERd32–1 → ERd32		2								—	—	↕	↕	↕	—	2	
DEC.L #2, ERd	L	ERd32–2 → ERd32		2								—	—	↕	↕	↕	—	2	
DAS.Rd	B	Rd8 decimal adjust → Rd8		2								—	*	↕	↕	*	—	2	
MULXU. B Rs, Rd	B	Rd8 × Rs8 → Rd16 (unsigned multiplication)		2								—	—	—	—	—	—	14	
MULXU. W Rs, ERd	W	Rd16 × Rs16 → ERd32 (unsigned multiplication)		2								—	—	—	—	—	—	22	
MULXS. B Rs, Rd	B	Rd8 × Rs8 → Rd16 (signed multiplication)		4								—	—	↕	↕	—	—	16	
MULXS. W Rs, ERd	W	Rd16 × Rs16 → ERd32 (signed multiplication)		4								—	—	↕	↕	—	—	24	
DIVXU. B Rs, Rd	B	Rd16 ÷ Rs8 → Rd16 (RdH: remainder, RdL: quotient) (unsigned division)		2								—	—	(6)	(7)	—	—	14	

Mnemonic	Operand Size	Operation	Addressing Mode and Instruction Length (bytes)									Condition Code						No. of States*1	
			#xx	Rn	@ERn	@(d, ERn)	@–ERn/@ERn+	@aa	@(d, PC)	@@aa	Implied	I	H	N	Z	V	C	Normal	Advanced
DIVXU. W Rs, ERd	W	ERd32 ÷ Rs16 →ERd32 (Ed: remainder, Rd: quotient) (unsigned division)		2								—	—	(6)	(7)	—	—	22	
DIVXS. B Rs, Rd	B	Rd16 ÷ Rs8 → Rd16 (RdH: remainder, RdL: quotient) (signed division)		4								—	—	(8)	(7)	—	—	16	
DIVXS. W Rs, ERd	W	ERd32 ÷ Rs16 → ERd32 (Ed: remainder, Rd: quotient) (signed division)		4								—	—	(8)	(7)	—	—	24	
CMP.B #xx:8, Rd	B	Rd8–#xx:8	2									—	↕	↕	↕	↕	↕	2	
CMP.B Rs, Rd	B	Rd8–Rs8		2								—	↕	↕	↕	↕	↕	2	
CMP.W #xx:16, Rd	W	Rd16–#xx:16	4									—	(1)	↕	↕	↕	↕	4	
CMP.W Rs, Rd	W	Rd16–Rs16		2								—	(1)	↕	↕	↕	↕	2	
CMP.L #xx:32, ERd	L	ERd32–#xx:32	6									—	(2)	↕	↕	↕	↕	4	
CMP.L ERs, ERd	L	ERd32–ERs32		2								—	(2)	↕	↕	↕	↕	2	
NEG.B Rd	B	0–Rd8 → Rd8		2								—	↕	↕	↕	↕	↕	2	
NEG.W Rd	W	0–Rd16 → Rd16		2								—	↕	↕	↕	↕	↕	2	
NEG.L ERd	L	0–ERd32 → ERd32		2								—	↕	↕	↕	↕	↕	2	
EXTU.W Rd	W	0 → (<bits 15 to 8> of Rd16)		2								—	—	0	↕	0	—	2	
EXTU.L ERd	L	0 → (<bits 31 to 16> of Rd32)		2								—	—	0	↕	0	—	2	
EXTS.W Rd	W	(<bit 7> of Rd16) → (<bits 15 to 8> of Rd16)		2								—	—	↕	↕	0	—	2	
EXTS.L ERd	L	(<bit 15> of Rd32) → (<bits 31 to 16> of ERd32)		2								—	—	↕	↕	0	—	2	

3. Logic instructions

Mnemonic	Operand Size	Operation	Addressing Mode and Instruction Length (bytes)									Condition Code						No. of States*1	
			#xx	Rn	@ERn	@(d, ERn)	@–ERn/@ERn+	@aa	@(d, PC)	@@aa	Implied	I	H	N	Z	V	C	Normal	Advanced
AND.B #xx:8, Rd	B	Rd8∧#xx:8 → Rd8	2									—	—	↕	↕	0	—	2	
AND.B Rs, Rd	B	Rd8∧Rs8 → Rd8		2								—	—	↕	↕	0	—	2	
AND.W #xx:16, Rd	W	Rd16∧#xx:16 → Rd16	4									—	—	↕	↕	0	—	4	
AND.W Rs, Rd	W	Rd16∧Rs16 → Rd16		2								—	—	↕	↕	0	—	2	
AND.L #xx:32, ERd	L	ERd32∧#xx:32 → ERd32	6									—	—	↕	↕	0	—	6	
AND.L ERs, ERd	L	ERd32∧ERs32 → ERd32		4								—	—	↕	↕	0	—	4	
OR.B #xx:8, Rd	B	Rd8∨#xx:8 → Rd8	2									—	—	↕	↕	0	—	2	
OR.B Rs, Rd	B	Rd8∨Rs8 → Rd8		2								—	—	↕	↕	0	—	2	
OR.W #xx:16, Rd	W	Rd16∨#xx:16 → Rd16	4									—	—	↕	↕	0	—	4	
OR.W Rs, Rd	W	Rd16∨Rs16 → Rd16		2								—	—	↕	↕	0	—	2	
OR.L #xx:32, ERd	L	ERd32∨#xx:32 → ERd32	6									—	—	↕	↕	0	—	6	
OR.L ERs, ERd	L	ERd32∨ERs32 → ERd32		4								—	—	↕	↕	0	—	4	
XOR.B #xx:8, Rd	B	Rd8⊕#xx:8 → Rd8	2									—	—	↕	↕	0	—	2	
XOR.B Rs, Rd	B	Rd8⊕Rs8 → Rd8		2								—	—	↕	↕	0	—	2	
XOR.W #xx:16, Rd	W	Rd16⊕#xx:16 → Rd16	4									—	—	↕	↕	0	—	4	
XOR.W Rs, Rd	W	Rd16⊕Rs16 → Rd16		2								—	—	↕	↕	0	—	2	
XOR.L #xx:32, ERd	L	ERd32⊕#xx:32 → ERd32	6									—	—	↕	↕	0	—	6	
XOR.L ERs, ERd	L	ERd32⊕ERs32 → ERd32		4								—	—	↕	↕	0	—	4	
NOT.B Rd	B	¬ Rd8 → Rd8		2								—	—	↕	↕	0	—	2	
NOT.W Rd	W	¬ Rd16 → Rd16		2								—	—	↕	↕	0	—	2	
NOT.L ERd	L	¬ Rd32 → Rd32		2								—	—	↕	↕	0	—	2	

4. Shift instructions

Mnemonic	Operand Size	Operation	Addressing Mode and Instruction Length (bytes)									Condition Code						No. of States*1	
			#xx	Rn	@ERn	@(d, ERn)	@–ERn/@ERn+	@aa	@(d, PC)	@@aa	Implied	I	H	N	Z	V	C	Normal	Advanced
SHAL.B Rd	B			2								—	—	↕	↕	↕	↕	2	
SHAL.W Rd	W			2								—	—	↕	↕	↕	↕	2	
SHAL.L ERd	L			2								—	—	↕	↕	↕	↕	2	
SHAR.B Rd	B			2								—	—	↕	↕	0	↕	2	
SHAR.W Rd	W			2								—	—	↕	↕	0	↕	2	
SHAR.L ERd	L			2								—	—	↕	↕	0	↕	2	
SHLL.B Rd	B			2								—	—	↕	↕	0	↕	2	
SHLL.W Rd	W			2								—	—	↕	↕	0	↕	2	
SHLL.L ERd	L			2								—	—	↕	↕	0	↕	2	
SHLR.B Rd	B			2								—	—	↕	↕	0	↕	2	
SHLR.W Rd	W			2								—	—	↕	↕	0	↕	2	
SHLR.L ERd	L			2								—	—	↕	↕	0	↕	2	
ROTXL.B Rd	B			2								—	—	↕	↕	0	↕	2	
ROTXL.W Rd	W			2								—	—	↕	↕	0	↕	2	
ROTXL.L ERd	L			2								—	—	↕	↕	0	↕	2	
ROTXR.B Rd	B			2								—	—	↕	↕	0	↕	2	
ROTXR.W Rd	W			2								—	—	↕	↕	0	↕	2	
ROTXR.L ERd	L			2								—	—	↕	↕	0	↕	2	
ROTL.B Rd	B			2								—	—	↕	↕	0	↕	2	
ROTL.W Rd	W			2								—	—	↕	↕	0	↕	2	
ROTL.L ERd	L			2								—	—	↕	↕	0	↕	2	
ROTR.B Rd	B			2								—	—	↕	↕	0	↕	2	
ROTR.W Rd	W			2								—	—	↕	↕	0	↕	2	
ROTR.L ERd	L			2								—	—	↕	↕	0	↕	2	

5. Bit manipulation instructions

Mnemonic	Operand Size	Operation	Addressing Mode and Instruction Length (bytes)									Condition Code						No. of States*1	
			#xx	Rn	@ERn	@(d, ERn)	@–ERn/@ERn+	@aa	@(d, PC)	@@aa	Implied	I	H	N	Z	V	C	Normal	Advanced
BSET #xx:3, Rd	B	(#xx:3 of Rd8) ← 1		2								—	—	—	—	—	—	2	
BSET #xx:3, @ERd	B	(#xx:3 of @ERd) ← 1			4							—	—	—	—	—	—	8	
BSET #xx:3, @aa:8	B	(#xx:3 of @aa:8) ← 1						4				—	—	—	—	—	—	8	
BSET Rn, Rd	B	(Rn8 of Rd8) ← 1		2								—	—	—	—	—	—	2	
BSET Rn, @ERd	B	(Rn8 of @ERd) ← 1			4							—	—	—	—	—	—	8	
BSET Rn, @aa:8	B	(Rn8 of @aa:8) ← 1						4				—	—	—	—	—	—	8	
BCLR #xx:3, Rd	B	(#xx:3 of Rd8) ← 0		2								—	—	—	—	—	—	2	
BCLR #xx:3, @ERd	B	(#xx:3 of @ERd) ← 0			4							—	—	—	—	—	—	8	
BCLR #xx:3, @aa:8	B	(#xx:3 of @aa:8) ← 0						4				—	—	—	—	—	—	8	
BCLR Rn, Rd	B	(Rn8 of Rd8) ← 0		2								—	—	—	—	—	—	2	
BCLR Rn, @ERd	B	(Rn8 of @ERd) ← 0			4							—	—	—	—	—	—	8	
BCLR Rn, @aa:8	B	(Rn8 of @aa:8) ← 0						4				—	—	—	—	—	—	8	
BNOT #xx:3, Rd	B	(#xx:3 of Rd8) ← ¬ (#xx:3 of Rd8)		2								—	—	—	—	—	—	2	
BNOT #xx:3, @ERd	B	(#xx:3 of @ERd) ← ¬ (#xx:3 of @ERd)			4							—	—	—	—	—	—	8	
BNOT #xx:3, @aa:8	B	(#xx:3 of @aa:8) ← ¬ (#xx:3 of @aa:8)						4				—	—	—	—	—	—	8	
BNOT Rn, Rd	B	(Rn8 of Rd8) ← ¬ (Rn8 of Rd8)		2								—	—	—	—	—	—	2	
BNOT Rn, @ERd	B	(Rn8 of @ERd) ← ¬ (Rn8 of @ERd)			4							—	—	—	—	—	—	8	
BNOT Rn, @aa:8	B	(Rn8 of @aa:8) ← ¬ (Rn8 of @aa:8)						4				—	—	—	—	—	—	8	
BTST #xx:3, Rd	B	¬ (#xx:3 of Rd8) → Z		2								—	—	—	↕	—	—	2	
BTST #xx:3, @ERd	B	¬ (#xx:3 of @ERd) → Z			4							—	—	—	↕	—	—	6	
BTST #xx:3, @aa:8	B	¬ (#xx:3 of @aa:8) → Z						4				—	—	—	↕	—	—	6	
BTST Rn, Rd	B	¬ (Rn8 of @Rd8) → Z		2								—	—	—	↕	—	—	2	
BTST Rn, @ERd	B	¬ (Rn8 of @ERd) → Z			4							—	—	—	↕	—	—	6	
BTST Rn, @aa:8	B	¬ (Rn8 of @aa:8) → Z						4				—	—	—	↕	—	—	6	
BLD #xx:3, Rd	B	(#xx:3 of Rd8) → C		2								—	—	—	—	—	↕	2	

Mnemonic	Operand Size	Operation	Addressing Mode and Instruction Length (bytes)									Condition Code						No. of States*1	
			#xx	Rn	@ERn	@(d, ERn)	@–ERn/@ERn+	@aa	@(d, PC)	@@aa	Implied	I	H	N	Z	V	C	Normal	Advanced
BLD #xx:3, @ERd	B	(#xx:3 of @ERd) → C			4							—	—	—	—	—	↕	6	
BLD #xx:3, @aa:8	B	(#xx:3 of @aa:8) → C						4				—	—	—	—	—	↕	6	
BILD #xx:3, Rd	B	¬ (#xx:3 of Rd8) → C		2								—	—	—	—	—	↕	2	
BILD #xx:3, @ERd	B	¬ (#xx:3 of @ERd) → C			4							—	—	—	—	—	↕	6	
BILD #xx:3, @aa:8	B	¬ (#xx:3 of @aa:8) → C						4				—	—	—	—	—	↕	6	
BST #xx:3, Rd	B	C → (#xx:3 of Rd8)		2								—	—	—	—	—	—	2	
BST #xx:3, @ERd	B	C → (#xx:3 of @ERd24)			4							—	—	—	—	—	—	8	
BST #xx:3, @aa:8	B	C → (#xx:3 of @aa:8)						4				—	—	—	—	—	—	8	
BIST #xx:3, Rd	B	¬ C → (#xx:3 of Rd8)		2								—	—	—	—	—	—	2	
BIST #xx:3, @ERd	B	¬ C → (#xx:3 of @ERd24)			4							—	—	—	—	—	—	8	
BIST #xx:3, @aa:8	B	¬ C → (#xx:3 of @aa:8)						4				—	—	—	—	—	—	8	
BAND #xx:3, Rd	B	C∧(#xx:3 of Rd8) → C		2								—	—	—	—	—	↕	2	
BAND #xx:3, @ERd	B	C∧(#xx:3 of @ERd24) → C			4							—	—	—	—	—	↕	6	
BAND #xx:3, @aa:8	B	C∧(#xx:3 of @aa:8) → C						4				—	—	—	—	—	↕	6	
BIAND #xx:3, Rd	B	C∧ ¬ (#xx:3 of Rd8) → C		2								—	—	—	—	—	↕	2	
BIAND #xx:3, @ERd	B	C∧ ¬ (#xx:3 of @ERd24) → C			4							—	—	—	—	—	↕	6	
BIAND #xx:3, @aa:8	B	C∧ ¬ (#xx:3 of @aa:8) → C						4				—	—	—	—	—	↕	6	
BOR #xx:3, Rd	B	C∨(#xx:3 of Rd8) → C		2								—	—	—	—	—	↕	2	
BOR #xx:3, @ERd	B	C∨(#xx:3 of @ERd24) → C			4							—	—	—	—	—	↕	6	
BOR #xx:3, @aa:8	B	C∨(#xx:3 of @aa:8) → C						4				—	—	—	—	—	↕	6	
BIOR #xx:3, Rd	B	C∨ ¬ (#xx:3 of Rd8) → C		2								—	—	—	—	—	↕	2	
BIOR #xx:3, @ERd	B	C∨ ¬ (#xx:3 of @ERd24) → C			4							—	—	—	—	—	↕	6	
BIOR #xx:3, @aa:8	B	C∨ ¬ (#xx:3 of @aa:8) → C						4				—	—	—	—	—	↕	6	
BXOR #xx:3, Rd	B	C ⊕ (#xx:3 of Rd8) → C		2								—	—	—	—	—	↕	2	
BXOR #xx:3, @ERd	B	C ⊕(#xx:3 of @ERd24) → C			4							—	—	—	—	—	↕	6	
BXOR #xx:3, @aa:8	B	C ⊕(#xx:3 of @aa:8)→ C						4				—	—	—	—	—	↕	6	
BIXOR #xx:3, Rd	B	C ⊕ ¬ (#xx:3 of Rd8) → C		2								—	—	—	—	—	↕	2	
BIXOR #xx:3, @ERd	B	C ⊕ ¬ (#xx:3 of @ERd24) → C			4							—	—	—	—	—	↕	6	
BIXOR #xx:3, @aa:8	B	C ⊕ ¬ (#xx:3 of @aa:8)→ C						4				—	—	—	—	—	↕	6	

6. Branching instructions

Mnemonic	Operand Size	Operation		Addressing Mode and Instruction Length (bytes)									Condition Code						No. of States*1	
				#xx	Rn	@ERn	@(d, ERn)	@–ERn/@ERn+	@aa	@(d, PC)	@@aa	Implied	I	H	N	Z	V	C	Normal	Advanced
BRA d:8 (BT d:8)	—	If condition is true then PC ← PC+d else next;	Always							2			—	—	—	—	—	—	4	
BRA d:16 (BT d:16)	—									4			—	—	—	—	—	—	6	
BRN d:8 (BF d:8)	—		Never							2			—	—	—	—	—	—	4	
BRN d:16 (BF d:16)	—									4			—	—	—	—	—	—	6	
BHI d:8	—		C ∨ Z = 0							2			—	—	—	—	—	—	4	
BHI d:16	—									4			—	—	—	—	—	—	6	
BLS d:8	—		C ∨ Z = 1							2			—	—	—	—	—	—	4	
BLS d:16	—									4			—	—	—	—	—	—	6	
BCC d:8 (BHS d:8)	—		C = 0							2			—	—	—	—	—	—	4	
BCC d:16 (BHS d:16)	—									4			—	—	—	—	—	—	6	
BCS d:8 (BLO d:8)	—		C = 1							2			—	—	—	—	—	—	4	
BCS d:16 (BLO d:16)	—									4			—	—	—	—	—	—	6	
BNE d:8	—		Z = 0							2			—	—	—	—	—	—	4	
BNE d:16	—									4			—	—	—	—	—	—	6	
BEQ d:8	—		Z = 1							2			—	—	—	—	—	—	4	
BEQ d:16	—									4			—	—	—	—	—	—	6	
BVC d:8	—		V = 0							2			—	—	—	—	—	—	4	
BVC d:16	—									4			—	—	—	—	—	—	6	
BVS d:8	—		V = 1							2			—	—	—	—	—	—	4	
BVS d:16	—									4			—	—	—	—	—	—	6	
BPL d:8	—		N = 0							2			—	—	—	—	—	—	4	
BPL d:16	—									4			—	—	—	—	—	—	6	
BMI d:8	—		N = 1							2			—	—	—	—	—	—	4	
BMI d:16	—									4			—	—	—	—	—	—	6	
BGE d:8	—		N ⊕ V = 0							2			—	—	—	—	—	—	4	
BGE d:16	—									4			—	—	—	—	—	—	6	
BLT d:8	—		N ⊕ V = 1							2			—	—	—	—	—	—	4	
BLT d:16	—									4			—	—	—	—	—	—	6	
BGT d:8	—		Z ∨ (N ⊕ V) = 0							2			—	—	—	—	—	—	4	
BGT d:16	—									4			—	—	—	—	—	—	6	

Mnemonic	Operand Size	Operation		Addressing Mode and Instruction Length (bytes)									Condition Code						No. of States*1	
				#xx	Rn	@ERn	@(d, ERn)	@–ERn/@ERn+	@aa	@(d, PC)	@@aa	Implied	I	H	N	Z	V	C	Normal	Advanced
BLE d:8	—	If condition is true then PC ← PC+d else next;	Z ∨ (N ⊕ V) = 1							2			—	—	—	—	—	—	4	
BLE d:16	—									4			—	—	—	—	—	—	6	
JMP @ERn	—	PC ← ERn				2							—	—	—	—	—	—	4	
JMP @aa:24	—	PC ← aa:24							4				—	—	—	—	—	—	6	
JMP @@aa:8	—	PC ← @aa:8									2		—	—	—	—	—	—	8	10
BSR d:8	—	PC → @–SP PC ← PC+d:8								2			—	—	—	—	—	—	6	8
BSR d:16	—	PC → @–SP PC ← PC+d:16								4			—	—	—	—	—	—	8	10
JSR @ERn	—	PC → @–SP PC ← @ERn				2							—	—	—	—	—	—	6	8
JSR @aa:24	—	PC → @–SP PC ← @aa:24							4				—	—	—	—	—	—	8	10
JSR @@aa:8	—	PC → @–SP PC ← @aa:8									2		—	—	—	—	—	—	8	12
RTS	—	PC ← @SP+										2	—	—	—	—	—	—	8	10

7. System control instructions

Mnemonic	Operand Size	Operation	Addressing Mode and Instruction Length (bytes)									Condition Code						No. of States*1	
			#xx	Rn	@ERn	@(d, ERn)	@–ERn/@ERn+	@aa	@(d, PC)	@@aa	Implied	I	H	N	Z	V	C	Normal	Advanced
TRAPA #x:2	—	PC → @–SP CCR → @–SP <vector> → PC									2	1	—	—	—	—	—	14	16
RTE	—	CCR ← @SP+ PC ← @SP+										↕	↕	↕	↕	↕	↕	10	
SLEEP	—	Transition to power-down state										—	—	—	—	—	—	2	
LDC #xx:8, CCR	B	#xx:8 → CCR	2									↕	↕	↕	↕	↕	↕	2	
LDC Rs, CCR	B	Rs8 → CCR		2								↕	↕	↕	↕	↕	↕	2	
LDC @ERs, CCR	W	@ERs → CCR			4							↕	↕	↕	↕	↕	↕	6	
LDC @(d:16, ERs), CCR	W	@(d:16, ERs) → CCR				6						↕	↕	↕	↕	↕	↕	8	
LDC @(d:24, ERs), CCR	W	@(d:24, ERs) → CCR				10						↕	↕	↕	↕	↕	↕	12	
LDC @ERs+, CCR	W	@ERs → CCR ERs32+2 → ERs32					4					↕	↕	↕	↕	↕	↕	8	
LDC @aa:16, CCR	W	@aa:16 → CCR						6				↕	↕	↕	↕	↕	↕	8	
LDC @aa:24, CCR	W	@aa:24 → CCR						8				↕	↕	↕	↕	↕	↕	10	
STC CCR, Rd	B	CCR → Rd8		2								—	—	—	—	—	—	2	
STC CCR, @ERd	W	CCR → @ERd			4							—	—	—	—	—	—	6	
STC CCR, @(d:16, ERd)	W	CCR → @(d:16, ERd)				6						—	—	—	—	—	—	8	
STC CCR, @(d:24, ERd)	W	CCR → @(d:24, ERd)				10						—	—	—	—	—	—	12	
STC CCR, @–ERd	W	ERd32–2 → ERd32 CCR → @ERd					4					—	—	—	—	—	—	8	
STC CCR, @aa:16	W	CCR → @aa:16						6				—	—	—	—	—	—	8	
STC CCR, @aa:24	W	CCR → @aa:24						8				—	—	—	—	—	—	10	
ANDC #xx:8, CCR	B	CCR∧#xx:8 → CCR	2									↕	↕	↕	↕	↕	↕	2	
ORC #xx:8, CCR	B	CCR∨#xx:8 → CCR	2									↕	↕	↕	↕	↕	↕	2	
XORC #xx:8, CCR	B	CCR⊕#xx:8 → CCR	2									↕	↕	↕	↕	↕	↕	2	
NOP	—	PC ← PC+2									2	—	—	—	—	—	—	2	

8. Block transfer instructions

Mnemonic	Operand Size	Operation	Addressing Mode and Instruction Length (bytes)									Condition Code						No. of States*1	
			#xx	Rn	@ERn	@(d, ERn)	@–ERn/@ERn+	@aa	@(d, PC)	@@aa	Implied	I	H	N	Z	V	C	Normal	Advanced
EEPMOV. B	—	if R4L ≠ 0 then repeat @R5 → @R6 R5+1 → R5 R6+1 → R6 R4L–1 → R4L until R4L=0 else next									4	—	—	—	—	—	—	8+4n*2	
EEPMOV. W	—	if R4 ≠ 0 then repeat @R5 → @R6 R5+1 → R5 R6+1 → R6 R4–1 → R4 until R4=0 else next									4	—	—	—	—	—	—	8+4n*2	

Notes: 1. The number of states is the number of states required for execution when the instruction and its operands are located in on-chip memory. For other cases see 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 previous value.

(5) The number of states required for execution of an instruction that transfers data in 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.

A.2 Operation Code Maps

Table A.2 Operation Code Map (1)

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	E	F
0	NOP	Table A.2 (2)	STC	LDC	ORC	XORC	ANDC	LDC	ADD		Table A.2 (2)	Table A.2 (2)	MOV		ADDX	Table A.2 (2)
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		SUBX	Table A.2 (2)
2	MOV.B															
3																
4	BRA	BRN	BHI	BLS	BCC	BCS	BNE	BEQ	BVC	BVS	BPL	BMI	BGE	BLT	BGT	BLE
5	MULXU	DIVXU	MULXU	DIVXU	RTS	BSR	RTE	TRAPA	Table A.2 (2)	JMP			BSR	JSR		
6	BSET	BNOT	BCLR	BTST	OR	XOR	AND	BST / BIST	MOV				MOV			
7					BOR / BIOR	BXOR / BIXOR	BAND / BIAND	BLD / BILD		Table A.2 (2)	Table A.2 (2)	EEPMOV	Table A.2 (3)			
8	ADD															
9	ADDX															
A	CMP															
B	SUBX															
C	OR															
D	XOR															
E	AND															
F	MOV															

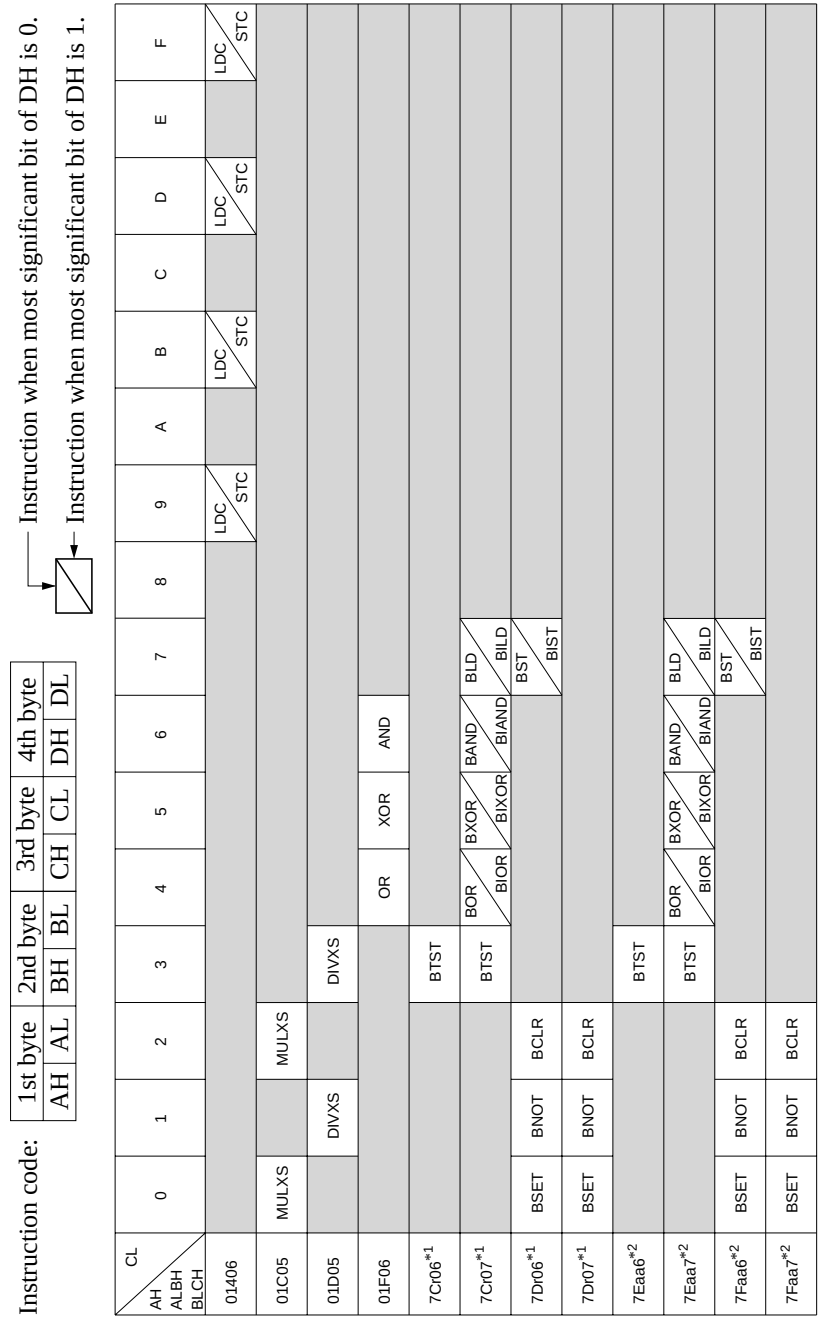
Table A.2 Operation Code Map (2)

Instruction code:

1st byte		2nd byte	
AH	AL	BH	BL

AH \ BH AL	0	1	2	3	4	5	6	7	8	9	A	B	C	D	E	F
01	MOV				LDC/STC				SLEEP				Table A.2 (3)	Table A.2 (3)		Table A.2 (3)
0A	INC								ADD							
0B	ADDS					INC		INC	ADDS					INC		INC
0F	DAA								MOV							
10	SHLL			SHLL					SHAL			SHAL				
11	SHLR			SHLR					SHAR			SHAR				
12	ROTXL			ROTXL					ROTL			ROTL				
13	ROTXR			ROTXR					ROTR			ROTR				
17	NOT			NOT		EXTU		EXTU	NEG			NEG		EXTS		EXTS
1A	DEC								SUB							
1B	SUBS					DEC		DEC	SUB					DEC		DEC
1F	DAS								CMP							
58	BRA	BRN	BHI	BLS	BCC	BCS	BNE	BEQ	BVC	BVS	BPL	BMI	BGE	BLT	BGT	BLE
79	MOV	ADD	CMP	SUB	OR	XOR	AND									
7A	MOV	ADD	CMP	SUB	OR	XOR	AND									

Table A.2 Operation Code Map (3)



Notes: 1. r is the register designation field.
2. aa is the absolute address field.

A.3 Number of States Required for Execution

The tables in this section can be used to calculate the number of states required for instruction execution by the H8/300H CPU. Table A.3 indicates the number of states required per cycle according to the bus size. Table A.4 indicates the number of instruction fetch, data read/write, and other cycles occurring in each instruction. The number of states required for execution of an instruction can be calculated from these two tables as follows:

$$\text{Number of states} = I \times S_I + J \times S_J + K \times S_K + L \times S_L + M \times S_M + N \times S_N$$

Examples of Calculation of Number of States Required for Execution

Examples: Advanced mode, stack located in external address space, on-chip supporting modules accessed with 8-bit bus width, external devices accessed in three states with one wait state and 16-bit bus width.

BSET #0, @FFFFC7:8

From table A.3, $S_I = 4$ and $S_L = 3$

From table A.4, I = L = 2 and J = K = M = N = 0

Number of states = $2 \times 4 + 2 \times 3 = 14$

JSR @@30

From table A.3, $S_I = S_J = S_K = 4$

From table A.4, I = J = K = 2 and L = M = N = 0

Number of states = $2 \times 4 + 2 \times 4 + 2 \times 4 = 24$

Table A.3 Number of States per Cycle

		Access Conditions						
			On-Chip Supporting Module		External Device			
					8-Bit Bus		16-Bit Bus	
Cycle		On-Chip Memory	8-Bit Bus	16-Bit Bus	2-State Access	3-State Access	2-State Access	3-State Access
Instruction fetch	S_I	2	6	3	4	6 + 2m	2	3 + m
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	1

Legend:

m: Number of wait states inserted in external device access

Table A.4 Number of Cycles per Instruction

Instruction	Mnemonic	Instruction Fetch I	Branch Addr. Read J	Stack Operation K	Byte Data Access L	Word Data Access M	Internal Operation N
ADD	ADD.B #xx:8, Rd	1					
	ADD.B Rs, Rd	1					
	ADD.W #xx:16, Rd	2					
	ADD.W Rs, Rd	1					
	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					
	BGE d:8	2					
	BLT d:8	2					
	BGT d:8	2					
	BLE d:8	2					

Instruction	Mnemonic	Instruction Fetch I	Branch Addr. Read J	Stack Operation K	Byte Data Access L	Word Data Access M	Internal Operation N
Bcc	BRA d:16 (BT d:16)	2					2
	BRN d:16 (BF d:16)	2					2
	BHI d:16	2					2
	BLS d:16	2					2
	BCC d:16 (BHS d:16)	2					2
	BCS d:16 (BLO d:16)	2					2
	BNE d:16	2					2
	BEQ d:16	2					2
	BVC d:16	2					2
	BVS d:16	2					2
	BPL d:16	2					2
	BMI d:16	2					2
	BGE d:16	2					2
	BLT d:16	2					2
	BGT d:16	2					2
	BLE d:16	2					2
BCLR	BCLR #xx:3, Rd	1					
	BCLR #xx:3, @ERd	2			2		
	BCLR #xx:3, @aa:8	2			2		
	BCLR Rn, Rd	1					
	BCLR Rn, @ERd	2			2		
	BCLR Rn, @aa:8	2			2		
BIAND	BIAND #xx:3, Rd	1					
	BIAND #xx:3, @ERd	2			1		
	BIAND #xx:3, @aa:8	2			1		
BILD	BILD #xx:3, Rd	1					
	BILD #xx:3, @ERd	2			1		
	BILD #xx:3, @aa:8	2			1		
BIOR	BIOR #xx:8, Rd	1					
	BIOR #xx:8, @ERd	2			1		
	BIOR #xx:8, @aa:8	2			1		
BIST	BIST #xx:3, Rd	1					
	BIST #xx:3, @ERd	2			2		
	BIST #xx:3, @aa:8	2			2		
BIXOR	BIXOR #xx:3, Rd	1					
	BIXOR #xx:3, @ERd	2			1		
	BIXOR #xx:3, @aa:8	2			1		

Instruction	Mnemonic		Instruction Fetch I	Branch Addr. Read J	Stack Operation K	Byte Data Access L	Word Data Access M	Internal Operation N
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	2		1			
		Advanced	2		2			
	BSR d:16	Normal	2		1			2
		Advanced	2		2			2
BST	BST #xx:3, Rd		1					
	BST #xx:3, @ERd		2			2		
	BST #xx:3, @aa:8		2			2		
BTST	BTST #xx:3, Rd		1					
	BTST #xx:3, @ERd		2			1		
	BTST #xx:3, @aa:8		2			1		
	BTST Rn, Rd		1					
	BTST Rn, @ERd		2			1		
	BTST Rn, @aa:8		2			1		
BXOR	BXOR #xx:3, Rd		1					
	BXOR #xx:3, @ERd		2			1		
	BXOR #xx:3, @aa:8		2			1		

Instruction	Mnemonic		Instruction Fetch I	Branch Addr. Read J	Stack Operation K	Byte Data Access L	Word Data Access M	Internal Operation N
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					12
	DIVXS.W Rs, ERd		2					20
DIVXU	DIVXU.B Rs, Rd		1					12
	DIVXU.W Rs, ERd		1					20
EEPMOV	EEPMOV.B		2			2n+2*1		
	EEPMOV.W		2			2n+2*1		
EXTS	EXTS.W Rd		1					
	EXTS.L ERd		1					
EXTU	EXTU.W Rd		1					
	EXTU.L ERd		1					
INC	INC.B Rd		1					
	INC.W #1/2, Rd		1					
	INC.L #1/2, ERd		1					
JMP	JMP @ERn		2					
	JMP @aa:24		2					2
	JMP @@aa:8	Normal	2	1				2
		Advanced	2	2				2
JSR	JSR @ERn	Normal	2		1			
		Advanced	2		2			
	JSR @aa:24	Normal	2		1			2
		Advanced	2		2			2
	JSR @@aa:8	Normal	2	1	1			
		Advanced	2	2	2			

Instruction	Mnemonic	Instruction Fetch I	Branch Addr. Read J	Stack Operation K	Byte Data Access L	Word Data Access M	Internal Operation N
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	2
	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		2
	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.B Rs, @(d:24, ERd)	4			1		
	MOV.B Rs, @–ERd	1			1		2
	MOV.B Rs, @aa:8	1			1		
	MOV.B Rs, @aa:16	2			1		
	MOV.B Rs, @aa:24	3			1		
	MOV.W #xx:16, Rd	2					
	MOV.W Rs, Rd	1					
	MOV.W @ERs, Rd	1				1	
	MOV.W @(d:16, ERs), Rd	2				1	
	MOV.W @(d:24, ERs), Rd	4				1	
	MOV.W @ERs+, Rd	1				1	2
	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	2
	MOV.W Rs, @aa:16	2				1	

Instruction	Mnemonic	Instruction Fetch I	Branch Addr. Read J	Stack Operation K	Byte Data Access L	Word Data Access M	Internal Operation N
MOV	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	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	2
	MOV.L ERs, @aa:16	3				2	
	MOV.L ERs, @aa:24	4				2	
MOVFPE	MOVFPE @aa:16, Rd*[2]	2			1		
MOVTPE	MOVTPE Rs, @aa:16*[2]	2			1		
MULXS	MULXS.B Rs, Rd	2					12
	MULXS.W Rs, ERd	2					20
MULXU	MULXU.B Rs, Rd	1					12
	MULXU.W Rs, ERd	1					20
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					

Instruction	Mnemonic		Instruction Fetch I	Branch Addr. Read J	Stack Operation K	Byte Data Access L	Word Data Access M	Internal Operation N
ORC	ORC #xx:8, CCR		1					
POP	POP.W Rn		1				1	2
	POP.L ERn		2				2	2
PUSH	PUSH.W Rn		1				1	2
	PUSH.L ERn		2				2	2
ROTL	ROTL.B Rd		1					
	ROTL.W Rd		1					
	ROTL.L ERd		1					
ROTR	ROTR.B Rd		1					
	ROTR.W Rd		1					
	ROTR.L ERd		1					
ROTXL	ROTXL.B Rd		1					
	ROTXL.W Rd		1					
	ROTXL.L ERd		1					
ROTXR	ROTXR.B Rd		1					
	ROTXR.W Rd		1					
	ROTXR.L ERd		1					
RTE	RTE		2		2			2
RTS	RTS	Normal	2		1			2
		Advanced	2		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					

Instruction	Mnemonic		Instruction Fetch I	Branch Addr. Read J	Stack Operation K	Byte Data Access L	Word Data Access M	Internal Operation N
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	2
	STC CCR, @aa:16		3				1	
	STC CCR, @aa:24		4				1	
SUB	SUB.B Rs, Rd		1					
	SUB.W #xx:16, Rd		2					
	SUB.W Rs, Rd		1					
	SUB.L #xx:32, ERd		3					
	SUB.L ERs, ERd		1					
SUBS	SUBS #1/2/4, ERd		1					
SUBX	SUBX #xx:8, Rd		1					
	SUBX Rs, Rd		1					
TRAPA	TRAPA #x:2	Normal	2	1	2			4
		Advanced	2	2	2			4
XOR	XOR.B #xx:8, Rd		1					
	XOR.B Rs, Rd		1					
	XOR.W #xx:16, Rd		2					
	XOR.W Rs, Rd		1					
	XOR.L #xx:32, ERd		3					
	XOR.L ERs, ERd		2					
XORC	XORC #xx:8, CCR		1					

Notes: 1. n is the value set in register R4L or R4. The source and destination are accessed n+1 times each.

2. Not used with this LSI.

Appendix B Internal I/O Register Field

B.1 Addresses

Address (low)	Register Name	Data Bus Width	Bit Names								Module Name
			Bit 7	Bit 6	Bit 5	Bit 4	Bit 3	Bit 2	Bit 1	Bit 0	
H'1C											
H'1D											
H'1E											
H'1F											
H'20	—		—	—	—	—	—	—	—	—	
H'21	—		—	—	—	—	—	—	—	—	
H'22	—		—	—	—	—	—	—	—	—	
H'23	—		—	—	—	—	—	—	—	—	
H'24	—		—	—	—	—	—	—	—	—	
H'25	—		—	—	—	—	—	—	—	—	
H'26	—		—	—	—	—	—	—	—	—	
H'27	—		—	—	—	—	—	—	—	—	
H'28	—		—	—	—	—	—	—	—	—	
H'29	—		—	—	—	—	—	—	—	—	
H'2A	—		—	—	—	—	—	—	—	—	
H'2B	—		—	—	—	—	—	—	—	—	
H'2C	—		—	—	—	—	—	—	—	—	
H'2D	—		—	—	—	—	—	—	—	—	
H'2E	—		—	—	—	—	—	—	—	—	
H'2F	—		—	—	—	—	—	—	—	—	
H'30	—		—	—	—	—	—	—	—	—	
H'31	—		—	—	—	—	—	—	—	—	
H'32	—		—	—	—	—	—	—	—	—	
H'33	—		—	—	—	—	—	—	—	—	
H'34	—		—	—	—	—	—	—	—	—	
H'35	—		—	—	—	—	—	—	—	—	
H'36	—		—	—	—	—	—	—	—	—	
H'37	—		—	—	—	—	—	—	—	—	
H'38	—		—	—	—	—	—	—	—	—	
H'39	—		—	—	—	—	—	—	—	—	
H'3A	—		—	—	—	—	—	—	—	—	
H'3B	—		—	—	—	—	—	—	—	—	
H'3C	—		—	—	—	—	—	—	—	—	
H'3D	—		—	—	—	—	—	—	—	—	
H'3E	—		—	—	—	—	—	—	—	—	
H'3F	—		—	—	—	—	—	—	—	—	

Address (low)	Register Name	Data Bus Width	Bit Names								Module Name
			Bit 7	Bit 6	Bit 5	Bit 4	Bit 3	Bit 2	Bit 1	Bit 0	
H'40	FLMCR	8	FWE	SWE	ESU	PSU	EV	PV	E	P	Flash memory
H'41	—		—	—	—	—	—	—	—	—	
H'42	EBR	8	EB7	EB6	EB5	EB4	EB3	EB2	EB1	EB0	
H'43	—		—	—	—	—	—	—	—	—	
H'44	—		—	—	—	—	—	—	—	—	
H'45	—		—	—	—	—	—	—	—	—	
H'46	—		—	—	—	—	—	—	—	—	
H'47	RAMCR	8	—	—	—	—	RAMS	RAM2	RAM1	—	
H'48	—		—	—	—	—	—	—	—	—	
H'49	—		—	—	—	—	—	—	—	—	
H'4A	—		—	—	—	—	—	—	—	—	
H'4B	—		—	—	—	—	—	—	—	—	
H'4C	—		—	—	—	—	—	—	—	—	
H'4D	FLMSR	8	FLER	—	—	—	—	—	—	—	
H'4E	—		—	—	—	—	—	—	—	—	
H'4F	—		—	—	—	—	—	—	—	—	
H'50	—		—	—	—	—	—	—	—	—	
H'51	—		—	—	—	—	—	—	—	—	
H'52	—		—	—	—	—	—	—	—	—	
H'53	—		—	—	—	—	—	—	—	—	
H'54	—		—	—	—	—	—	—	—	—	
H'55	—		—	—	—	—	—	—	—	—	
H'56	—		—	—	—	—	—	—	—	—	
H'57	—		—	—	—	—	—	—	—	—	
H'58	—		—	—	—	—	—	—	—	—	
H'59	—		—	—	—	—	—	—	—	—	
H'5A	—		—	—	—	—	—	—	—	—	
H'5B	—		—	—	—	—	—	—	—	—	
H'5C	—		—	—	—	—	—	—	—	—	
H'5D	DIVCR	8	—	—	—	—	—	—	DIV1	DIV0	System control
H'5E	MSTCR	8	PSTOP	—	MSTOP5	MSTOP4	MSTOP3	—	—	MSTOP0	
H'5F	—		—	—	—	—	—	—	—	—	

Address (low)	Register Name	Data Bus Width	Bit Names								Module Name
			Bit 7	Bit 6	Bit 5	Bit 4	Bit 3	Bit 2	Bit 1	Bit 0	
H'60	TSTR	8	—	—	—	STR4	STR3	STR2	STR1	STR0	ITU (all channels)
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	ITU channel 0
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	ITU channel 1
H'6F	TIOR1	8	—	IOB2	IOB1	IOB0	—	IOA2	IOA1	IOA0	
H'70	TIER1	8	—	—	—	—	—	OVIE	IMIEB	IMIEA	
H'71	TSR1	8	—	—	—	—	—	OVF	IMFB	IMFA	
H'72	TCNT1H	16									
H'73	TCNT1L										
H'74	GRA1H	16									
H'75	GRA1L										
H'76	GRB1H	16									
H'77	GRB1L										
H'78	TCR2	8	—	CCLR1	CCLR0	CKEG1	CKEG0	TPSC2	TPSC1	TPSC0	ITU channel 2
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										

Address (low)	Register Name	Data Bus Width	Bit Names								Module Name
			Bit 7	Bit 6	Bit 5	Bit 4	Bit 3	Bit 2	Bit 1	Bit 0	
H'82	TCR3	8	—	CCLR1	CCLR0	CKEG1	CKEG0	TPSC2	TPSC1	TPSC0	ITU channel 3
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'90	TOER	8	—	—	EXB4	EXA4	EB3	EB4	EA4	EA3	ITU (all channel)
H'91	TOCR	8	—	—	—	XTGD	—	—	OLS4	OLS3	
H'92	TCR4	8	—	CCLR1	CCLR0	CKEG1	CKEG0	TPSC2	TPSC1	TPSC0	ITU channel 4
H'93	TIOR4	8	—	IOB2	IOB1	IOB0	—	IOA2	IOA1	IOA0	
H'94	TIER4	8	—	—	—	—	—	OVIE	IMIEB	IMIEA	
H'95	TSR4	8	—	—	—	—	—	OVF	IMFB	IMFA	
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										

Address (low)	Register Name	Data Bus Width	Bit Names								Module Name
			Bit 7	Bit 6	Bit 5	Bit 4	Bit 3	Bit 2	Bit 1	Bit 0	
H'A0	TPMR	8	—	—	—	—	G3NOV	G2NOV	G1NOV	G0NOV	TPC
H'A1	TPCR	8	G3CMS1	G3CMS0	G2CMS1	G2CMS0	G1CMS1	G1CMS0	G0CMS1	G0CMS0	
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	WDT
H'A9	TCNT*[2]	8									
H'AA	—		—	—	—	—	—	—	—	—	
H'AB	RSTCSR*[2]	8	WRST	RSTOE	—	—	—	—	—	—	
H'AC	—		—	—	—	—	—	—	—	—	
H'AD	—		—	—	—	—	—	—	—	—	
H'AE	—		—	—	—	—	—	—	—	—	
H'AF	—		—	—	—	—	—	—	—	—	
H'B0	SMR	8	C/$\overline{A}$	CHR	PE	O/$\overline{E}$	STOP	MP	CKS1	CKS0	SCI0
H'B1	BRR	8									
H'B2	SCR	8	TIE	RIE	TE	RE	MPIE	TEIE	CKE1	CKE0	
H'B3	TDR	8									
H'B4	SSR	8	TDRE	RDRF	ORER	FER	PER	TEND	MPB	MPBT	
H'B5	RDR	8									
H'B6	SCMR	8	—	—	—	—	SDIR	SINV	—	SMIF	Smart card interface
H'B7											
H'B8	SMR	8	C/$\overline{A}$	CHR	PE	O/$\overline{E}$	STOP	MP	CKS1	CKS0	SCI1
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	—		—	—	—	—	—	—	—	—	

Address (low)	Register Name	Data Bus Width	Bit Names								Module Name
			Bit 7	Bit 6	Bit 5	Bit 4	Bit 3	Bit 2	Bit 1	Bit 0	
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	Port 1
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	Port 2
H'C2	P1DR	8	$P1_7$	$P1_6$	$P1_5$	$P1_4$	$P1_3$	$P1_2$	$P1_1$	$P1_0$	Port 1
H'C3	P2DR	8	$P2_7$	$P2_6$	$P2_5$	$P2_4$	$P2_3$	$P2_2$	$P2_1$	$P2_0$	Port 2
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	Port 3
H'C5	—		—	—	—	—	—	—	—	—	
H'C6	P3DR	8	$P3_7$	$P3_6$	$P3_5$	$P3_4$	$P3_3$	$P3_2$	$P3_1$	$P3_0$	Port 3
H'C7	—		—	—	—	—	—	—	—	—	
H'C8	P5DDR	8	—	—	—	—	$P5_3$DDR	$P5_2$DDR	$P5_1$DDR	$P5_0$DDR	Port 5
H'C9	P6DDR	8	—	—	$P6_5$DDR	$P6_4$DDR	$P6_3$DDR	—	—	$P6_0$DDR	Port 6
H'CA	P5DR	8	—	—	—	—	$P5_3$	$P5_2$	$P5_1$	$P5_0$	Port 5
H'CB	P6DR	8	—	—	$P6_5$	$P6_4$	$P6_3$	—	—	$P6_0$	Port 6
H'CC	—		—	—	—	—	—	—	—	—	
H'CD	P8DDR	8	—	—	—	—	—	—	$P8_1$DDR	$P8_0$DDR	Port 8
H'CE	P7DR	8	$P7_7$	$P7_6$	$P7_5$	$P7_4$	$P7_3$	$P7_2$	$P7_1$	$P7_0$	Port 7
H'CF	P8DR	8	—	—	—	—	—	—	$P8_1$	$P8_0$	Port 8
H'D0	P9DDR	8	—	—	$P9_5$DDR	$P9_4$DDR	$P9_3$DDR	$P9_2$DDR	$P9_1$DDR	$P9_0$DDR	Port 9
H'D1	PADDR	8	PA_7DDR	PA_6DDR	PA_5DDR	PA_4DDR	PA_3DDR	PA_2DDR	PA_1DDR	PA_0DDR	Port A
H'D2	P9DR	8	—	—	$P9_5$	$P9_4$	$P9_3$	$P9_2$	$P9_1$	$P9_0$	Port 9
H'D3	PADR	8	PA_7	PA_6	PA_5	PA_4	PA_3	PA_2	PA_1	PA_0	Port A
H'D4	PBDDR	8	PB_7DDR	—	PB_5DDR	PB_4DDR	PB_3DDR	PB_2DDR	PB_1DDR	PB_0DDR	Port B
H'D5	—		—	—	—	—	—	—	—	—	
H'D6	PBDR	8	PB_7	—	PB_5	PB_4	PB_3	PB_2	PB_1	PB_0	Port B
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	Port 2
H'D9	—		—	—	—	—	—	—	—	—	
H'DA	—		—	—	—	—	—	—	—	—	
H'DB	P5PCR	8	—	—	—	—	$P5_3$PCR	$P5_2$PCR	$P5_1$PCR	$P5_0$PCR	Port 5
H'DC	—		—	—	—	—	—	—	—	—	
H'DD	—		—	—	—	—	—	—	—	—	
H'DE	—		—	—	—	—	—	—	—	—	
H'DF	—		—	—	—	—	—	—	—	—	

Address (low)	Register Name	Data Bus Width	Bit Names								Module Name
			Bit 7	Bit 6	Bit 5	Bit 4	Bit 3	Bit 2	Bit 1	Bit 0	
H'E0	ADDRAH	8	AD9	AD8	AD7	AD6	AD5	AD4	AD3	AD2	A/D
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	—		—	—	—	—	—	—	—	—	Bus controller
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'F0	—		—	—	—	—	—	—	—	—	
H'F1	MDCR	8	—	—	—	—	—	MDS2	MDS1	MDS0	System control
H'F2	SYSCR	8	SSBY	STS2	STS1	STS0	UE	NMIEG	—	RAME	
H'F3	ADRCR	8	A_{23}E	A_{22}E	A_{21}E	—	—	—	—	—	Bus controller
H'F4	ISCR	8	—	—	IRQ5SC	IRQ4SC	—	—	IRQ1SC	IRQ0SC	Interrupt controller
H'F5	IER	8	—	—	IRQ5E	IRQ4E	—	—	IRQ1E	IRQ0E	
H'F6	ISR	8	—	—	IRQ5F	IRQ4F	—	—	IRQ1F	IRQ0F	
H'F7	—		—	—	—	—	—	—	—	—	
H'F8	IPRA	8	IPRA7	IPRA6	—	IPRA4	IPRA3	IPRA2	IPRA1	IPRA0	
H'F9	IPRB	8	IPRB7	IPRB6	—	—	IPRB3	IPRB2	IPRB1	—	
H'FA	—		—	—	—	—	—	—	—	—	
H'FB	—		—	—	—	—	—	—	—	—	
H'FC											
H'FD	—		—	—	—	—	—	—	—	—	
H'FE	—		—	—	—	—	—	—	—	—	
H'FF	—		—	—	—	—	—	—	—	—	

Legend:

ITU: 16-bit integrated timer unit

TPC: Programmable timing pattern controller

WDT: Watchdog timer

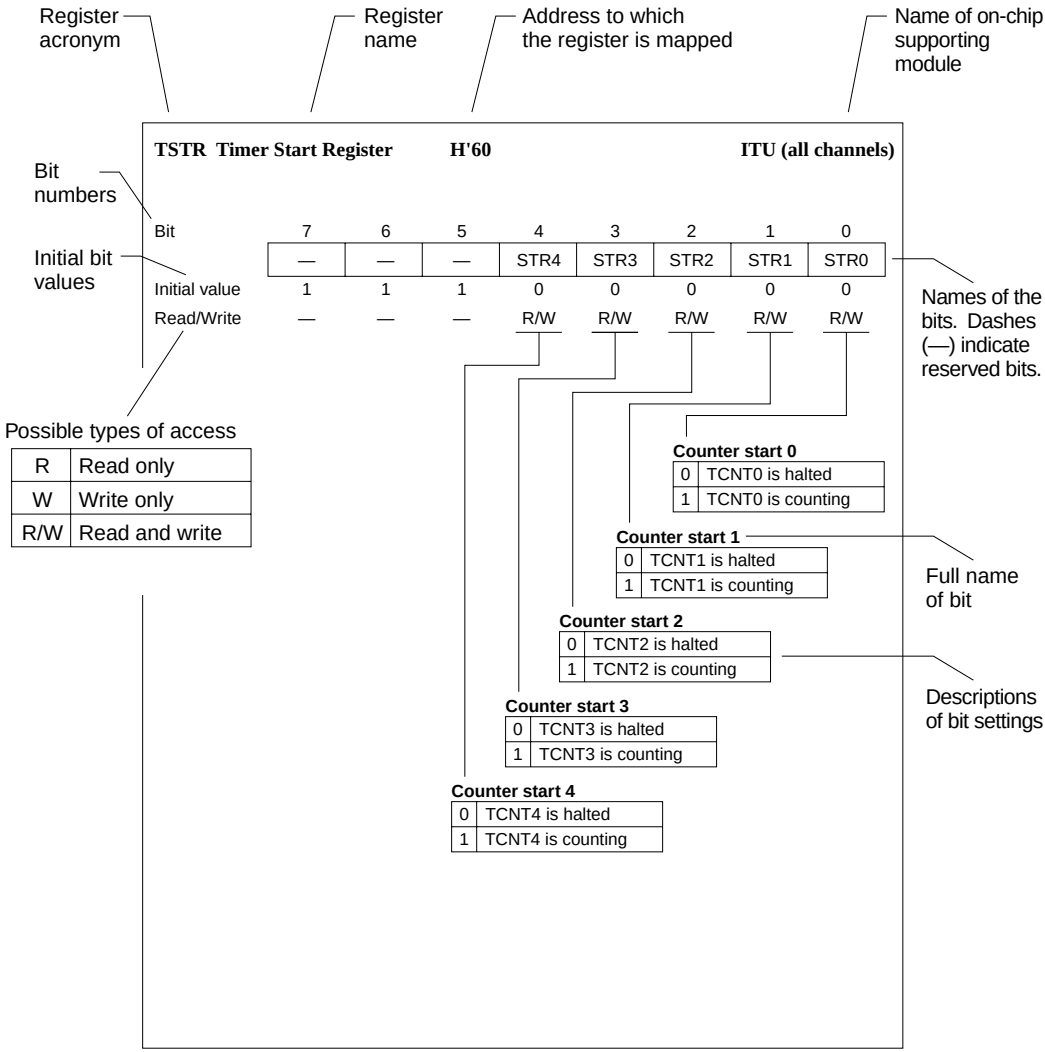
SCI: Serial communication interface

A/D: A/D converter

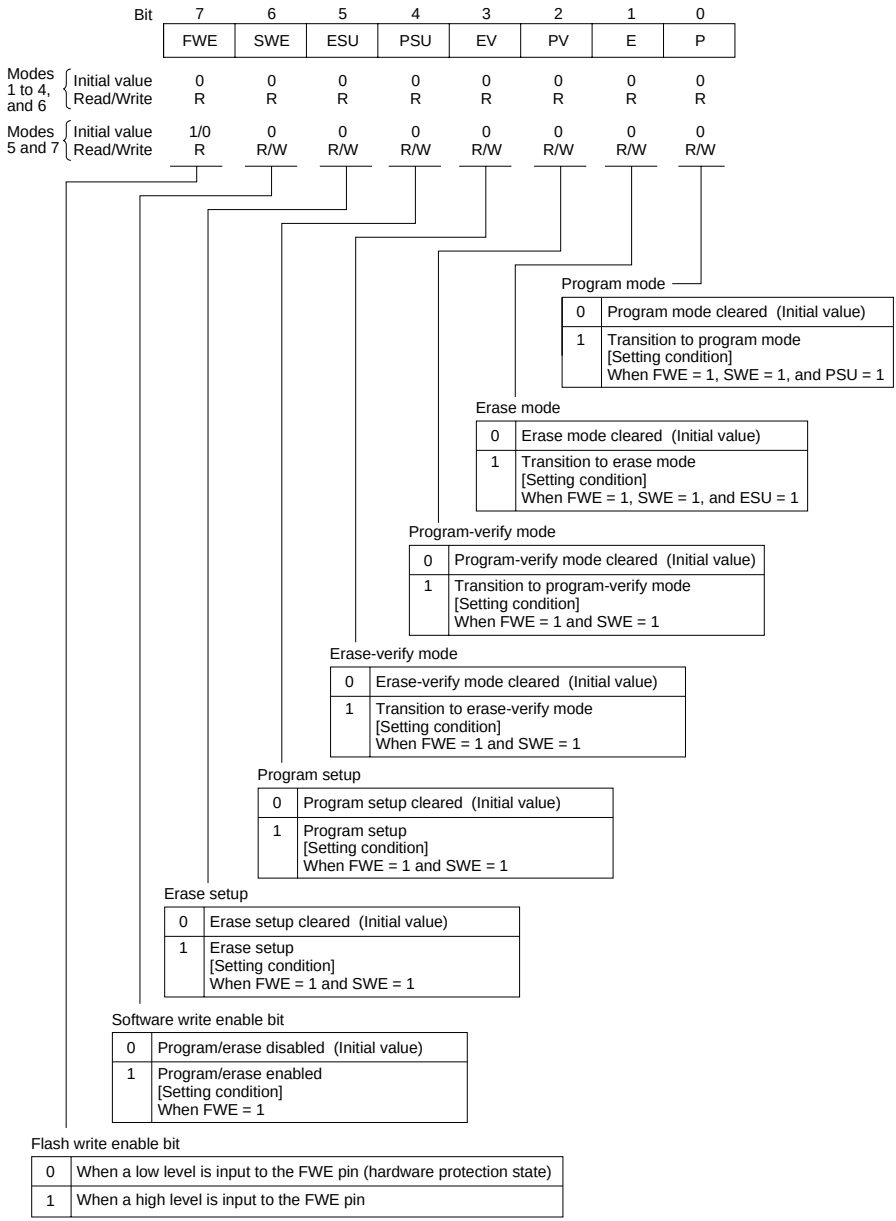
Notes: 1. The address depends on the output trigger setting.

2. For write access to TCSR, TCNT, and RSTCSR, see section 10.2.4, Notes on Register Access.

B.2 Function


Register acronym
Register name
Address to which the register is mapped
Name of on-chip supporting module
TSTR Timer Start Register
H'60
ITU (all channels)
Bit numbers
Initial bit values
Bit
7 6 5 4 3 2 1 0
— — — 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
Names of the bits. Dashes (—) indicate reserved bits.
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
Full name of bit
Counter start 2
0 TCNT2 is halted
1 TCNT2 is counting
Descriptions of bit settings
Counter start 3
0 TCNT3 is halted
1 TCNT3 is counting
Counter start 4
0 TCNT4 is halted
1 TCNT4 is counting

FLMCR—Flash Memory Control Register H'40 Flash memory



Note: This register is used only in the flash memory versions.
Reading the corresponding address in a mask ROM version will always return 1s, and writes to this address are disabled.
Fix the FWE pin low in mode 6.

EBR—Erase Block Register H'42 Flash memory

Bit		7	6	5	4	3	2	1	0
		EB7	EB6	EB5	EB4	EB3	EB2	EB1	EB0
Modes 1 to 4, and 6	Initial value	0	0	0	0	0	0	0	0
	Read/Write	R	R	R	R	R	R	R	R
Modes 5 to 7	Initial value	0	0	0	0	0	0	0	0
	Read/Write	R/W	R/W	R/W	R/W	R/W	R/W	R/W	R/W

Block 7 to 0

0	Block EB7 to EB0 is not selected (Initial value)
1	Block EB7 to EB0 is selected

Note: When not erasing flash memory, EBR should be cleared to H'00.
This register is used only in the flash memory versions. Reading the corresponding address in a mask ROM version will always return 1s, and writes to this address are disabled.
1s cannot be set in this register in mode 6.

RAMCR—RAM Control Register **H'47** **Flash Memory**

Bit		7	6	5	4	3	2	1	0
		—	—	—	—	RAMS	RAM2	RAM1	—
Modes 1 to 4	Initial value	1	1	1	1	0	0	0	1
	Read/Write	—	—	—	—	R	R	R	—
Modes 5 to 7	Initial value	1	1	1	1	0	0	0	1
	Read/Write	—	—	—	—	R/W*	R/W*	R/W*	—

Reserved bits

RAM select, RAM2, RAM1

Bit 3	Bit 2	Bit 1	RAM Area	RAM Emulation Status
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: This register is used only in the flash memory versions. Reading the corresponding address in a mask ROM version will always return 1s, and writes to this address are disabled.

* In mode 6 (single-chip normal mode), flash memory emulation by RAM is not supported; these bits can be modified, but must not be set to 1.

FLMSR—Flash Memory Status Register **H'4D** **Flash memory**

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

Flash memory error

0	Flash memory write/erase protection is disabled (Initial value)
1	An error has occurred during flash memory writing/erasing Flash memory error protection is enabled

DIVCR—Division Control Register **H'5D** **System control**

Bit	7	6	5	7	3	2	1	0
	—	—	—	—	—	—	DIV1	DIV0
Initial value	1	1	1	1	1	1	0	0
Read/Write	—	—	—	—	—	—	R/W	R/W

Divide bits 1 and 0

Bit 1	Bit 0	Frequency Division Ratio
DIV1	DIV0	
0	0	1/1initial value
	1	1/2
1	0	1/4
	1	1/8

MSTCR—Module Standby Control Register **H'5E** **System control**

Bit	7	6	5	4	3	2	1	0
	PSTOP	—	MSTOP5	MSTOP4	MSTOP3	—	—	MSTOP0
Initial value	0	1	0	0	0	0	0	0
Read/Write	R/W	—	R/W	R/W	R/W	R/W	R/W	R/W

Module standby 0

0	A/D converter operates normally
1	A/D converter is in standby state

Module standby 3

0	SCI1 operates normally
1	SCI1 is in standby state

Module standby 4

0	SCI0 operates normally
1	SCI0 is in standby state

Module standby 5

0	ITU operates normally
1	ITU is in standby state

ϕ clock stop

0	ϕ clock output is enabled
1	ϕ clock output is disabled

TSTR—Timer Start Register **H'60** **ITU (all channels)**

Bit	7	6	5	4	3	2	1	0
	—	—	—	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

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

TSNC—Timer Synchro Register **H'61** **ITU (all channels)**

Bit	7	6	5	4	3	2	1	0
	—	—	—	SYNC4	SYNC3	SYNC2	SYNC1	SYNC0
Initial value	1	1	1	0	0	0	0	0
Read/Write	—	—	—	R/W	R/W	R/W	R/W	R/W

Timer sync 0

0	TCNT0 operates independently
1	TCNT0 is synchronized

Timer sync 1

0	TCNT1 operates independently
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

TMDR—Timer Mode Register **H'62** **ITU (all channels)**

Bit	7	6	5	4	3	2	1	0
	—	MDF	FDIR	PWM4	PWM3	PWM2	PWM1	PWM0
Initial value	1	0	0	0	0	0	0	0
Read/Write	—	R/W	R/W	R/W	R/W	R/W	R/W	R/W

PWM mode 0

0	Channel 0 operates normally
1	Channel 0 operates in PWM mode

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

TFCR—Timer Function Control Register H'63 ITU (all channels)

Bit	7	6	5	4	3	2	1	0
	—	—	CMD1	CMD0	BFB4	BFA4	BFB3	BFA3
Initial value	1	1	0	0	0	0	0	0
Read/Write	—	—	R/W	R/W	R/W	R/W	R/W	R/W

Buffer mode A3

0	GRA3 operates normally
1	GRA3 is buffered by BRA3

Buffer mode B3

0	GRB3 operates normally
1	GRB3 is buffered by BRB3

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	Operating Mode of Channels 3 and 4
CMD1	CMD0	
0	0	Channels 3 and 4 operate normally
	1	
1	0	Channels 3 and 4 operate together in complementary PWM mode
	1	Channels 3 and 4 operate together in reset-synchronized PWM mode

TCR0—Timer Control Register 0 **H'64** **ITU0**

Bit	7	6	5	4	3	2	1	0
	—	CCLR1	CCLR0	CKEG1	CKEG0	TPSC2	TPSC1	TPSC0
Initial value	1	0	0	0	0	0	0	0
Read/Write	—	R/W	R/W	R/W	R/W	R/W	R/W	R/W

Timer prescaler 2 to 0

Bit 2	Bit 1	Bit 0	TCNT Clock Source
TPSC2	TPSC1	TPSC0	
0	0	0	Internal clock: φ
		1	Internal clock: φ/2
	1	0	Internal clock: φ/4
		1	Internal clock: φ/8
1	0	0	External clock A: TCLKA input
		1	External clock B: TCLKB input
	1	0	External clock C: TCLKC input
		1	External clock D: TCLKD input

Clock edge 1 and 0

Bit 4	Bit 3	Counted Edges of External Clock
CKEG1	CKEG0	
0	0	Rising edges counted
	1	Falling edges counted
1	—	Both edges counted

Counter clear 1 and 0

Bit 6	Bit 5	TCNT Clear Source
CCLR1	CCLR0	
0	0	TCNT is not cleared
	1	TCNT is cleared by GRA compare match or input capture
1	0	TCNT is cleared by GRB compare match or input capture
	1	Synchronous clear: TCNT is cleared in synchronization with other synchronized timers

TIOR0—Timer I/O Control Register 0 **H'65** **ITU0**

Bit	7	6	5	4	3	2	1	0
	—	IOB2	IOB1	IOB0	—	IOA2	IOA1	IOA0
Initial value	1	0	0	0	1	0	0	0
Read/Write	—	R/W	R/W	R/W	—	R/W	R/W	R/W

I/O control A2 to A0

Bit 2	Bit 1	Bit 0	GRA Function	
IOA2	IOA1	IOA0		
0	0	0	GRA is an output compare register	No output at compare match
		1		0 output at GRA compare match
	1	0		1 output at GRA compare match
		1		Output toggles at GRA compare match
1	0	0	GRA is an input capture register	GRA captures rising edge of input
		1		GRA captures falling edge of input
	1	0		GRA captures both edges of input
		1		

I/O control B2 to B0

Bit 6	Bit 5	Bit 4	GRB Function	
IOB2	IOB1	IOB0		
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 match
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		

TIER0—Timer Interrupt Enable Register 0 **H'66** **ITU0**

Bit	7	6	5	4	3	2	1	0
	—	—	—	—	—	OVIE	IMIEB	IMIEA
Initial value	1	1	1	1	1	0	0	0
Read/Write	—	—	—	—	—	R/W	R/W	R/W

Input capture/compare match interrupt enable A

0	IMIA interrupt requested by IMFA is disabled
1	IMIA interrupt requested by IMFA is enabled

Input capture/compare match interrupt enable B

0	IMIB interrupt requested by IMFB is disabled
1	IMIB interrupt requested by IMFB is enabled

Overflow interrupt enable

0	OVI interrupt requested by OVF is disabled
1	OVI interrupt requested by OVF is enabled

TSR0—Timer Status Register 0 **H'67** **ITU0**

Bit	7	6	5	4	3	2	1	0
	—	—	—	—	—	OVF	IMFB	IMFA
Initial value	1	1	1	1	1	0	0	0
Read/Write	—	—	—	—	—	R/(W)*	R/(W)*	R/(W)*

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 a compare match register. • TCNT value is transferred to GRA by an input capture signal, when GRA functions as an input capture register.

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 a compare match 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

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

TCNT0 H/L—Timer Counter 0 H/L **H'68, H'69** **ITU0**

Bit	15	14	13	12	11	10	9	8	7	6	5	4	3	2	1	0
Initial value	0	0	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	R/W	R/W

Up-counter

GRA0 H/L—General Register A0 H/L **H'6A, H'6B** **ITU0**

Bit	15	14	13	12	11	10	9	8	7	6	5	4	3	2	1	0
Initial value	1	1	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	R/W	R/W

Output compare or input capture register

GRB0 H/L—General Register B0 H/L **H'6C, H'6D** **ITU0**

Bit	15	14	13	12	11	10	9	8	7	6	5	4	3	2	1	0
Initial value	1	1	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	R/W	R/W

Output compare or input capture register

TCR1—Timer Control Register 1 **H'6E** **ITU1**

Bit	7	6	5	4	3	2	1	0
	—	CCLR1	CCLR0	CKEG1	CKEG0	TPSC2	TPSC1	TPSC0
Initial value	1	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 ITU0.

TIOR1—Timer I/O Control Register 1 **H'6F** **ITU1**

Bit	7	6	5	4	3	2	1	0
	—	IOB2	IOB1	IOB0	—	IOA2	IOA1	IOA0
Initial value	1	0	0	0	1	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** **ITU1**

Bit	7	6	5	4	3	2	1	0
	—	—	—	—	—	OVIE	IMIEB	IMIEA
Initial value	1	1	1	1	1	0	0	0
Read/Write	—	—	—	—	—	R/W	R/W	R/W

Note: Bit functions are the same as for ITU0.

TSR1—Timer Status Register 1 **H'71** **ITU1**

Bit	7	6	5	4	3	2	1	0
	—	—	—	—	—	OVF	IMFB	IMFA
Initial value	1	1	1	1	1	0	0	0
Read/Write	—	—	—	—	—	R/(W)*	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** **ITU1**

Bit	15	14	13	12	11	10	9	8	7	6	5	4	3	2	1	0
Initial value	0	0	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	R/W	R/W

Note: Bit functions are the same as for ITU0.

GRA1 H/L—General Register A1 H/L **H'74, H'75** **ITU1**

Bit	15	14	13	12	11	10	9	8	7	6	5	4	3	2	1	0
Initial value	1	1	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	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** **ITU1**

Bit	15	14	13	12	11	10	9	8	7	6	5	4	3	2	1	0
Initial value	1	1	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	R/W	R/W

Note: Bit functions are the same as for ITU0.

TCR2—Timer Control Register 2 **H'78** **ITU2**

Bit	7	6	5	4	3	2	1	0
	—	CCLR1	CCLR0	CKEG1	CKEG0	TPSC2	TPSC1	TPSC0
Initial value	1	0	0	0	0	0	0	0
Read/Write	—	R/W	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 selection by bits CKEG1, CKEG0 and TPSC2 to TPSC0 is ignored.

TIOR2—Timer I/O Control Register 2 **H'79** **ITU2**

Bit	7	6	5	4	3	2	1	0
	—	IOB2	IOB1	IOB0	—	IOA2	IOA1	IOA0
Initial value	1	0	0	0	1	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. Channel 2 does not have a compare match toggle output function. If this setting is used, 1 output will be selected automatically.

TIER2—Timer Interrupt Enable Register 2 **H'7A** **ITU2**

Bit	7	6	5	4	3	2	1	0
	—	—	—	—	—	OVIE	IMIEB	IMIEA
Initial value	1	1	1	1	1	0	0	0
Read/Write	—	—	—	—	—	R/W	R/W	R/W

Note: Bit functions are the same as for ITU0.

TSR2—Timer Status Register 2 **H'7B** **ITU2**

Bit	7	6	5	4	3	2	1	0
	—	—	—	—	—	OVF	IMFB	IMFA
Initial value	1	1	1	1	1	0	0	0
Read/Write	—	—	—	—	—	R/(W)*	R/(W)*	R/(W)*

Bit functions are the same as for ITU0

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.

TCNT2 H/L—Timer Counter 2 H/L **H'7C, H'7D** **ITU2**

Bit	15	14	13	12	11	10	9	8	7	6	5	4	3	2	1	0
Initial value	0	0	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	R/W	R/W

Phase counting mode: up/down-counter
Other modes: up-counter

GRA2 H/L—General Register A2 H/L **H'7E, H'7F** **ITU2**

Bit	15	14	13	12	11	10	9	8	7	6	5	4	3	2	1	0
Initial value	1	1	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	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** **ITU2**

Bit	15	14	13	12	11	10	9	8	7	6	5	4	3	2	1	0
Initial value	1	1	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	R/W	R/W

Note: Bit functions are the same as for ITU0.

TCR3—Timer Control Register 3 **H'82** **ITU3**

Bit	7	6	5	4	3	2	1	0
	—	CCLR1	CCLR0	CKEG1	CLEG0	TPSC2	TPSC1	TPSC0
Initial value	1	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 ITU0.

TIOR3—Timer I/O Control Register 3 **H'83** **ITU3**

Bit	7	6	5	4	3	2	1	0
	—	IOB2	IOB1	IOB0	—	IOA2	IOA1	IOA0
Initial value	1	0	0	0	1	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.

TIER3—Timer Interrupt Enable Register 3 **H'84** **ITU3**

Bit	7	6	5	4	3	2	1	0
	—	—	—	—	—	OVIE	IMIEB	IMIEA
Initial value	1	1	1	1	1	0	0	0
Read/Write	—	—	—	—	—	R/W	R/W	R/W

Note: Bit functions are the same as for ITU0.

TSR3—Timer Status Register 3 **H'85** **ITU3**

Bit	7	6	5	4	3	2	1	0
	—	—	—	—	—	OVF	IMFB	IMFA
Initial value	1	1	1	1	1	0	0	0
Read/Write	—	—	—	—	—	R/(W)*	R/(W)*	R/(W)*

Bit functions are the same as for ITU0

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.

TCNT3 H/L—Timer Counter 3 H/L **H'86, H'87** **ITU3**

Bit	15	14	13	12	11	10	9	8	7	6	5	4	3	2	1	0
Initial value	0	0	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	R/W	R/W

Complementary PWM mode: up/down counter
Other modes: up-counter

GRA3 H/L—General Register A3 H/L **H'88, H'89** **ITU3**

Bit	15	14	13	12	11	10	9	8	7	6	5	4	3	2	1	0
Initial value	1	1	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	R/W	R/W

Output compare or input capture register (can be buffered)

GRB3 H/L—General Register B3 H/L **H'8A, H'8B** **ITU3**

Bit	15	14	13	12	11	10	9	8	7	6	5	4	3	2	1	0
Initial value	1	1	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	R/W	R/W

Output compare or input capture register (can be buffered)

BRA3 H/L—Buffer Register A3 H/L **H'8C, H'8D** **ITU3**

Bit	15	14	13	12	11	10	9	8	7	6	5	4	3	2	1	0
Initial value	1	1	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	R/W	R/W

Used to buffer GRA

BRB3 H/L—Buffer Register B3 H/L **H'8E, H'8F** **ITU3**

Bit	15	14	13	12	11	10	9	8	7	6	5	4	3	2	1	0
Initial value	1	1	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	R/W	R/W

Used to buffer GRB

TOER—Timer Output Enable Register **H'90** **ITU (all channels)**

Bit	7	6	5	4	3	2	1	0
	—	—	EXB4	EXA4	EB3	EB4	EA4	EA3
Initial value	1	1	1	1	1	1	1	1
Read/Write	—	—	R/W	R/W	R/W	R/W	R/W	R/W

Master enable $TIOCA_3$

0	$TIOCA_3$ output is disabled regardless of TIOR3, TMDR, and TFCR settings
1	$TIOCA_3$ is enabled for output according to TIOR3, TMDR, and TFCR settings

Master enable $TIOCA_4$

0	$TIOCA_4$ output is disabled regardless of TIOR4, TMDR, and TFCR settings
1	$TIOCA_4$ is enabled for output according to TIOR4, TMDR, and TFCR settings

Master enable $TIOCB_4$

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 $TIOCB_3$

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 $TOCXA_4$

0	$TOCXA_4$ output is disabled regardless of TFCR settings
1	$TOCXA_4$ is enabled for output according to TFCR settings

Master enable $TOCXB_4$

0	$TOCXB_4$ output is disabled regardless of TFCR settings
1	$TOCXB_4$ is enabled for output according to TFCR settings

TOCR—Timer Output Control Register **H'91** **ITU (all channels)**

Bit	7	6	5	4	3	2	1	0
	—	—	—	XTGD	—	—	OLS4	OLS3
Initial value	1	1	1	1	1	1	1	1
Read/Write	—	—	—	R/W	—	—	R/W	R/W

Output level select 3

0	$TIOCB_3$, $TOCXA_4$, and $TOCXB_4$ outputs are inverted
1	$TIOCB_3$, $TOCXA_4$, and $TOCXB_4$ outputs are not inverted

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, disabling ITU output.

TCR4—Timer Control Register 4 **H'92** **ITU4**

Bit	7	6	5	4	3	2	1	0
	—	CCLR1	CCLR0	CKEG1	CKEG0	TPSC2	TPSC1	TPSC0
Initial value	1	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 ITU0.

TIOR4—Timer I/O Control Register 4 **H'93** **ITU4**

Bit	7	6	5	4	3	2	1	0
	—	IOB2	IOB1	IOB0	—	IOA2	IOA1	IOA0
Initial value	1	0	0	0	1	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.

TIER4—Timer Interrupt Enable Register 4 **H'94** **ITU4**

Bit	7	6	5	4	3	2	1	0
	—	—	—	—	—	OVIE	IMIEB	IMIEA
Initial value	1	1	1	1	1	0	0	0
Read/Write	—	—	—	—	—	R/W	R/W	R/W

Note: Bit functions are the same as for ITU0.

TSR4—Timer Status Register 4 **H'95** **ITU4**

Bit	7	6	5	4	3	2	1	0
	—	—	—	—	—	OVF	IMFB	IMFA
Initial value	1	1	1	1	1	0	0	0
Read/Write	—	—	—	—	—	R/(W)*	R/(W)*	R/(W)*

Notes: Bit functions are the same as for ITU0.
* Only 0 can be written to clear the flag.

TCNT4 H/L—Timer Counter 4 H/L **H'96, H'97** **ITU4**

Bit	15	14	13	12	11	10	9	8	7	6	5	4	3	2	1	0
Initial value	0	0	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	R/W	R/W

Note: Bit functions are the same as for ITU3.

GRA4 H/L—General Register A4 H/L **H'98, H'99** **ITU4**

Bit	15	14	13	12	11	10	9	8	7	6	5	4	3	2	1	0
Initial value	1	1	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	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** **ITU4**

Bit	15	14	13	12	11	10	9	8	7	6	5	4	3	2	1	0
Initial value	1	1	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	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** **ITU4**

Bit	15	14	13	12	11	10	9	8	7	6	5	4	3	2	1	0
Initial value	1	1	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	R/W	R/W

Note: Bit functions are the same as for ITU3.

BRB4 H/L—Buffer Register B4 H/L **H'9E, H'9F** **ITU4**

Bit	15	14	13	12	11	10	9	8	7	6	5	4	3	2	1	0
Initial value	1	1	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	R/W	R/W

Note: Bit functions are the same as for ITU3.

TPMR—TPC Output Mode Register **H'A0** **TPC**

Bit	7	6	5	4	3	2	1	0
	—	—	—	—	G3NOV	G2NOV	G1NOV	G0NOV
Initial value	1	1	1	1	0	0	0	0
Read/Write	—	—	—	—	R/W	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 channel
1	Non-overlapping TPC output in group 0, controlled by compare match 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 channel
1	Non-overlapping TPC output in group 1, controlled by compare match 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

TPCR—TPC Output Control Register **H'A1** **TPC**

Bit	7	6	5	4	3	2	1	0
	G3CMS1	G3CMS0	G2CMS1	G2CMS0	G1CMS1	G1CMS0	G0CMS1	G0CMS0
Initial value	1	1	1	1	1	1	1	1
Read/Write	R/W	R/W	R/W	R/W	R/W	R/W	R/W	R/W

Group 0 compare match select 1 and 0

Bit 1	Bit 0	ITU Channel Selected as Output Trigger
G0CMS1	G0CMS0	
0	0	TPC output group 0 (TP_3 to TP_0) is triggered by compare match in ITU channel 0
	1	TPC output group 0 (TP_3 to TP_0) is triggered by compare match in ITU channel 1
1	0	TPC output group 0 (TP_3 to TP_0) is triggered by compare match in ITU channel 2
	1	TPC output group 0 (TP_3 to TP_0) is triggered by compare match in ITU channel 3

Group 1 compare match select 1 and 0

Bit 3	Bit 2	ITU Channel Selected as Output Trigger
G1CMS1	G1CMS0	
0	0	TPC output group 1 (TP_7 to TP_4) is triggered by compare match in ITU channel 0
	1	TPC output group 1 (TP_7 to TP_4) is triggered by compare match in ITU channel 1
1	0	TPC output group 1 (TP_7 to TP_4) is triggered by compare match in ITU channel 2
	1	TPC output group 1 (TP_7 to TP_4) is triggered by compare match in ITU channel 3

Group 2 compare match select 1 and 0

Bit 5	Bit 4	ITU Channel Selected as Output Trigger
G2CMS1	G2CMS0	
0	0	TPC output group 2 (TP_{11} to TP_8) is triggered by compare match in ITU channel 0
	1	TPC output group 2 (TP_{11} to TP_8) is triggered by compare match in ITU channel 1
1	0	TPC output group 2 (TP_{11} to TP_8) is triggered by compare match in ITU channel 2
	1	TPC output group 2 (TP_{11} to TP_8) is triggered by compare match in ITU channel 3

Group 3 compare match select 1 and 0

Bit 7	Bit 6	ITU Channel Selected as Output Trigger
G3CMS1	G3CMS0	
0	0	TPC output group 3 (TP_{15} to TP_{12})* is triggered by compare match in ITU channel 0
	1	TPC output group 3 (TP_{15} to TP_{12})* is triggered by compare match in ITU channel 1
1	0	TPC output group 3 (TP_{15} to TP_{12})* is triggered by compare match in ITU channel 2
	1	TPC output group 3 (TP_{15} to TP_{12})* is triggered by compare match in ITU channel 3

Note: * Since this LSI does not have a TP_{14} pin, the TP_{14} signal cannot be output off-chip.

NDERB—Next Data Enable Register B **H'A2** **TPC**

Bit	7	6	5	4	3	2	1	0
	NDER15	NDER14	NDER13	NDER12	NDER11	NDER10	NDER9	NDER8
Initial value	0	0	0	0	0	0	0	0
Read/Write	R/W	R/W	R/W	R/W	R/W	R/W	R/W	R/W

Next data enable 15 to 8

Bits 7 to 0	Description
NDER15 to NDER8	
0	TPC outputs TP_{15} to TP_8* are disabled (NDR15 to NDR8 are not transferred to PB_7 to PB_0)
1	TPC outputs TP_{15} to TP_8* are enabled (NDR15 to NDR8 are transferred to PB_7 to PB_0)

Note: * Since this LSI does not have a TP_{14} pin, the TP_{14} signal cannot be output off-chip.

NDERA—Next Data Enable Register A **H'A3** **TPC**

Bit	7	6	5	4	3	2	1	0
	NDER7	NDER6	NDER5	NDER4	NDER3	NDER2	NDER1	NDER0
Initial value	0	0	0	0	0	0	0	0
Read/Write	R/W	R/W	R/W	R/W	R/W	R/W	R/W	R/W

Next data enable 7 to 0

Bits 7 to 0	Description
NDER7 to NDER0	
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)

NDRB—Next Data Register B **H'A4/H'A6** **TPC**

- Same output trigger for TPC output groups 2 and 3

  Address H'FFA4

Bit	7	6	5	4	3	2	1	0
	NDR15	NDR14	NDR13	NDR12	NDR11	NDR10	NDR9	NDR8
Initial value	0	0	0	0	0	0	0	0
Read/Write	R/W	R/W	R/W	R/W	R/W	R/W	R/W	R/W

Next output data for TPC output group 3*

Next output data for TPC output group 2

  Address H'FFA6

Bit	7	6	5	4	3	2	1	0
	—	—	—	—	—	—	—	—
Initial value	1	1	1	1	1	1	1	1
Read/Write	—	—	—	—	—	—	—	—

- Different output triggers for TPC output groups 2 and 3

  Address H'FFA4

Bit	7	6	5	4	3	2	1	0
	NDR15	NDR14	NDR13	NDR12	—	—	—	—
Initial value	0	0	0	0	1	1	1	1
Read/Write	R/W	R/W	R/W	R/W	—	—	—	—

Next output data for TPC output group 3*

  Address H'FFA6

Bit	7	6	5	4	3	2	1	0
	—	—	—	—	NDR11	NDR10	NDR9	NDR8
Initial value	1	1	1	1	0	0	0	0
Read/Write	—	—	—	—	R/W	R/W	R/W	R/W

Next output data for TPC output group 2

Note: * Since this LSI does not have a TP_{14} pin, the TP_{14} signal cannot be output off-chip.

NDRA—Next Data Register A **H'A5/H'A7** **TPC**

- Same output trigger for TPC output groups 0 and 1

  Address H'FFA5

Bit	7	6	5	4	3	2	1	0
	NDR7	NDR6	NDR5	NDR4	NDR3	NDR2	NDR1	NDR0
Initial value	0	0	0	0	0	0	0	0
Read/Write	R/W	R/W	R/W	R/W	R/W	R/W	R/W	R/W
	Next output data for TPC output group 1				Next output data for TPC output group 0			

  Address H'FFA7

Bit	7	6	5	4	3	2	1	0
	—	—	—	—	—	—	—	—
Initial value	1	1	1	1	1	1	1	1
Read/Write	—	—	—	—	—	—	—	—

- Different output triggers for TPC output groups 0 and 1

  Address H'FFA5

Bit	7	6	5	4	3	2	1	0
	NDR7	NDR6	NDR5	NDR4	—	—	—	—
Initial value	0	0	0	0	1	1	1	1
Read/Write	R/W	R/W	R/W	R/W	—	—	—	—
	Next output data for TPC output group 1							

  Address H'FFA7

Bit	7	6	5	4	3	2	1	0
	—	—	—	—	NDR3	NDR2	NDR1	NDR0
Initial value	1	1	1	1	0	0	0	0
Read/Write	—	—	—	—	R/W	R/W	R/W	R/W
					Next output data for TPC output group 0			

TCSR—Timer Control/Status Register **H'A8** **WDT**

Bit	7	6	5	4	3	2	1	0
	OVF	WT/$\overline{IT}$	TME	—	—	CKS2	CKS1	CKS0
Initial value	0	0	0	1	1	0	0	0
Read/Write	R/(W)*	R/W	R/W	—	—	R/W	R/W	R/W

Clock select 2 to 0

CKS2	CKS1	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

Timer enable

0	TCNT is initialized to H'00 and halted
1	TCNT is counting

Timer mode select

0	Interval timer: requests interval timer interrupts
1	Watchdog timer: generates a reset signal

Overflow flag

0	[Clearing condition] Read OVF when OVF = 1, then write 0 in OVF
1	[Setting condition] TCNT changes from H'FF to H'00

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

TCNT—Timer Counter **H'A9 (read), H'A8 (write)** **WDT**

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

Count value

RSTCSR—Reset Control/Status Register **H'AB (read), H'AA (write)** **WDT**

Bit	7	6	5	4	3	2	1	0
	WRST	RSTOE	—	—	—	—	—	—
Initial value	0	0	1	1	1	1	1	1
Read/Write	R/(W)*	R/W	—	—	—	—	—	—

Reset output enable

0	Reset signal is not output externally
1	Reset signal is output externally

Watchdog timer reset

0	[Clearing condition] Reset signal input at $\overline{\text{RES}}$ pin, or 0 written by software
1	[Setting condition] TCNT overflow generates a reset signal

Note: * Only 0 can be written in bit 7 to clear the flag.

SMR—Serial Mode Register **H'B0** **SCI0**

Bit	7	6	5	7	3	2	1	0
	C/$\overline{A}$	CHR	PE	O/$\overline{E}$	STOP	MP	CKS1	CKS0
Initial value	0	0	0	0	0	0	0	0
Read/Write	R/W	R/W	R/W	R/W	R/W	R/W	R/W	R/W

Clock select 1 and 0

Bit 1	Bit 0	Clock Source
CKS1	CKS0	
0	0	φ clock
	1	φ/4 clock
1	0	φ/16 clock
	1	φ/64 clock

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

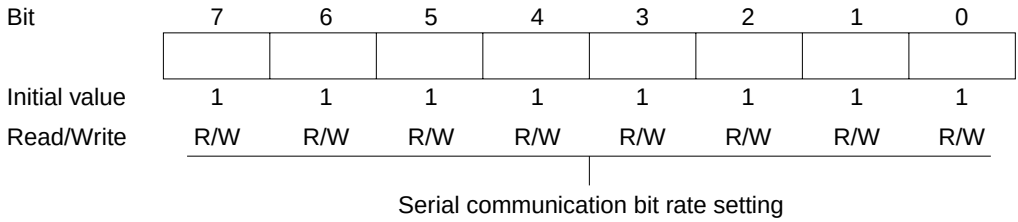
Character length

0	8-bit data
1	7-bit data

Communication mode

0	Asynchronous mode
1	Synchronous mode

BRR—Bit Rate Register **H'B1** **SCI0**



SCR—Serial Control Register **H'B2** **SCI0**

Bit	7	6	5	4	3	2	1	0
	TIE	RIE	TE	RE	MPIE	TEIE	CKE1	CKE0
Initial value	0	0	0	0	0	0	0	0
Read/Write	R/W	R/W	R/W	R/W	R/W	R/W	R/W	R/W

Clock enable 1 and 0

Bit 1	Bit 2	Clock Selection and Output	
CKE1	CKE2		
0	0	Asynchronous mode	Internal clock, SCK pin available for generic input/output
		Synchronous mode	Internal clock, SCK pin used for serial clock output
	1	Asynchronous mode	Internal clock, SCK pin used for clock output
		Synchronous mode	Internal clock, SCK pin used for serial clock output
1	0	Asynchronous mode	External clock, SCK pin used for clock input
		Synchronous mode	External clock, SCK pin used for serial clock input
	1	Asynchronous mode	External clock, SCK pin used for clock input
		Synchronous mode	External clock, SCK pin used for serial clock input

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

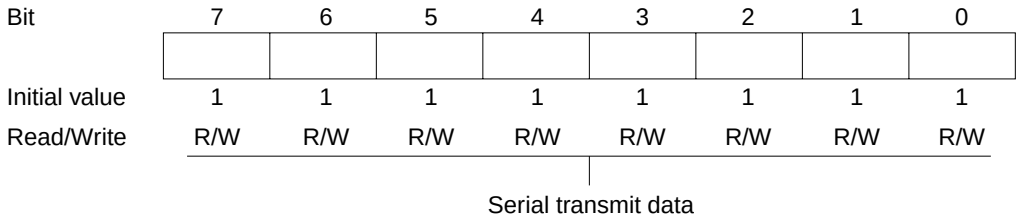
Receive interrupt enable

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

TDR—Transmit Data Register **H'B3** **SCI0**



SSR—Serial Status Register **H'B4** **SCI0**

Bit	7	6	5	4	3	2	1	0
	TDRE	RDRF	ORER	FER	PER	TEND	MPB	MPBT
Initial value	1	0	0	0	0	1	0	0
Read/Write	R/(W)*	R/(W)*	R/(W)*	R/(W)*	R/(W)*	R	R	R/W

Multiprocessor bit

0	Multiprocessor bit value in receive data is 0
1	Multiprocessor bit value in receive data is 1

Multiprocessor bit transfer

0	Multiprocessor bit value in transmit data is 0
1	Multiprocessor bit value in transmit data is 1

Transmit end

0	[Clearing conditions] Read TDRE when TDRE = 1, then write 0 in TDRE.
1	[Setting conditions] Reset or transition to standby mode. TE is cleared to 0 in SCR. TDRE is 1 when last bit of serial character is transmitted.

Framing error

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)

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 of O/$\overline{E}$ in SMR)

Receive data register full

0	[Clearing conditions] Reset or transition to standby mode. Read RDRF when RDRF = 1, then write 0 in RDRF.
1	[Setting condition] Serial data is received normally and transferred from RSR to RDR

Overrun error

0	[Clearing conditions] Reset or transition to standby mode. Read ORER when ORER = 1, then write 0 in ORER.
1	[Setting condition] Overrun error (reception of next serial data ends when RDRF = 1)

Transmit data register empty

0	[Clearing conditions] Read TDRE when TDRE = 1, then write 0 in TDRE.
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.

RDR—Receive Data Register **H'B5** **SCI0**

Bit	7	6	5	4	3	2	1	0
Initial value	0	0	0	0	0	0	0	0
Read/Write	R	R	R	R	R	R	R	R

Serial receive data

SCMR—Smart Card Mode Register **H'B6** **SCI0**

Bit	7	6	5	4	3	2	1	0
	—	—	—	—	SDIR	SINV	—	SMIF
Initial value	1	1	1	1	0	0	1	0
Read/Write	—	—	—	—	R/W	R/W	—	R/W

Smart card interface mode select

0	Smart card interface function is disabled
1	Smart card interface function is enabled

Smart card data invert

0	Unmodified TDR contents are transmitted Received data is stored unmodified in RDR
1	Inverted 1/0 logic levels of TDR contents are transmitted 1/0 logic levels of received data are inverted before storage in RDR

Smart card data transfer direction

0	TDR contents are transmitted LSB-first Received data is stored LSB-first in RDR
1	TDR contents are transmitted MSB-first Received data is stored MSB-first in RDR

SMR—Serial Mode Register **H'B8** **SCI1**

Bit	7	6	5	7	3	2	1	0
	C/$\overline{A}$	CHR	PE	O/$\overline{E}$	STOP	MP	CKS1	CKS0
Initial value	0	0	0	0	0	0	0	0
Read/Write	R/W	R/W	R/W	R/W	R/W	R/W	R/W	R/W

Note: Bit functions are the same as for SCI0.

BRR—Bit Rate Register **H'B9** **SCI1**

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

Note: Bit functions are the same as for SCI0.

SCR—Serial Control Register **H'BA** **SCI1**

Bit	7	6	5	4	3	2	1	0
	TIE	RIE	TE	RE	MPIE	TEIE	CKE1	CKE0
Initial value	0	0	0	0	0	0	0	0
Read/Write	R/W	R/W	R/W	R/W	R/W	R/W	R/W	R/W

Note: Bit functions are the same as for SCI0.

TDR—Transmit Data Register **H'BB** **SCI1**

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

Note: Bit functions are the same as for SCI0.

SSR—Serial Status Register **H'BC** **SCI1**

Bit	7	6	5	4	3	2	1	0
	TDRE	RDRF	ORER	FER	PER	TEND	MPB	MPBT
Initial value	1	0	0	0	0	1	0	0
Read/Write	R/(W)*	R/(W)*	R/(W)*	R/(W)*	R/(W)*	R	R	R/W

Notes: Bit functions are the same as for SCI0.
* Only 0 can be written to clear the flag.

RDR—Receive Data Register **H'BD** **SCI1**

Bit	7	6	5	4	3	2	1	0
Initial value	0	0	0	0	0	0	0	0
Read/Write	R	R	R	R	R	R	R	R

Note: Bit functions are the same as for SCI0.

P1DDR—Port 1 Data Direction Register **H'C0** **Port 1**

Bit		7	6	5	4	3	2	1	0
		$P1_7DDR$	$P1_6DDR$	$P1_5DDR$	$P1_4DDR$	$P1_3DDR$	$P1_2DDR$	$P1_1DDR$	$P1_0DDR$
Modes 1 and 3	Initial value	1	1	1	1	1	1	1	1
	Read/Write	—	—	—	—	—	—	—	—
Modes 5 to 7	Initial value	0	0	0	0	0	0	0	0
	Read/Write	W	W	W	W	W	W	W	W

Port 1 input/output select

0	Generic input pin
1	Generic output pin

P2DDR—Port 2 Data Direction Register **H'C1** **Port 2**

Bit		7	6	5	4	3	2	1	0
		$P2_7$DDR	$P2_6$DDR	$P2_5$DDR	$P2_4$DDR	$P2_3$DDR	$P2_2$DDR	$P2_1$DDR	$P2_0$DDR
Modes 1 and 3	Initial value	1	1	1	1	1	1	1	1
	Read/Write	—	—	—	—	—	—	—	—
Modes 5 to 7	Initial value	0	0	0	0	0	0	0	0
	Read/Write	W	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** **Port 1**

Bit	7	6	5	4	3	2	1	0
	$P1_7$	$P1_6$	$P1_5$	$P1_4$	$P1_3$	$P1_2$	$P1_1$	$P1_0$
Initial value	0	0	0	0	0	0	0	0
Read/Write	R/W	R/W	R/W	R/W	R/W	R/W	R/W	R/W

Data for port 1 pins

P2DR—Port 2 Data Register **H'C3** **Port 2**

Bit	7	6	5	4	3	2	1	0
	$P2_7$	$P2_6$	$P2_5$	$P2_4$	$P2_3$	$P2_2$	$P2_1$	$P2_0$
Initial value	0	0	0	0	0	0	0	0
Read/Write	R/W	R/W	R/W	R/W	R/W	R/W	R/W	R/W

Data for port 2 pins

P3DDR—Port 3 Data Direction Register **H'C4** **Port 3**

Bit	7	6	5	4	3	2	1	0
	$P3_7DDR$	$P3_6DDR$	$P3_5DDR$	$P3_4DDR$	$P3_3DDR$	$P3_2DDR$	$P3_1DDR$	$P3_0DDR$
Initial value	0	0	0	0	0	0	0	0
Read/Write	W	W	W	W	W	W	W	W

Port 3 input/output select

0	Generic input pin
1	Generic output pin

P3DR—Port 3 Data Register **H'C6** **Port 3**

Bit	7	6	5	4	3	2	1	0
	$P3_7$	$P3_6$	$P3_5$	$P3_4$	$P3_3$	$P3_2$	$P3_1$	$P3_0$
Initial value	0	0	0	0	0	0	0	0
Read/Write	R/W	R/W	R/W	R/W	R/W	R/W	R/W	R/W

Data for port 3 pins

P5DDR—Port 5 Data Direction Register **H'C8** **Port 5**

Bit		7	6	5	4	3	2	1	0
		—	—	—	—	$P5_3DDR$	$P5_2DDR$	$P5_1DDR$	$P5_0DDR$
Modes 1 and 3	Initial value	1	1	1	1	1	1	1	1
	Read/Write	—	—	—	—	—	—	—	—
Modes 5 to 7	Initial value	1	1	1	1	0	0	0	0
	Read/Write	—	—	—	—	W	W	W	W

Port 5 input/output select

0	Generic input
1	Generic output

P6DDR—Port 6 Data Direction Register **H'C9** **Port 6**

Bit	7	6	5	4	3	2	1	0
	—	—	$P6_5DDR$	$P6_4DDR$	$P6_3DDR$	—	—	$P6_0DDR$
Initial value	1	0	0	0	0	0	0	0
Read/Write	—	W	W	W	W	W	W	W

Port 6 input/output select

0	Generic input
1	Generic output

P5DR—Port 5 Data Register **H'CA** **Port 5**

Bit	7	6	5	4	3	2	1	0
	—	—	—	—	$P5_3$	$P5_2$	$P5_1$	$P5_0$
Initial value	1	1	1	1	0	0	0	0
Read/Write	—	—	—	—	R/W	R/W	R/W	R/W

Data for port 5 pins

P6DR—Port 6 Data Register **H'CB** **Port 6**

Bit	7	6	5	4	3	2	1	0
	—	—	$P6_5$	$P6_4$	$P6_3$	—	—	$P6_0$
Initial value	1	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 6 pins

P8DDR—Port 8 Data Direction Register **H'CD** **Port 8**

Bit	7	6	5	4	3	2	1	0
	—	—	—	—	—	—	$P8_1DDR$	$P8_0DDR$
Initial value	1	1	1	0	0	0	0	0
Read/Write	—	—	—	W	W	W	W	W

Port 8 input/output select

0	Generic input
1	Generic output

P7DR—Port 7 Data Register **H'CE** **Port 7**

Bit	7	6	5	4	3	2	1	0
	$P7_7$	$P7_6$	$P7_5$	$P7_4$	$P7_3$	$P7_2$	$P7_1$	$P7_0$
Initial value	—*	—*	—*	—*	—*	—*	—*	—*
Read/Write	R	R	R	R	R	R	R	R

Data for port 7 pins

Note: * Determined by pins $P7_7$ to $P7_0$.

P8DR—Port 8 Data Register **H'CF** **Port 8**

Bit	7	6	5	4	3	2	1	0
	—	—	—	—	—	—	$P8_1$	$P8_0$
Initial value	1	1	1	0	0	0	0	0
Read/Write	—	—	—	R/W	R/W	R/W	R/W	R/W

Data for port 8 pins

P9DDR—Port 9 Data Direction Register **H'D0** **Port 9**

Bit	7	6	5	4	3	2	1	0
	—	—	$P9_5DDR$	$P9_4DDR$	$P9_3DDR$	$P9_2DDR$	$P9_1DDR$	$P9_0DDR$
Initial value	1	1	0	0	0	0	0	0
Read/Write	—	—	W	W	W	W	W	W

Port 9 input/output select

0	Generic input
1	Generic output

PADDR—Port A Data Direction Register **H'D1** **Port A**

	Bit	7	6	5	4	3	2	1	0
		PA_7DDR	PA_6DDR	PA_5DDR	PA_4DDR	PA_3DDR	PA_2DDR	PA_1DDR	PA_0DDR
Mode 3	Initial value	1	0	0	0	0	0	0	0
	Read/Write	—	W	W	W	W	W	W	W
Modes 1 and 5 to 7	Initial value	0	0	0	0	0	0	0	0
	Read/Write	W	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** **Port 9**

Bit	7	6	5	4	3	2	1	0
	—	—	$P9_5$	$P9_4$	$P9_3$	$P9_2$	$P9_1$	$P9_0$
Initial value	1	1	0	0	0	0	0	0
Read/Write	—	—	R/W	R/W	R/W	R/W	R/W	R/W

Data for port 9 pins

PADR—Port A Data Register **H'D3** **Port A**

Bit	7	6	5	4	3	2	1	0
	PA_7	PA_6	PA_5	PA_4	PA_3	PA_2	PA_1	PA_0
Initial value	0	0	0	0	0	0	0	0
Read/Write	R/W	R/W	R/W	R/W	R/W	R/W	R/W	R/W

Data for port A pins

PBDDR—Port B Data Direction Register **H'D4** **Port B**

Bit	7	6	5	4	3	2	1	0
	PB_7DDR	—	PB_5DDR	PB_4DDR	PB_3DDR	PB_2DDR	PB_1DDR	PB_0DDR
Initial value	0	0	0	0	0	0	0	0
Read/Write	W	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** **Port B**

Bit	7	6	5	4	3	2	1	0
	PB_7	—	PB_5	PB_4	PB_3	PB_2	PB_1	PB_0
Initial value	0	0	0	0	0	0	0	0
Read/Write	R/W	R/W	R/W	R/W	R/W	R/W	R/W	R/W

Data for port B pins

P2PCR—Port 2 Input Pull-Up Control Register H'D8 Port 2

Bit	7	6	5	4	3	2	1	0
	$P2_7PCR$	$P2_6PCR$	$P2_5PCR$	$P2_4PCR$	$P2_3PCR$	$P2_2PCR$	$P2_1PCR$	$P2_0PCR$
Initial value	0	0	0	0	0	0	0	0
Read/Write	R/W	R/W	R/W	R/W	R/W	R/W	R/W	R/W

Port 2 input pull-up control 7 to 0

0	Input pull-up transistor is off
1	Input pull-up transistor is on

Note: Valid when the corresponding P2DDR bit is cleared to 0 (designating generic input).

P5PCR—Port 5 Input Pull-Up Control Register H'DB Port 5

Bit	7	6	5	4	3	2	1	0
	—	—	—	—	$P5_3PCR$	$P5_2PCR$	$P5_1PCR$	$P5_0PCR$
Initial value	1	1	1	1	0	0	0	0
Read/Write	—	—	—	—	R/W	R/W	R/W	R/W

Port 5 input pull-up control 3 to 0

0	Input pull-up transistor is off
1	Input pull-up transistor is on

Note: Valid when the corresponding P5DDR bit is cleared to 0 (designating generic input).

ADDRA H/L—A/D Data Register A H/L H'E0, H'E1 A/D

Bit	15	14	13	12	11	10	9	8	7	6	5	4	3	2	1	0
	AD9	AD8	AD7	AD6	AD5	AD4	AD3	AD2	AD1	AD0	—	—	—	—	—	—
Initial value	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0
Read/Write	R	R	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

ADDRB H/L—A/D Data Register B H/L **H'E2, H'E3** **A/D**

Bit	15	14	13	12	11	10	9	8	7	6	5	4	3	2	1	0
	AD9	AD8	AD7	AD6	AD5	AD4	AD3	AD2	AD1	AD0	—	—	—	—	—	—
Initial value	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0
Read/Write	R	R	R	R	R	R	R	R	R	R	R	R	R	R	R	R
	ADDRBH								ADDRBL							

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**

Bit	15	14	13	12	11	10	9	8	7	6	5	4	3	2	1	0
	AD9	AD8	AD7	AD6	AD5	AD4	AD3	AD2	AD1	AD0	—	—	—	—	—	—
Initial value	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0
Read/Write	R	R	R	R	R	R	R	R	R	R	R	R	R	R	R	R
	ADDRCH								ADDRCL							

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**

Bit	15	14	13	12	11	10	9	8	7	6	5	4	3	2	1	0
	AD9	AD8	AD7	AD6	AD5	AD4	AD3	AD2	AD1	AD0	—	—	—	—	—	—
Initial value	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0
Read/Write	R	R	R	R	R	R	R	R	R	R	R	R	R	R	R	R
	ADDRDH								ADDRDL							

A/D conversion data
10-bit data giving an
A/D conversion result

ADCR—A/D Control Register **H'E9** **A/D**

Bit	7	6	5	4	3	2	1	0
	TRGE	—	—	—	—	—	—	—
Initial value	0	1	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 ($\overline{\text{ADTRG}}$)

ADCSR—A/D Control/Status Register **H'E8** **A/D**

Bit	7	6	5	4	3	2	1	0
	ADF	ADIE	ADST	SCAN	CKS	CH2	CH1	CH0
Initial value	0	0	0	0	0	0	0	0
Read/Write	R/(W)*	R/W	R/W	R/W	R/W	R/W	R/W	R/W

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 Mode
0	0	0	AN_0	AN_0
		1	AN_1	AN_0, AN_1
	1	0	AN_2	AN_0 to AN_2
		1	AN_3	AN_0 to AN_3
1	0	0	AN_4	AN_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

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 when conversion ends Scan mode: A/D conversion starts and continues, cycling among the selected channels, until ADST is cleared to 0 by software, by a reset, or by a transition to standby mode

A/D interrupt enable

0	A/D end interrupt request is disabled
1	A/D end interrupt request is enabled

A/D end flag

0	[Clearing 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.

ASTCR—Access State Control Register **H'ED** **Bus controller**

Bit	7	6	5	4	3	2	1	0
	AST7	AST6	AST5	AST4	AST3	AST2	AST1	AST0
Initial value	1	1	1	1	1	1	1	1
Read/Write	R/W	R/W	R/W	R/W	R/W	R/W	R/W	R/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

WCR—Wait Control Register **H'EE** **Bus controller**

Bit	7	6	5	4	3	2	1	0
	—	—	—	—	WMS1	WMS0	WC1	WC0
Initial value	1	1	1	1	0	0	1	1
Read/Write	—	—	—	—	R/W	R/W	R/W	R/W

Wait mode select 1 and 0

Bit 3 WMS1	Bit 2 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 WC1	Bit 0 WC0	Number of Wait States
0	0	No wait states inserted by wait-state controller
	1	1 state inserted
1	0	2 states inserted
	1	3 states inserted

WCER—Wait Controller Enable Register **H'EF** **Bus controller**

Bit	7	6	5	4	3	2	1	0
	WCE7	WCE6	WCE5	WCE4	WCE3	WCE2	WCE1	WCE0
Initial value	1	1	1	1	1	1	1	1
Read/Write	R/W	R/W	R/W	R/W	R/W	R/W	R/W	R/W

Wait state controller enable 7 to 0

0	Wait-state control is disabled (pin wait mode 0)
1	Wait-state control is enabled

MDCR—Mode Control Register **H'F1** **System control**

Bit	7	6	5	4	3	2	1	0
	—	—	—	—	—	MDS2	MDS1	MDS0
Initial value	1	1	0	0	0	—*	—*	—*
Read/Write	—	—	—	—	—	R	R	R

Mode select 2 to 0

Bit 2	Bit 1	Bit 0	Operating mode
MD_2	MD_1	MD_0	
0	0	0	—
		1	Mode 1
	1	0	—
		1	Mode 3
1	0	0	—
		1	Mode 5
	1	0	Mode 6
		1	Mode 7

Note: * Determined by the state of the mode pins (MD_2 to MD_0).

SYSCR—System Control Register **H'F2** **System control**

Bit	7	6	5	4	3	2	1	0
	SSBY	STS2	STS1	STS0	UE	NMIEG	—	RAME
Initial value	0	0	0	0	1	0	1	1
Read/Write	R/W	R/W	R/W	R/W	R/W	R/W	—	R/W

RAM enable

0	On-chip RAM is disabled
1	On-chip RAM is enabled

NMI edge select

0	An interrupt is requested at the falling edge of NMI
1	An interrupt is requested at the rising edge of NMI

User bit enable

0	CCR bit 6 (UI) is used as an interrupt mask bit
1	CCR bit 6 (UI) is used as a user bit

Standby timer select 2 to 0

Bit 6	Bit 5	Bit 4	Standby Timer
STS2	STS1	STS0	
0	0	0	Waiting time = 8192 states
		1	Waiting time = 16384 states
	1	0	Waiting time = 32768 states
		1	Waiting time = 65536 states
1	0	—	Waiting time = 131072 states
	1	—	Illegal setting

Software standby

0	SLEEP instruction causes transition to sleep mode
1	SLEEP instruction causes transition to software standby mode

ADRCR—Address Control Register **H'F3** **Bus controller**

	Bit	7	6	5	4	3	2	1	0
		$A_{23}E$	$A_{22}E$	$A_{21}E$	—	—	—	—	—
Modes 1 and 5 to 7	Initial value	1	1	1	1	1	1	1	0
	Read/Write	—	—	—	—	—	—	—	R/W
Mode 3	Initial value	1	1	1	1	1	1	1	0
	Read/Write	R/W	R/W	R/W	—	—	—	—	R/W

Address 23 to 21 enable

0	Address output
1	I/O pins other than the above

ISCR—IRQ Sense Control Register **H'F4** **Interrupt controller**

Bit	7	6	5	4	3	2	1	0
	—	—	IRQ5SC	IRQ4SC	—	—	IRQ1SC	IRQ0SC
Initial value	0	0	0	0	0	0	0	0
Read/Write	R/W	R/W	R/W	R/W	R/W	R/W	R/W	R/W

IRQ_5, IRQ_4, IRQ_1 and IRQ_0 sense control

0	Interrupts are requested when $\overline{IRQ}_5$, $\overline{IRQ}_4$, $\overline{IRQ}_1$, and $\overline{IRQ}_0$ inputs are low
1	Interrupts are requested by falling-edge input at $\overline{IRQ}_5$, $\overline{IRQ}_4$, $\overline{IRQ}_1$ and $\overline{IRQ}_0$

IER—IRQ Enable Register **H'F5** **Interrupt controller**

Bit	7	6	5	4	3	2	1	0
	—	—	IRQ5E	IRQ4E	—	—	IRQ1E	IRQ0E
Initial value	0	0	0	0	0	0	0	0
Read/Write	R/W	R/W	R/W	R/W	R/W	R/W	R/W	R/W

IRQ_5, IRQ_4, IRQ_1, IRQ_0 enable

0	IRQ_5, IRQ_4, IRQ_1 and IRQ_0 interrupts are disabled
1	IRQ_5, IRQ_4, IRQ_1 and IRQ_0 interrupts are enabled

ISR—IRQ Status Register **H'F6** **Interrupt controller**

Bit	7	6	5	4	3	2	1	0
	—	—	IRQ5F	IRQ4F	—	—	IRQ1F	IRQ0F
Initial value	0	0	0	0	0	0	0	0
Read/Write	—	—	R/(W)*	R/(W)*	—	—	R/(W)*	R/(W)*

IRQ_5, IRQ_4, IRQ_1 and IRQ_0 flags

Bits 5, 4, 1 and 0 IRQ5F IRQ4F IRQ1F IRQ0F	Setting and Clearing Conditions
0	[Clearing conditions] • Read IRQnF when IRQnF = 1, then write 0 in IRQnF. • IRQnSC = 0, $\overline{IRQn}$ input is high, and interrupt exception handling is carried out. • IRQnSC = 1 and IRQn interrupt exception handling is carried out.
1	[Setting conditions] • IRQnSC = 0 and $\overline{IRQn}$ input is low. • IRQnSC = 1 and $\overline{IRQn}$ input changes from high to low.

Note: n = 5, 4, 1 and 0

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

IPRA—Interrupt Priority Register A **H'F8** **Interrupt controller**

Bit	7	6	5	4	3	2	1	0
	IPRA7	IPRA6	—	IPRA4	IPRA3	IPRA2	IPRA1	IPRA0
Initial value	0	0	0	0	0	0	0	0
Read/Write	R/W	R/W	R/W	R/W	R/W	R/W	R/W	R/W

Priority level A7, A6, A4 to A0

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 —	Bit 4 IPRA4	Bit 3 IPRA3	Bit 2 IPRA2	Bit 1 IPRA1	Bit 0 IPRA0
Interrupt source	IRQ_0	IRQ_1	—	IRQ_4, IRQ_5	WDT	ITU channel 0	ITU channel 1	ITU channel 2

IPRB—Interrupt Priority Register B **H'F9** **Interrupt controller**

Bit	7	6	5	4	3	2	1	0
	IPRB7	IPRB6	—	—	IPRB3	IPRB2	IPRB1	—
Initial value	0	0	0	0	0	0	0	0
Read/Write	R/W	R/W	R/W	R/W	R/W	R/W	R/W	R/W

Priority level B7, B6, B3 to B1

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 —	Bit 4 —	Bit 3 IPRB3	Bit 2 IPRB2	Bit 1 IPRB1	Bit 0 —
Interrupt source	ITU channel 3	ITU channel 4	—	—	SCI channel 0	SCI channel 1	A/D converter	—

Appendix C I/O Block Diagrams

C.1 Port 1 Block Diagram

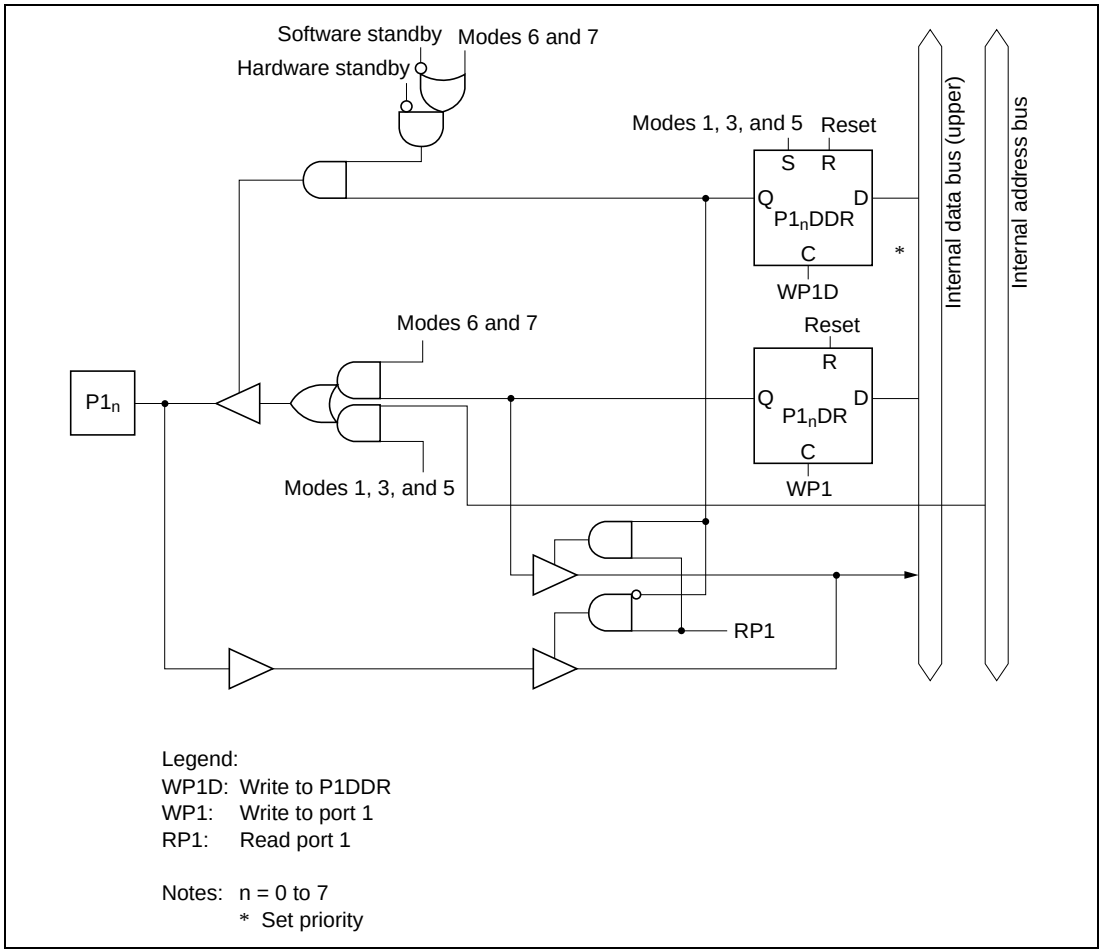


Figure C.1 Port 1 Block Diagram

C.2 Port 2 Block Diagram

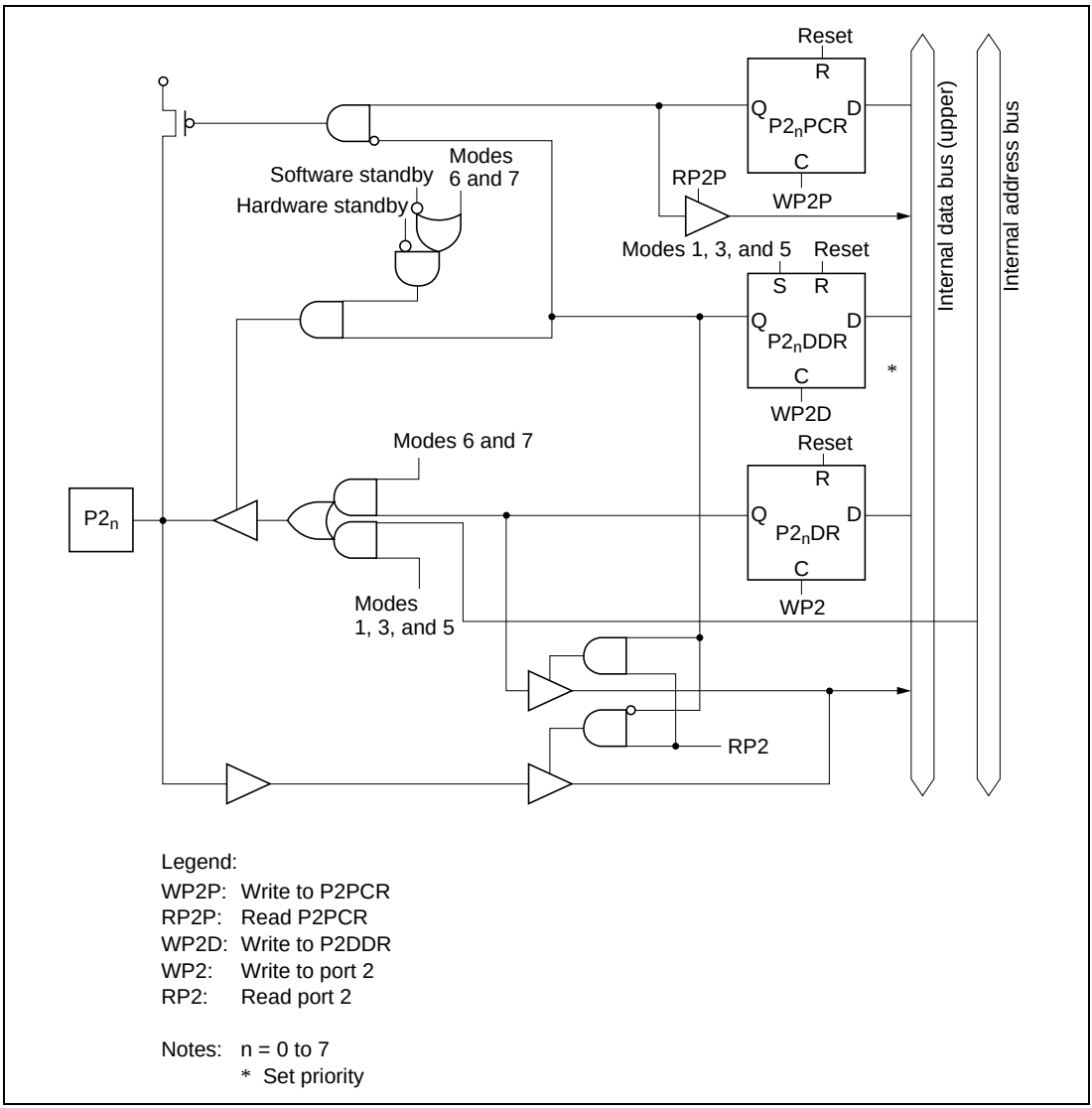


Figure C.2 Port 2 Block Diagram

C.3 Port 3 Block Diagram

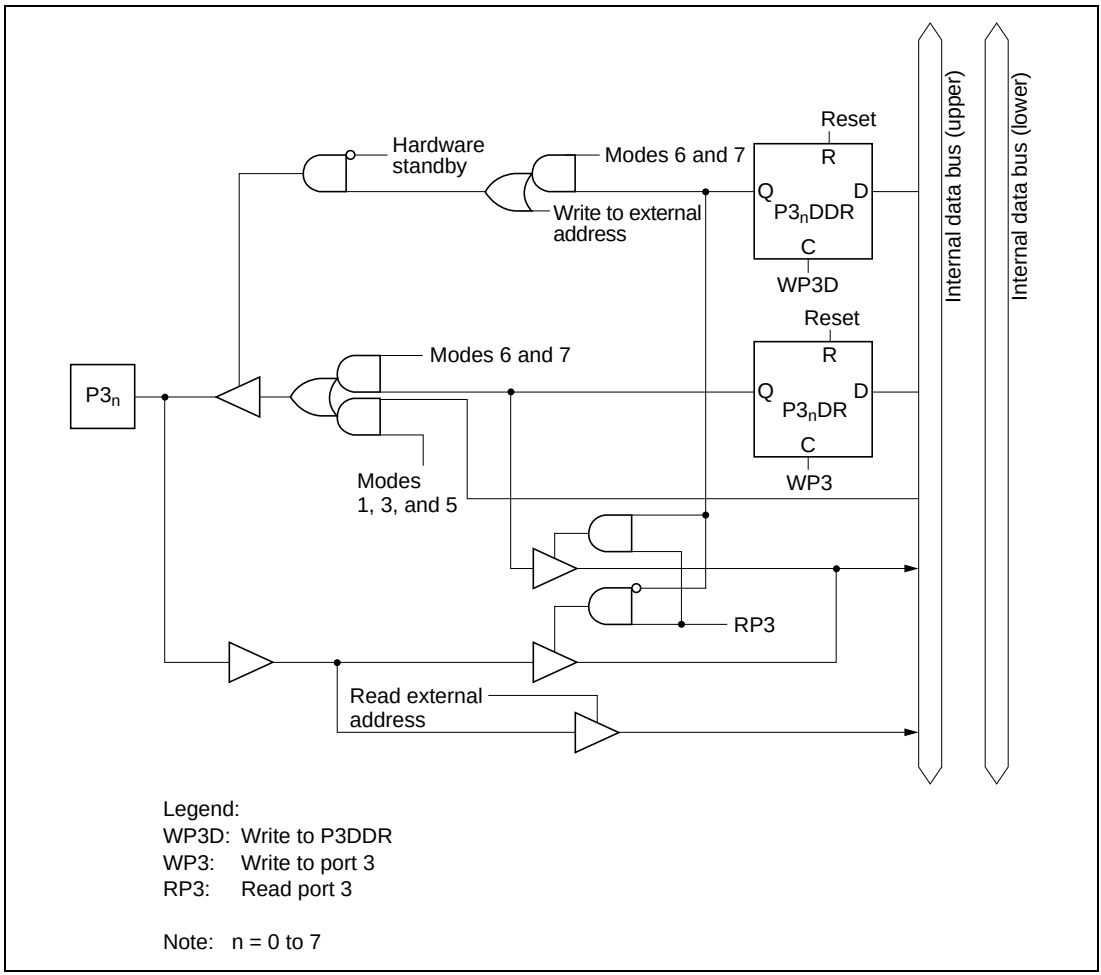


Figure C.3 Port 3 Block Diagram

C.4 Port 5 Block Diagram

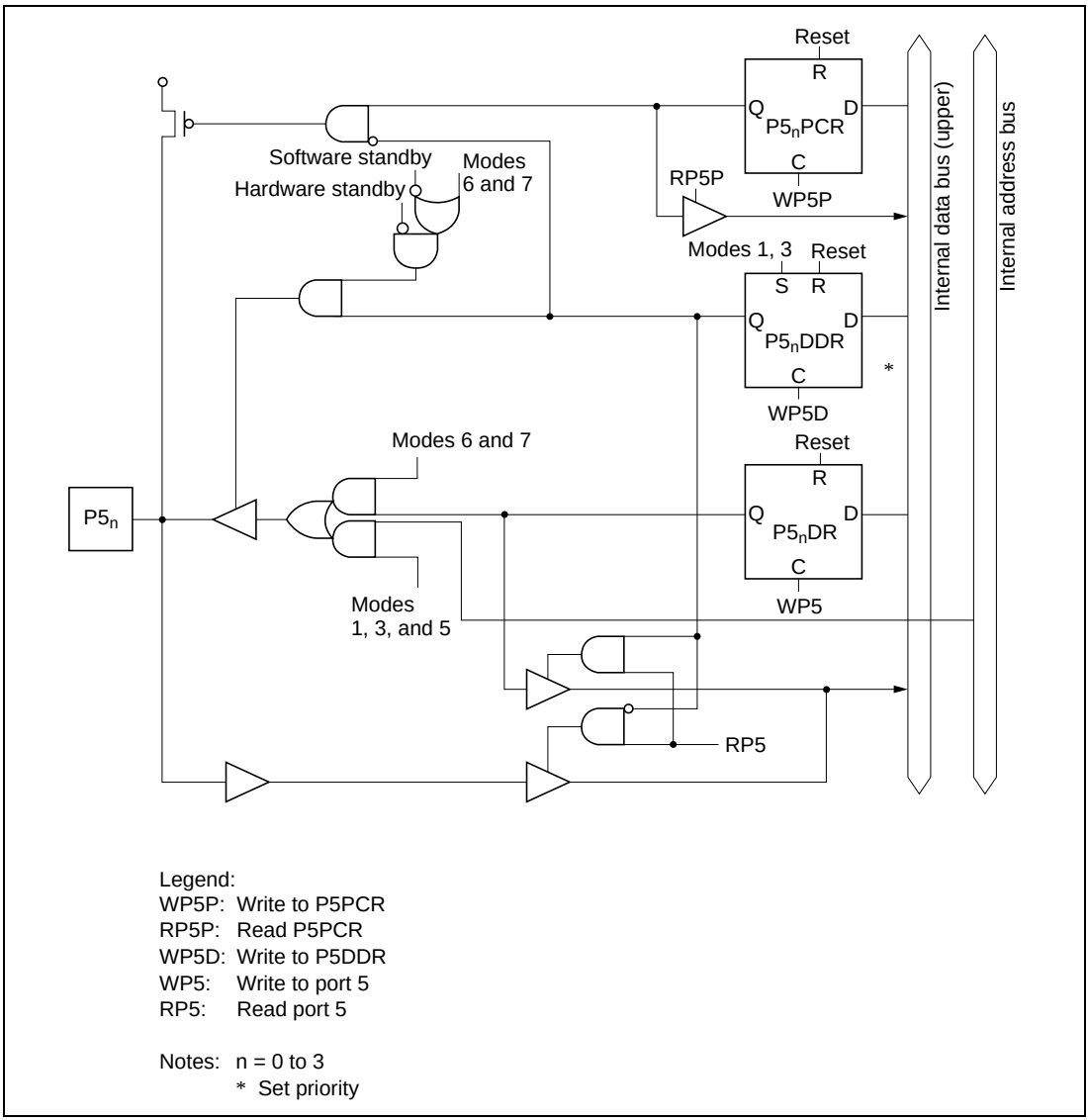


Figure C.4 Port 5 Block Diagram

C.5 Port 6 Block Diagram

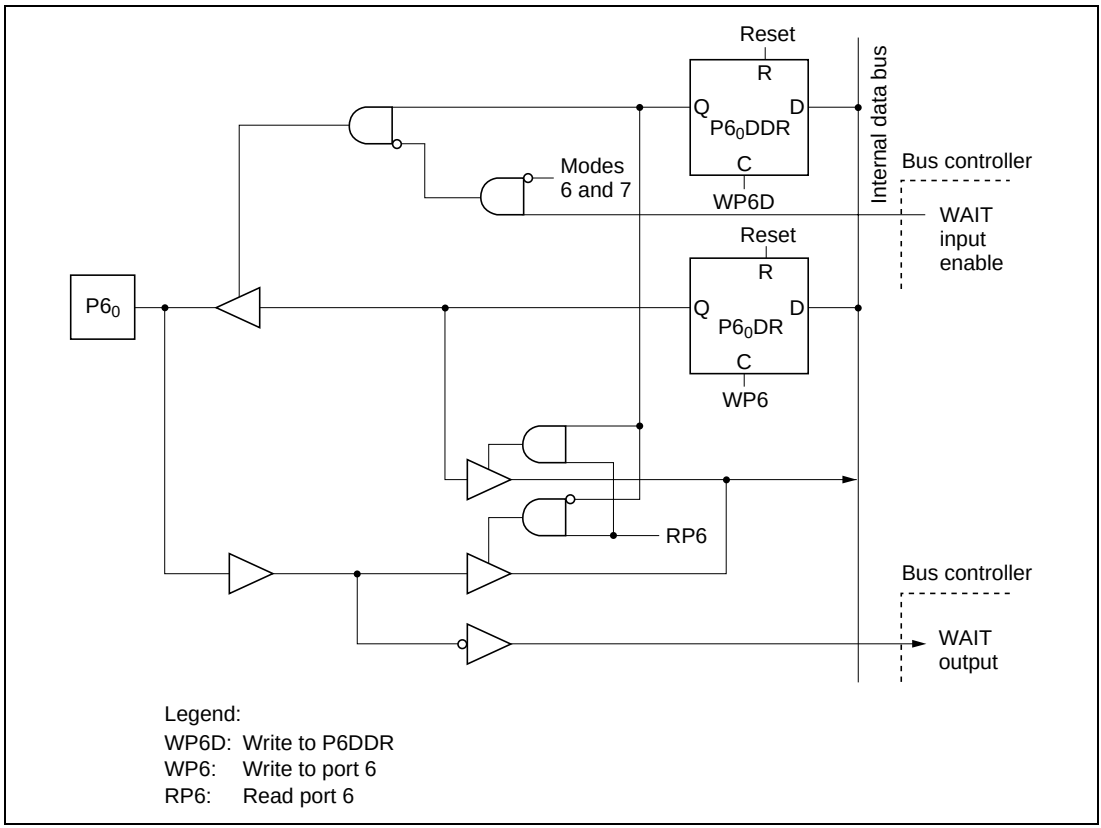


Figure C.5 (a) Port 6 Block Diagram (Pin $P6_0$)

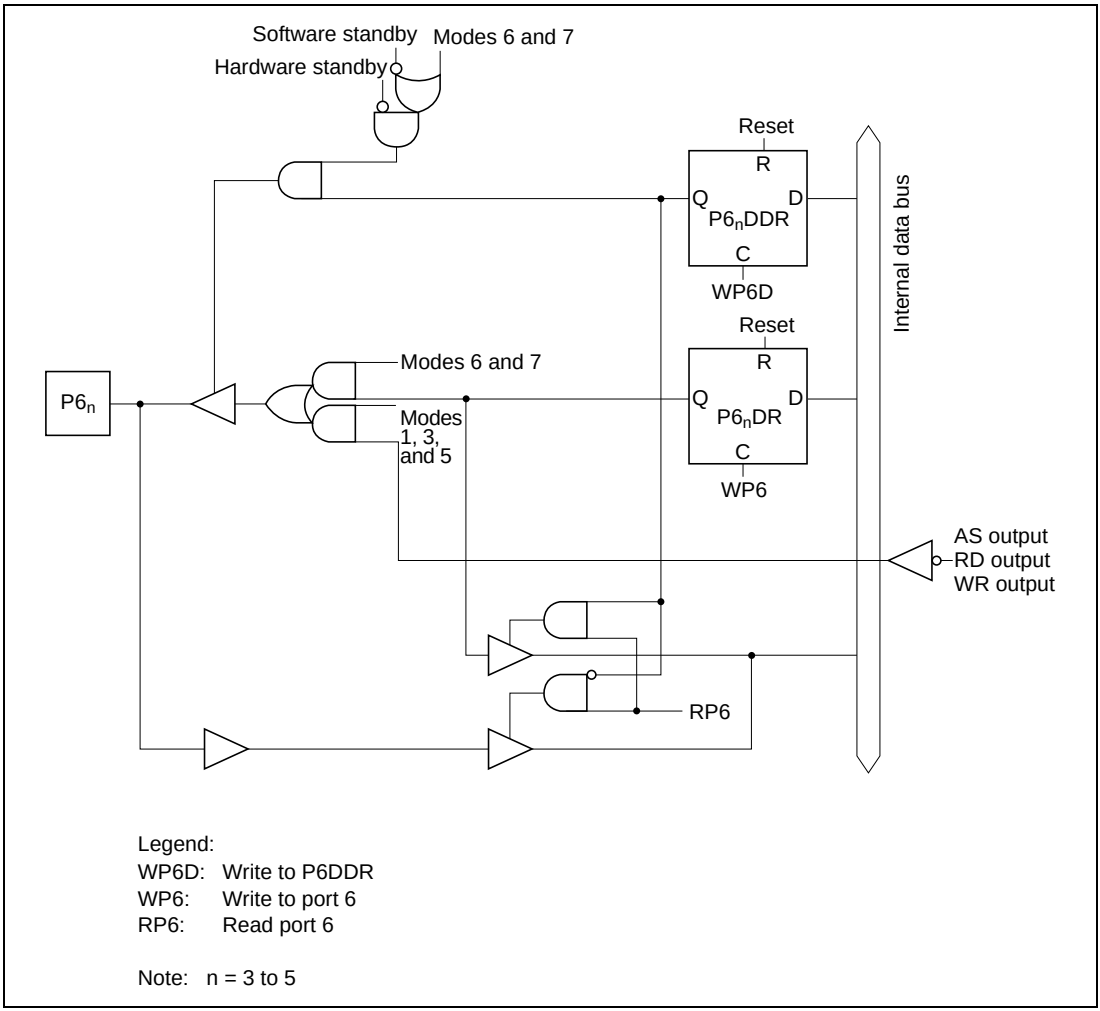


Figure C.5 (b) Port 6 Block Diagram (Pins $P6_3$ to $P6_5$)

C.6 Port 7 Block Diagram

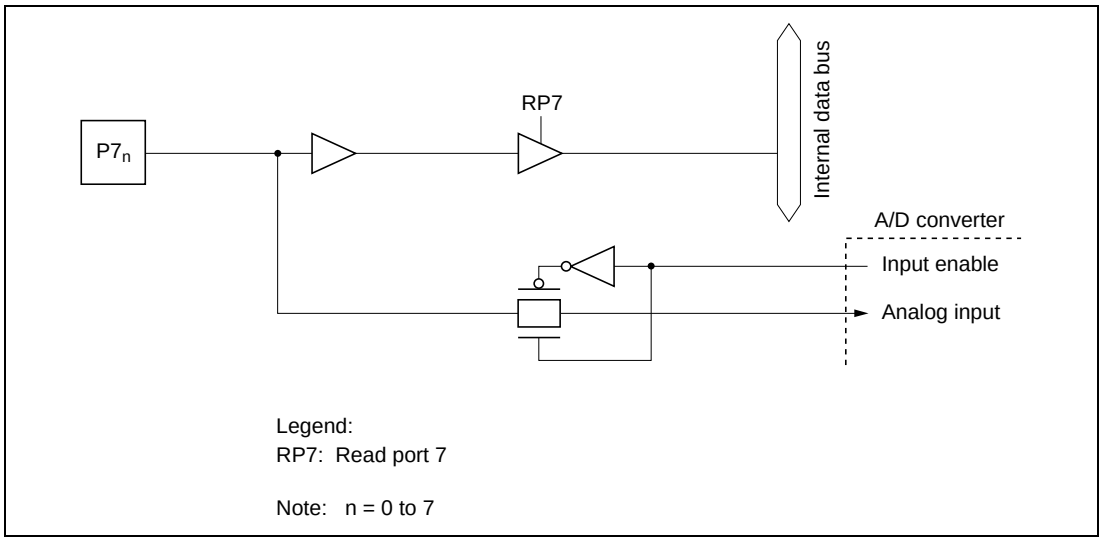


Figure C.6 Port 7 Block Diagram

C.7 Port 8 Block Diagram

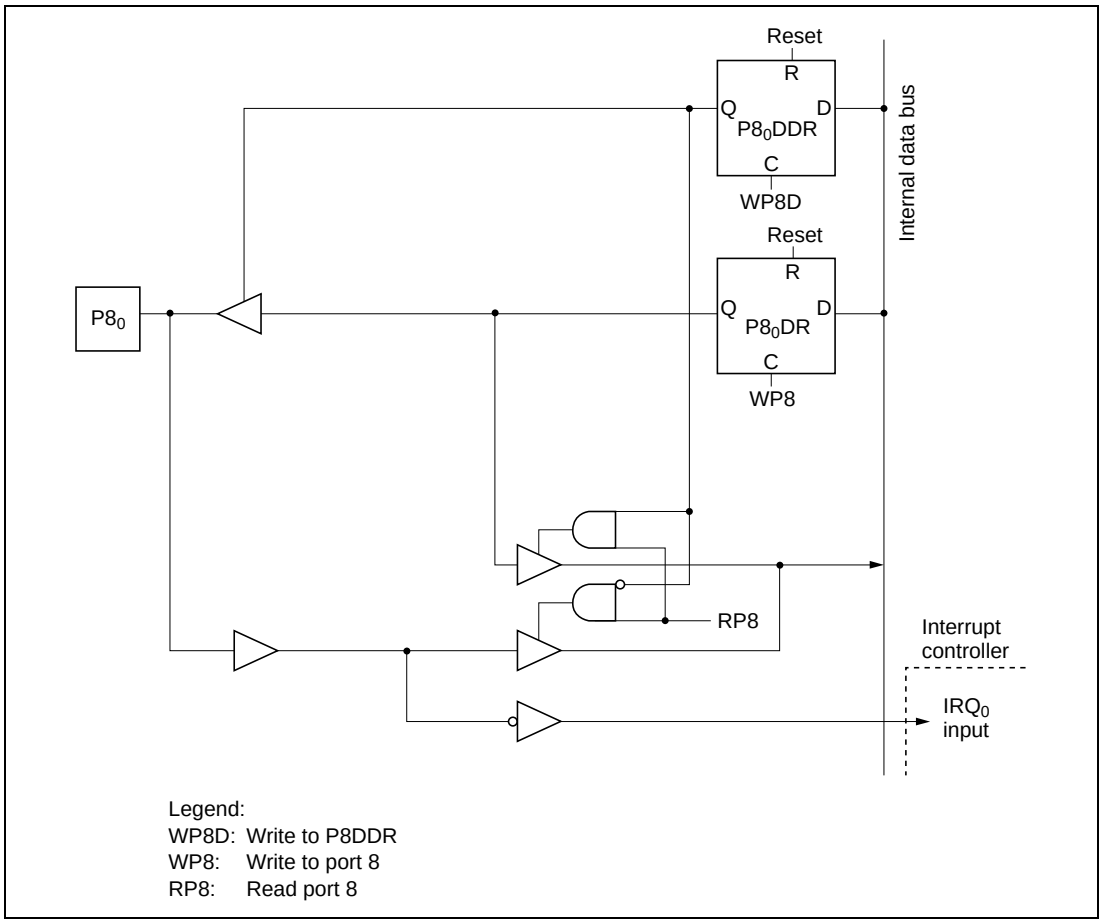


Figure C.7(a) Port 8 Block Diagram (Pin P8$_0$)

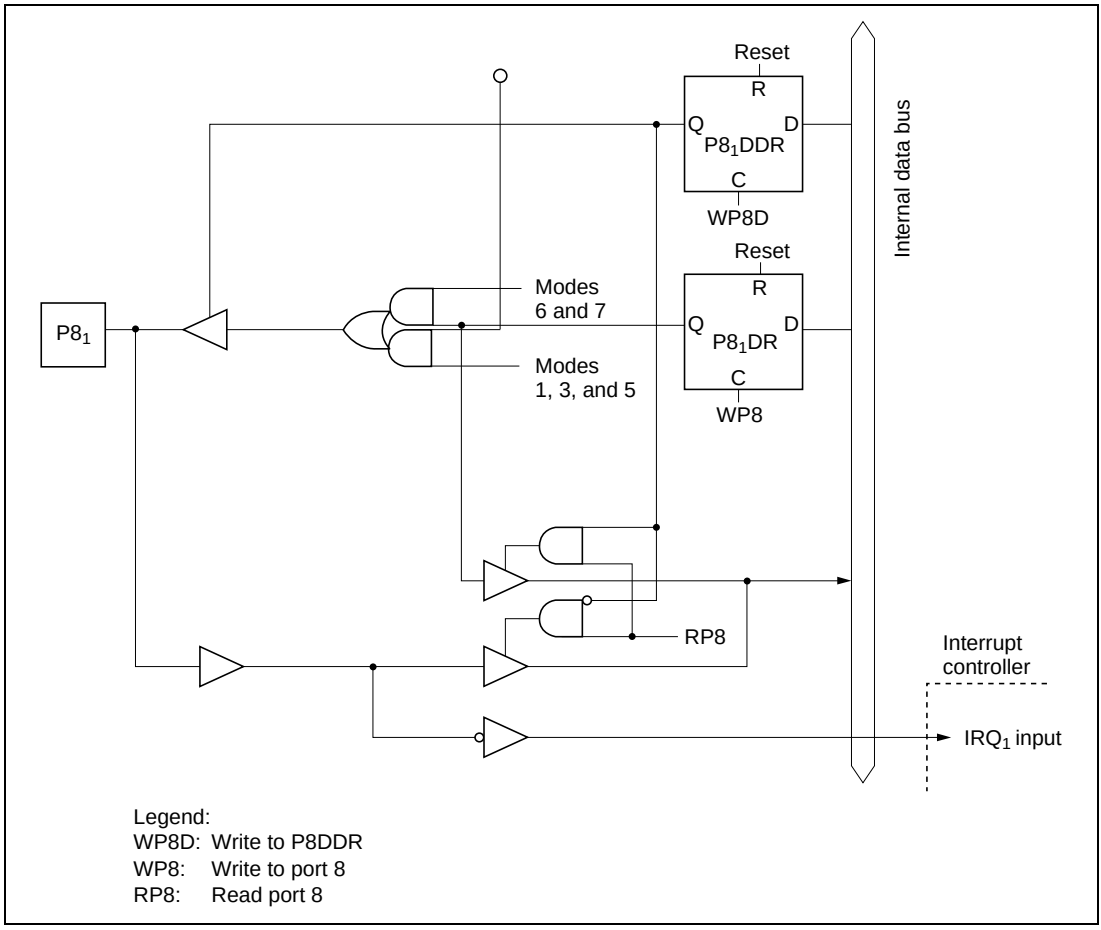


Figure C.7 (b) Port 8 Block Diagram (Pin P8$_1$)

C.8 Port 9 Block Diagram

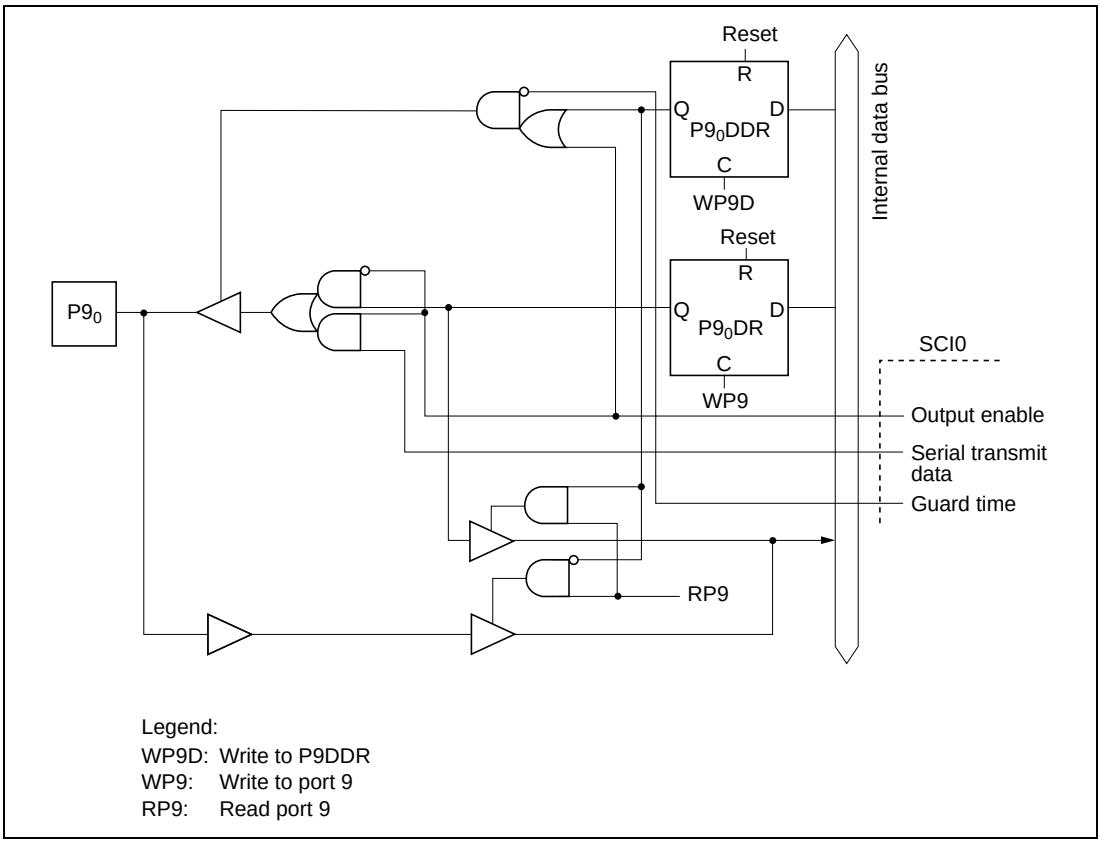


Figure C.8 (a) Port 9 Block Diagram (Pin P9$_0$)

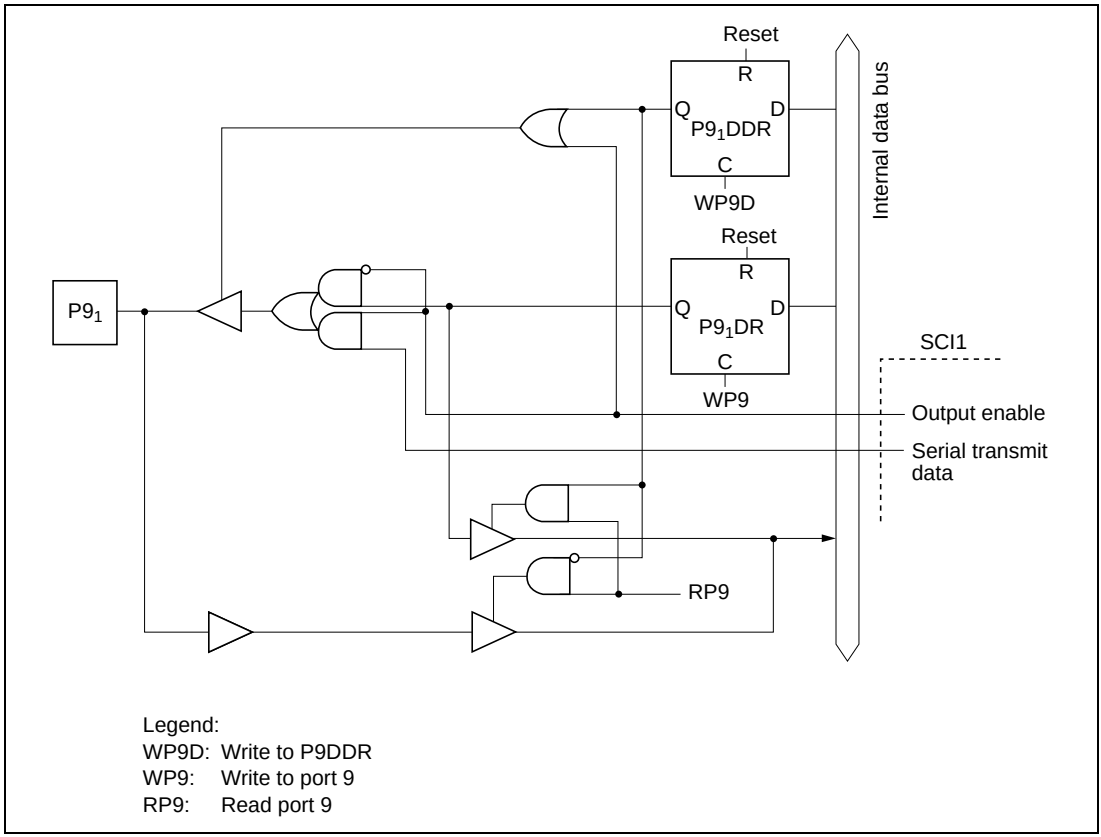


Figure C.8 (b) Port 9 Block Diagram (Pin P9$_1$)

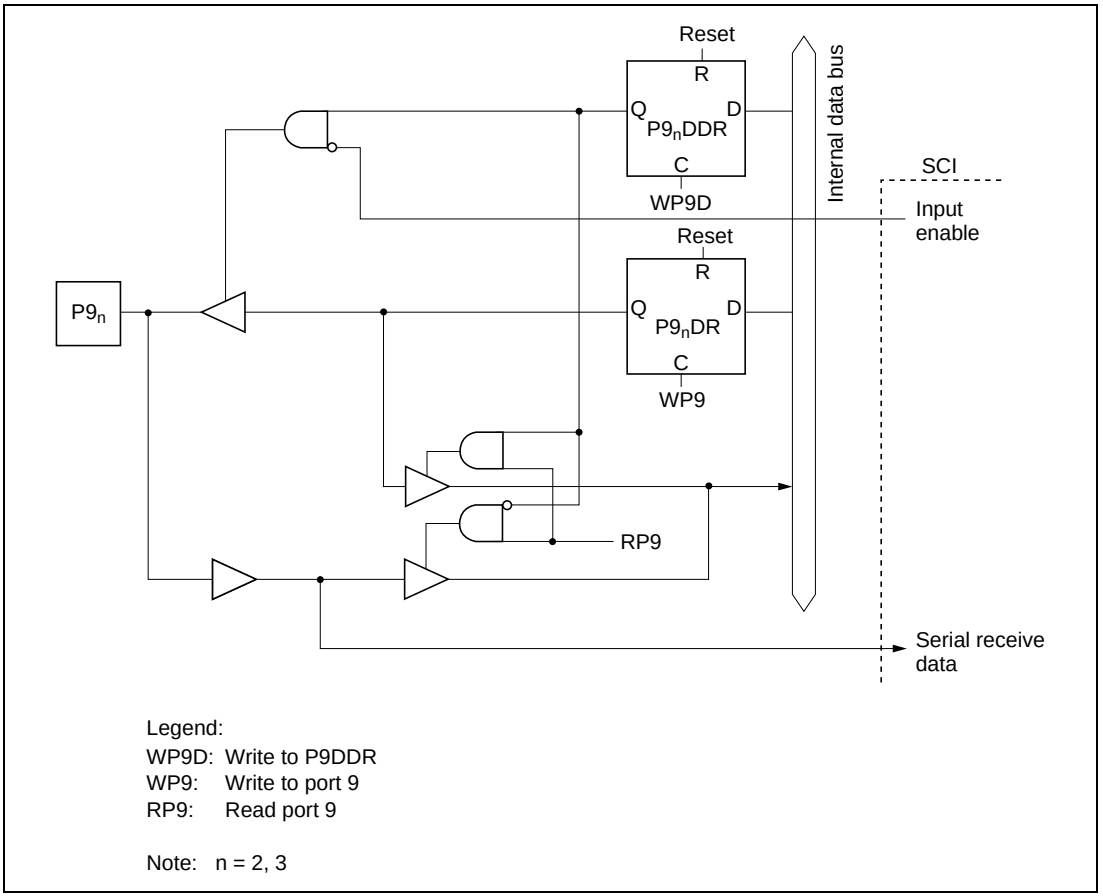


Figure C.8 (c) Port 9 Block Diagram (Pin $P9_2$, $P9_3$)

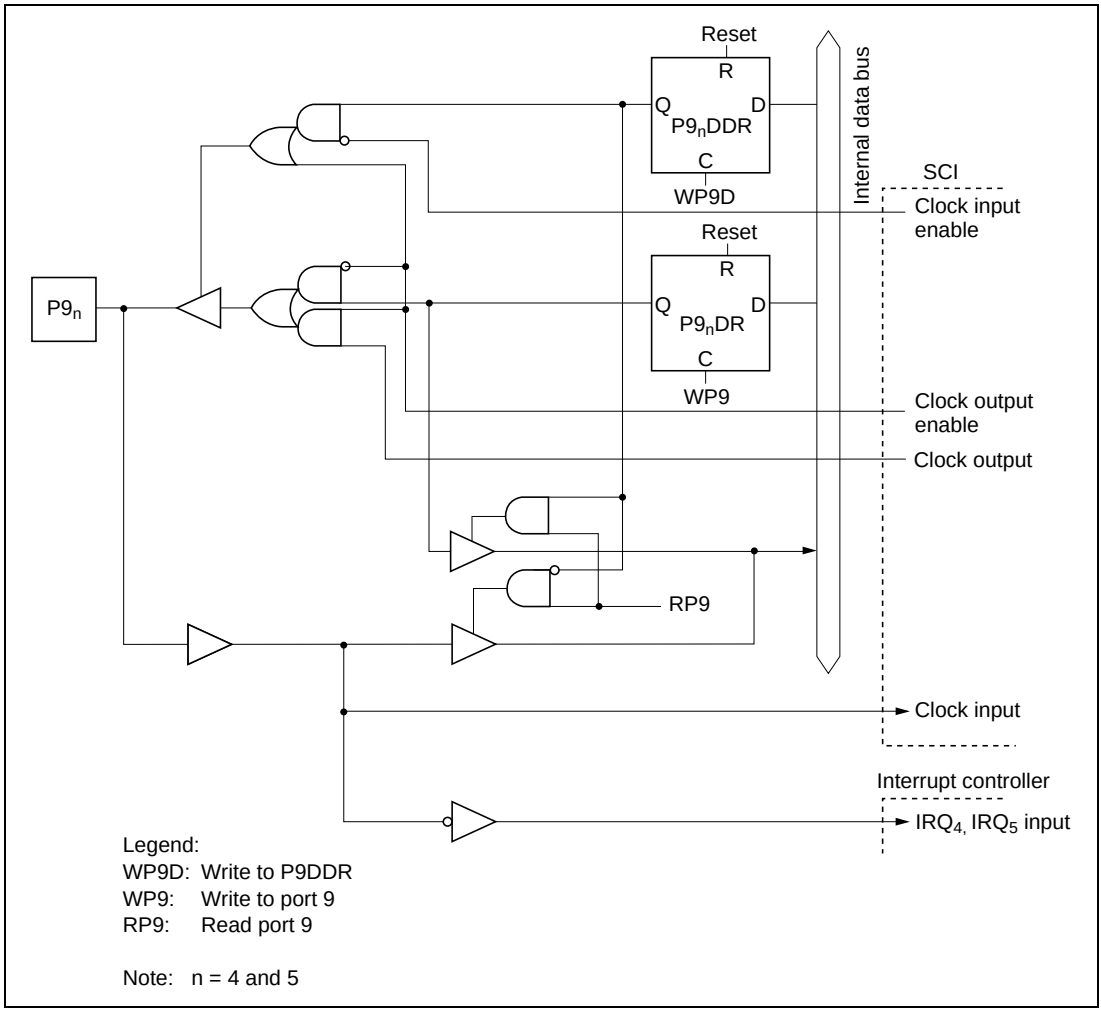


Figure C.8 (d) Port 9 Block Diagram (Pin $P9_4$, $P9_5$)

C.9 Port A Block Diagram

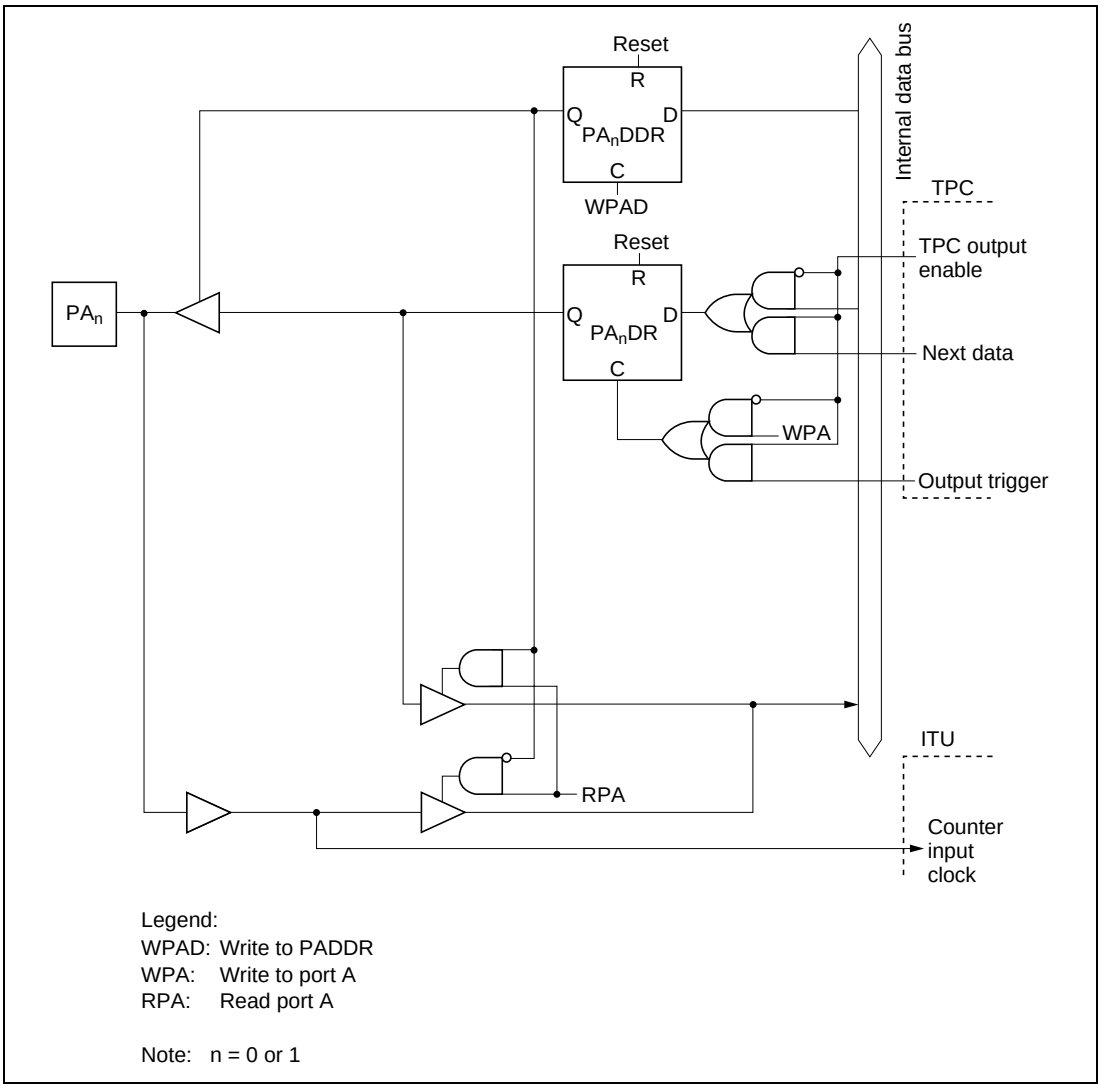


Figure C.9 (a) Port A Block Diagram (Pins PA_0, PA_1)

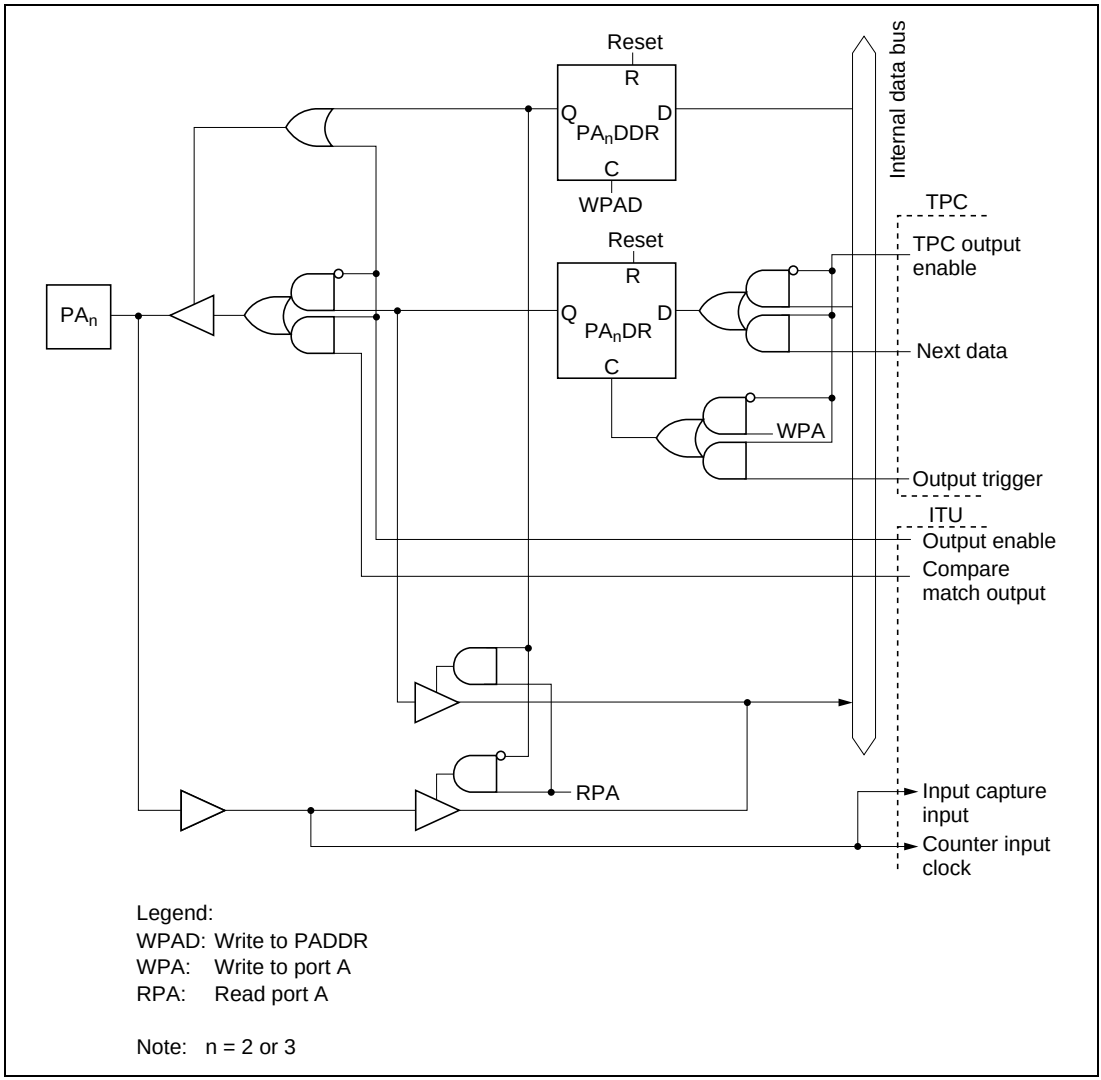
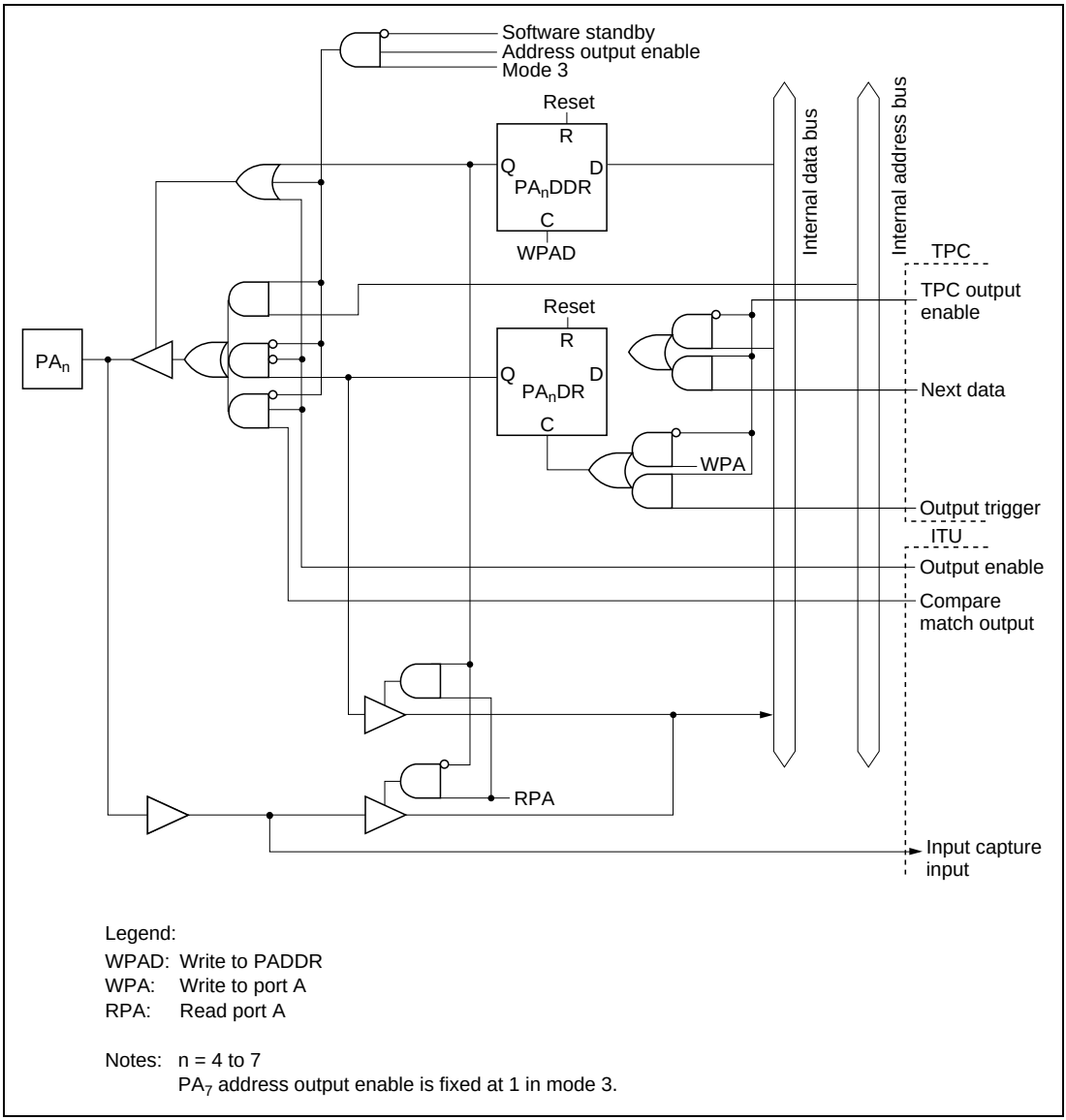


Figure C.9 (b) Port A Block Diagram (Pins PA_2, PA_3)



Legend:
WPAD: Write to PADDR
WPA: Write to port A
RPA: Read port A

Notes: n = 4 to 7
PA_7 address output enable is fixed at 1 in mode 3.

Figure C.9 (c) Port A Block Diagram (Pins PA_4 to PA_7)

C.10 Port B Block Diagram

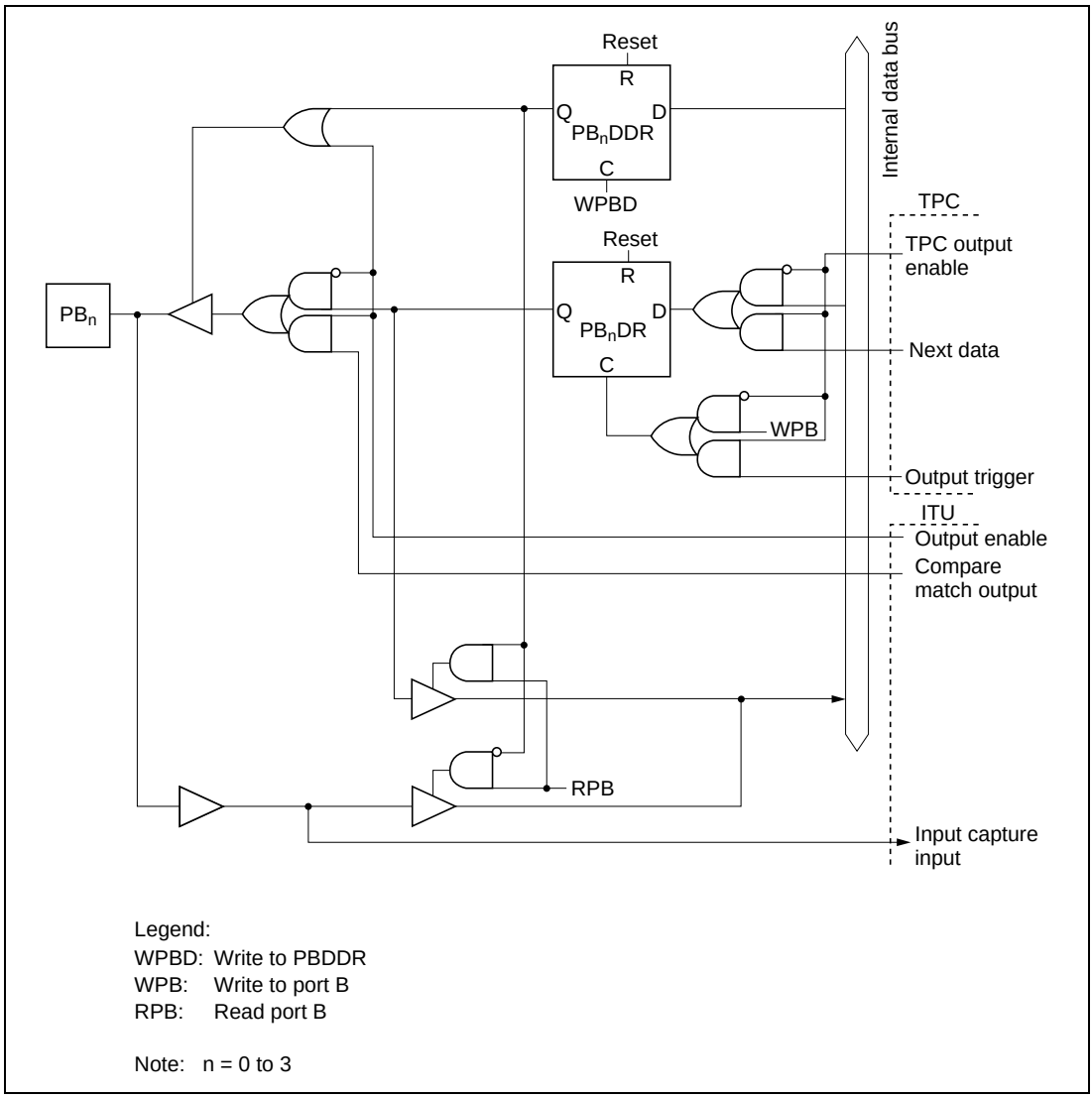


Figure C.10 (a) Port B Block Diagram (Pins PB_0 to PB_3)

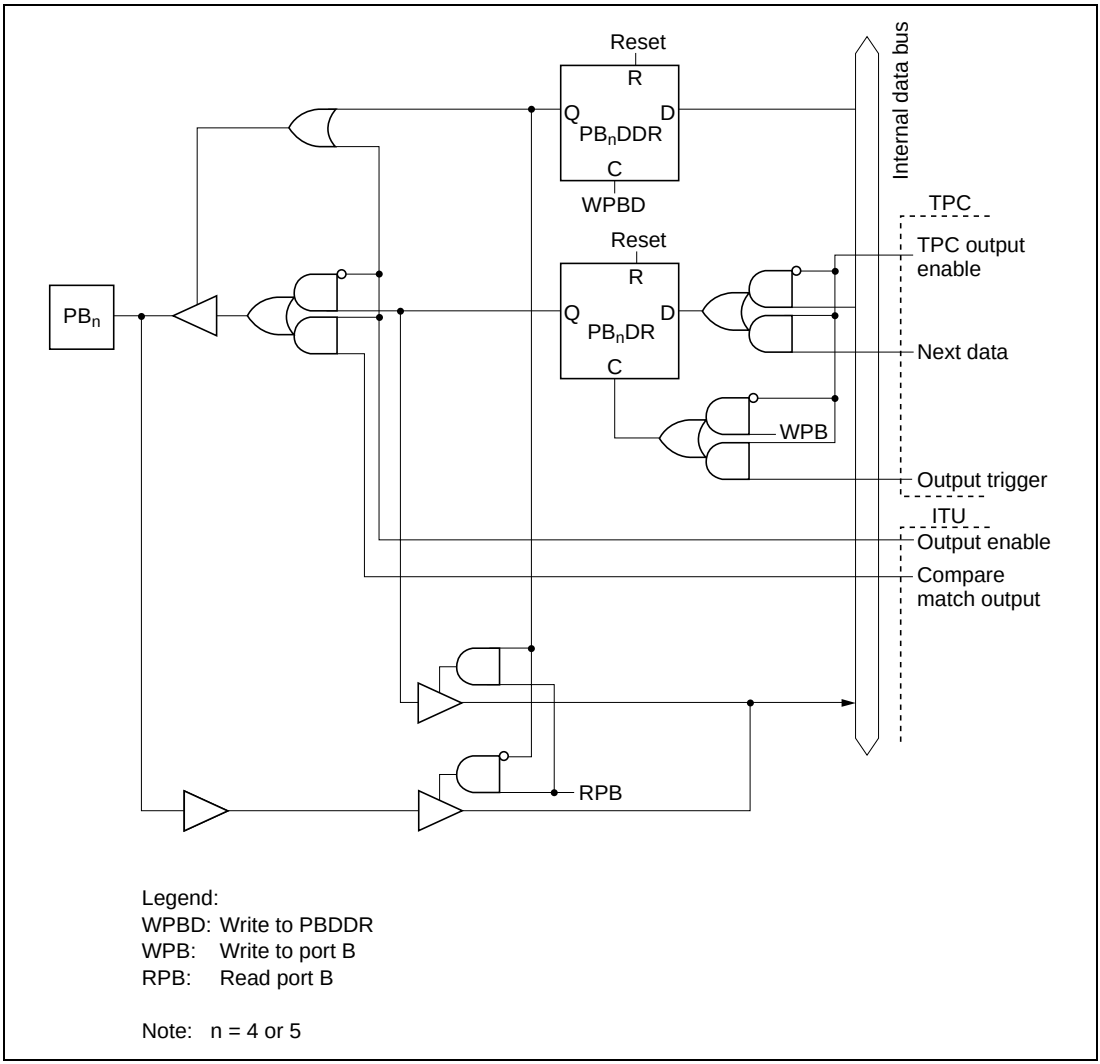


Figure C.10 (b) Port B Block Diagram (Pins PB_4, PB_5)

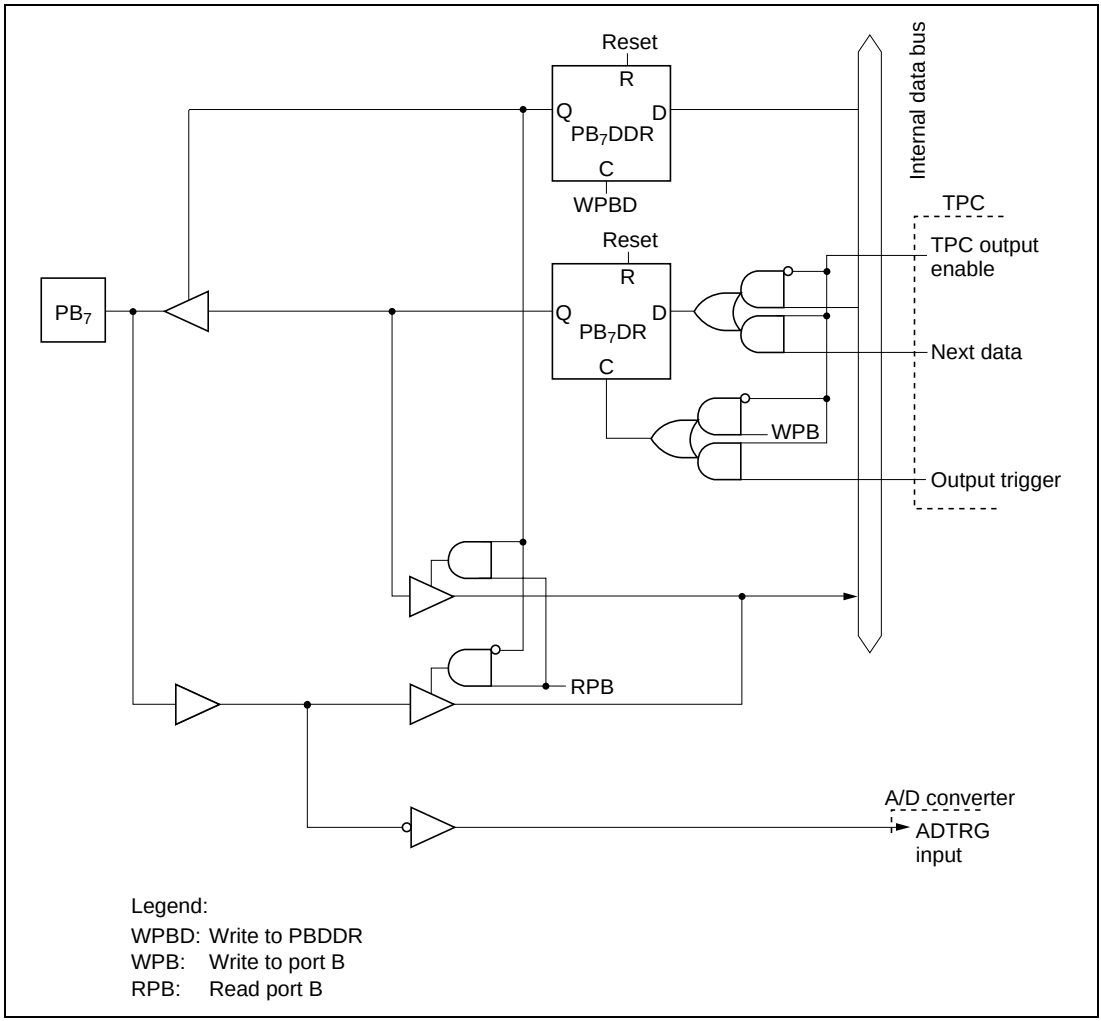


Figure C.10 (c) Port B Block Diagram (Pin PB_7)

Appendix D Pin States

D.1 Port States in Each Mode

Table D.1 Port States

Pin Name	Mode	Reset State	Hardware Standby Mode	Software Standby Mode	Program Execution State Sleep Mode
ϕ	—	Clock output	T	H	Clock output
$\overline{\text{RESO}}$*[1]	—	T*[2]	T	T	$\overline{\text{RESO}}$
$P1_7$ to $P1_0$	1, 3	L	T	T	A_7 to A_0
	5	T	T	keep	Input port (DDR = 0)
				T	A_7 to A_0 (DDR = 1)
	6, 7	T	T	keep	I/O port
$P2_7$ to $P2_0$	1, 3	L	T	T	A_{15} to A_8
	5	T	T	keep	Input port (DDR = 0)
				T	A_{15} to A_8 (DDR = 1)
	6, 7	T	T	keep	I/O port
$P3_7$ to $P3_0$	1, 3, 5	T	T	T	D_7 to D_0
	6, 7	T	T	keep	I/O port
$P5_3$ to $P5_0$	1, 3	L	T	T	A_{19} to A_{16}
	5	T	T	keep	Input port (DDR = 0)
				T	A_{19} to A_{16} (DDR = 1)
	6, 7	T	T	keep	I/O port
$P6_0$	1, 3, 5	T	T	keep	I/O port, $\overline{\text{WAIT}}$
	6, 7	T	T	keep	I/O port
$P6_5$ to $P6_3$	1, 3, 5	H	T	T	$\overline{\text{WR}}$, $\overline{\text{RD}}$, $\overline{\text{AS}}$
	6, 7	T	T	keep	I/O port
$P7_7$ to $P7_0$	1, 3, 5 to 7	T	T	T	Input port
$P8_0$	1, 3, 5	T	T	keep	I/O port
	6, 7	T	T	keep	I/O port

Pin Name	Mode	Reset State	Hardware Standby Mode	Software Standby Mode	Program Execution State Sleep Mode
$P8_1$	1, 3, 5	T	T	T [DDR = 0]	Input port [DDR = 0]
				H [DDR = 1]	H [DDR = 1]
	6, 7	T	T	keep	I/O port
$P9_5$ to $P9_0$	1, 3, 5 to 7	T	T	keep	I/O port
PA_3 to PA_0	1, 3, 5 to 7	T	T	keep	I/O port
PA_6 to PA_4	3	T	T	[ADRCR = 0] T	(ADRCR = 0) A_{21} to A_{23}
				[ADRCR = 1] keep	(ADRCR = 1) I/O port
	1, 5, 6, 7	T	T	keep	I/O port
PA_7	3	L	T	T	A_{20}
	1, 5, 6, 7	T	T	keep	I/O port
PB_7, PB_5 to PB_0	1, 3, 5 to 7	T	T	keep	I/O port

Legend:

H: High

L: Low

T: High-impedance state

keep: Input pins are in the high-impedance state; output pins maintain their previous state.

DDR: Data direction register bit

ADRCR: Address control register

Notes: 1 Mask ROM version. Dedicated FWE input pin for the F-ZTAT version.

2 Low output only when WDT overflows causes a reset.

D.2 Pin States at Reset

Reset in T1 State

Figure D.1 is a timing diagram for the case in which $\overline{RES}$ goes low during the T1 state of an external memory access cycle. As soon as $\overline{RES}$ goes low, all ports are initialized to the input state. $\overline{AS}$, $\overline{RD}$, and $\overline{WR}$ go high, and the data bus goes to the high-impedance state. The address bus is initialized to the low output level 0.5 state after the low level of $\overline{RES}$ is sampled. Sampling of $\overline{RES}$ takes place at the fall of the system clock (ϕ).

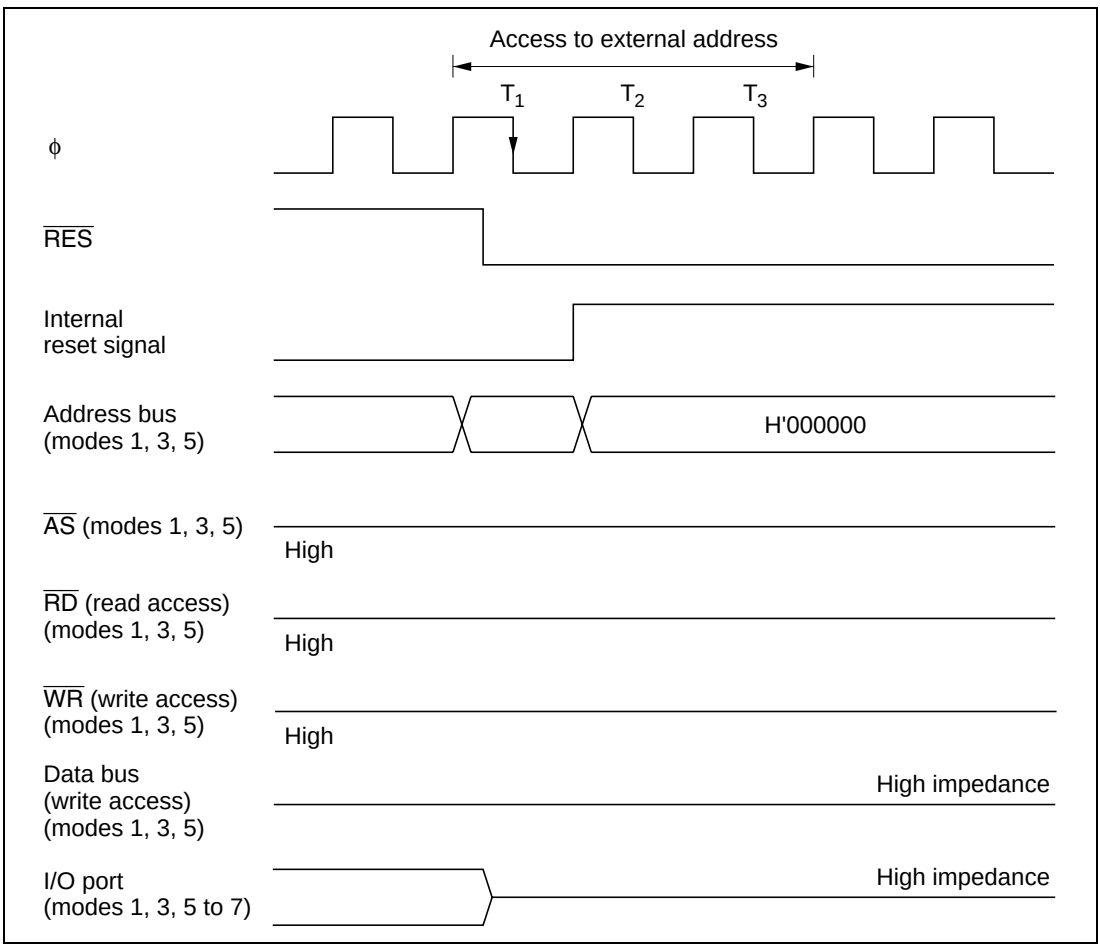


Figure D.1 Reset during Memory Access (Reset during T1 State)

Reset in T2 State

Figure D.2 is a timing diagram for the case in which $\overline{RES}$ goes low during the T2 state of an external memory access cycle. As soon as $\overline{RES}$ goes low, all ports are initialized to the input state. $\overline{AS}$, $\overline{RD}$, and $\overline{WR}$ go high, and the data bus goes to the high-impedance state. The address bus is initialized to the low output level 0.5 state after the low level of $\overline{RES}$ is sampled. The same timing applies when a reset occurs during a wait state (T_w).

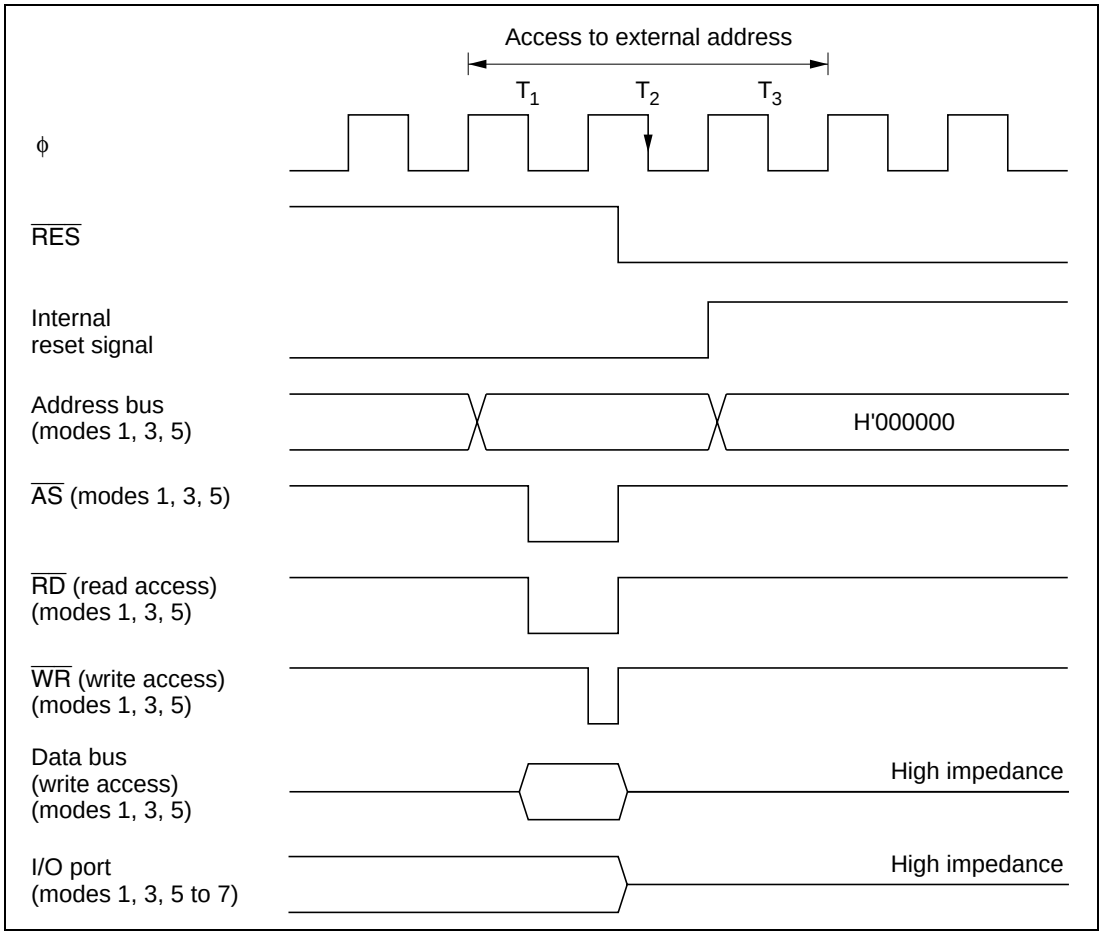


Figure D.2 Reset during Memory Access (Reset during T2 State)

Reset in T3 State

Figure D.3 is a timing diagram for the case in which $\overline{RES}$ goes low during the T_3 state of an external memory access cycle. As soon as $\overline{RES}$ goes low, all ports are initialized to the input state. $\overline{AS}$, $\overline{RD}$, and $\overline{WR}$ go high, and the data bus goes to the high-impedance state. The address bus outputs are held during the T_3 state.The same timing applies when a reset occurs in the T_2 state of an access cycle to a two-state-access area.

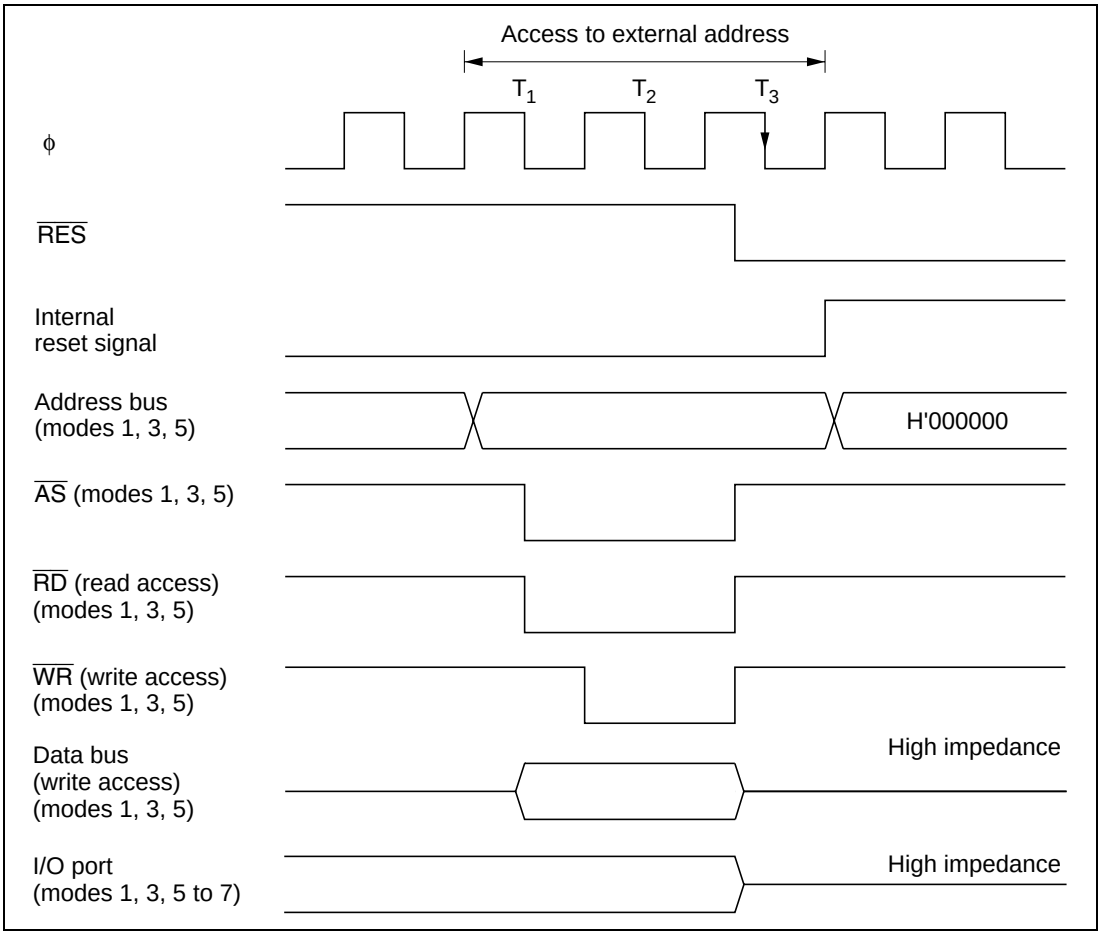
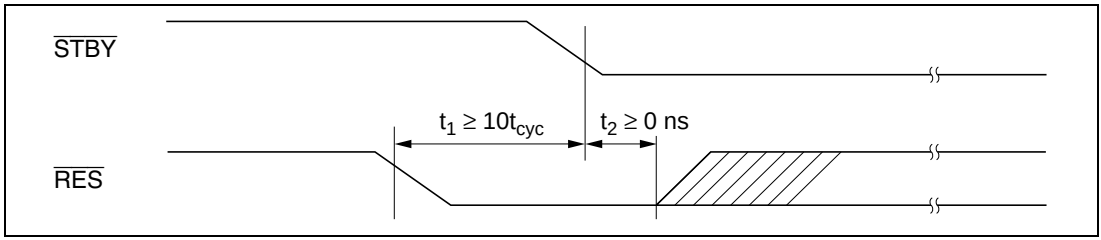


Figure D.3 Reset during Memory Access (Reset during T3 State)

Appendix E Timing of Transition to and Recovery from Hardware Standby Mode

Timing of Transition to Hardware Standby Mode

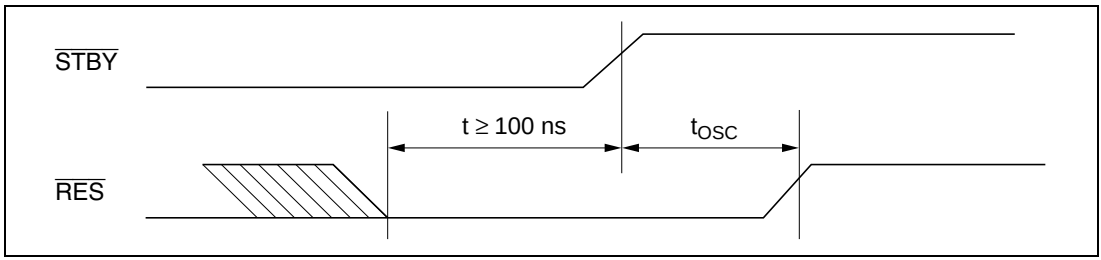
(1) To retain RAM contents with the RAME bit set to 1 in SYSCR, drive the $\overline{\text{RES}}$ signal low 10 system clock cycles before the $\overline{\text{STBY}}$ signal goes low, as shown below. $\overline{\text{RES}}$ must remain low until $\overline{\text{STBY}}$ goes low (minimum delay from $\overline{\text{STBY}}$ low to $\overline{\text{RES}}$ high: 0 ns).



(2) To retain RAM contents with the RAME bit cleared to 0 in SYSCR, $\overline{\text{RES}}$ 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.



Appendix F Product Lineup

Table F.1 H8/3039 Group Product Lineup

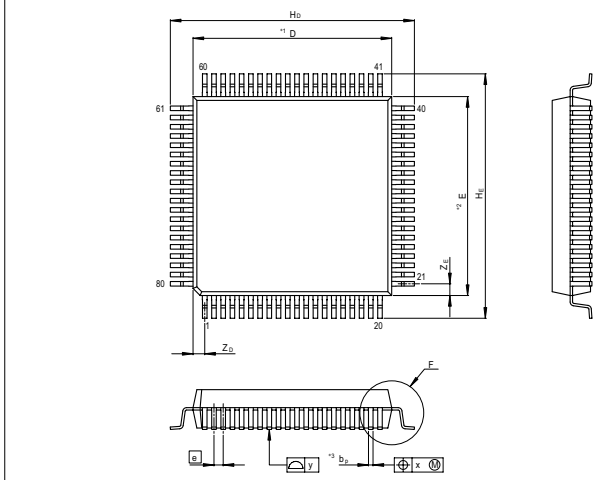
Product Type			Part Number	Mark Code	Package (Package Code)
H8/3039	Flash memory version	5 V version	HD64F3039F	HD64F3039F	80-pin QFP (FP-80A)
			HD64F3039TE	HD64F3039TE	80-pin TQFP (TFP-80C)
		3 V version	HD64F3039VF	HD64F3039VF	80-pin QFP (FP-80A)
			HD64F3039VTE	HD64F3039VTE	80-pin TQFP (TFP-80C)
	Mask ROM version	5 V version	HD6433039F	HD6433039(***)F	80-pin QFP (FP-80A)
			HD6433039TE	HD6433039(***)TE	80-pin TQFP (TFP-80C)
		3 V version	HD6433039VF	HD6433039(***)VF	80-pin QFP (FP-80A)
			HD6433039VTE	HD6433039(***)VTE	80-pin TQFP (TFP-80C)
H8/3038	Mask ROM version	5 V version	HD6433038F	HD6433038(***)F	80-pin QFP (FP-80A)
			HD6433038TE	HD6433038(***)TE	80-pin TQFP (TFP-80C)
		3 V version	HD6433038VF	HD6433038(***)VF	80-pin QFP (FP-80A)
			HD6433038VTE	HD6433038(***)VTE	80-pin TQFP (TFP-80C)
H8/3037	Mask ROM version	5 V version	HD6433037F	HD6433037(***)F	80-pin QFP (FP-80A)
			HD6433037TE	HD6433037(***)TE	80-pin TQFP (TFP-80C)
		3 V version	HD6433037VF	HD6433037(***)VF	80-pin QFP (FP-80A)
			HD6433037VTE	HD6433037(***)VTE	80-pin TQFP (TFP-80C)
H8/3036	Mask ROM version	5 V version	HD6433036F	HD6433036(***)F	80-pin QFP (FP-80A)
			HD6433036TE	HD6433036(***)TE	80-pin TQFP (TFP-80C)
		3 V version	HD6433036VF	HD6433036(***)VF	80-pin QFP (FP-80A)
			HD6433036VTE	HD6433036(***)VTE	80-pin TQFP (TFP-80C)

Note: (***) in mask ROM versions is the ROM code.

Appendix G Package Dimensions

The package dimension that is shown in the Renesas Semiconductor Package Data Book has priority.

JEITA Package Code	RENESAS Code	Previous Code	MASS[Typ.]
P-QFP80-14x14-0.65	PRQP0080JB-A	FP-80A/FP-80AV	1.2g

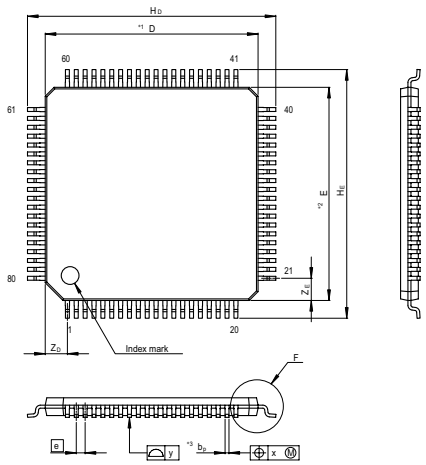


NOTE)
1. DIMENSIONS"*1"AND"*2" DO NOT INCLUDE MOLD FLASH
2. DIMENSION"*3"DOES NOT INCLUDE TRIM OFFSET.

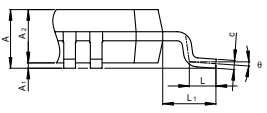
Reference Symbol	Dimension in Millimeters		
	Min	Nom	Max
D	—	14	—
E	—	14	—
A_2	—	2.70	—
H_D	16.9	17.2	17.5
H_E	16.9	17.2	17.5
A	—	—	3.05
A_1	0.00	0.10	0.25
b_p	0.24	0.32	0.40
b_1	—	0.30	—
c	0.12	0.17	0.22
c_1	—	0.15	—
θ	0°	—	8°
e	—	0.65	—
x	—	—	0.12
y	—	—	0.10
Z_D	—	0.83	—
Z_E	—	0.83	—
L	0.5	0.8	1.1
L_1	—	1.6	—

Figure G.1 Package Dimensions (FP-80A)

JEITA Package Code	RENESAS Code	Previous Code	MASS[Typ.]
P-TQFP80-12x12-0.50	PTQP0080KC-A	TFP-80C/TFP-80CV	0.4g



Terminal cross section



Detail F

NOTE)
1. DIMENSIONS"*1"AND"*2" DO NOT INCLUDE MOLD FLASH
2. DIMENSION"*3"DOES NOT INCLUDE TRIM OFFSET.

Reference Symbol	Dimension in Millimeters		
	Min	Nom	Max
D	—	12	—
E	—	12	—
A_2	—	1.00	—
H_D	13.8	14.0	14.2
H_1	13.8	14.0	14.2
A	—	—	1.20
A_1	0.00	0.10	0.20
b_p	0.17	0.22	0.27
b_1	—	0.20	—
c	0.12	0.17	0.22
c_1	—	0.15	—
θ	0°	—	8°
e	—	0.5	—
x	—	—	0.10
y	—	—	0.10
Z_D	—	1.25	—
Z_E	—	1.25	—
L	0.4	0.5	0.6
L_1	—	1.0	—

Figure G.2 Package Dimensions (TFP-80C)

Renesas 16-Bit Single-Chip Microcomputer
Hardware Manual
H8/3039 Group, H8/3039F-ZTAT™

Publication Date:	1st Edition, December 1997
	Rev.3.00, March 26, 2007
Published by:	Sales Strategic Planning Div.
	Renesas Technology Corp.
Edited by:	Customer Support Department
	Global Strategic Communication Div.
	Renesas Solutions Corp.

Renesas Technology Corp. Sales Strategic Planning Div. Nippon Bldg., 2-6-2, Ohte-machi, Chiyoda-ku, Tokyo 100-0004, Japan

RENESAS SALES OFFICES

http://www.renesas.com

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 Ehsan, Malaysia
Tel: <603> 7955-9390, Fax: <603> 7955-9510

H8/3039 Group, H8/3039F-ZTAT™
Hardware Manual

Renesas Electronics Corporation
1753, Shimonumabe, Nakahara-ku, Kawasaki-shi, Kanagawa 211-8668 Japan

REJ09B0353-0300