To our customers,

Old Company Name in Catalogs and Other Documents

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

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

April 1st, 2010
Renesas Electronics Corporation

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

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

Notice

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

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

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

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

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

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

User's Manual

16

H8/3039 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

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, algorith application circuit examples, is current as of the date this document is issued. Such information, hov subject to change without any prior notice. Before purchasing or using any Renesas products listed document, please confirm the latest product information with a Renesas sales office. Also, please p and careful attention to additional and different information to be disclosed by Renesas such as that through our website. (http://www.renesas.com)
5. Renesas has used reasonable care in compiling the information included in this document, but Rene assumes no liability whatsoever for any damages incurred as a result of errors or omissions in the in included in this document.
6. When using or otherwise relying on the information in this document, you should evaluate the inform light of the total system before deciding about the applicability of such information to the intended ap Renesas makes no representations, warranties or guaranties regarding the suitability of its products particular application and specifically disclaims any liability arising out of the application and use of t information in this document or Renesas products.
7. With the exception of products specified by Renesas as suitable for automobile applications, Renes products are not designed, manufactured or tested for applications or otherwise in systems the failu malfunction of which may cause a direct threat to human life or create a risk of human injury or whic especially high quality and reliability such as safety systems, or equipment or systems for transporta traffic, healthcare, combustion control, aerospace and aeronautics, nuclear power, or undersea com transmission. If you are considering the use of our products for such purposes, please contact a Re sales office beforehand. Renesas shall have no liability for damages arising out of the uses set forth
8. Notwithstanding the preceding paragraph, you should not use Renesas products for the purposes lis
   (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 purcha elect to use Renesas products in any of the foregoing applications shall indemnify and hold harmles Technology Corp., its affiliated companies and their officers, directors, and employees against any a damages arising out of such applications.
9. You should use the products described herein within the range specified by Renesas, especially wit to the maximum rating, operating supply voltage range, movement power voltage range, heat radiat characteristics, installation and other product characteristics. Renesas shall have no liability for malf 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 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 or damage caused by fire in the event of the failure of a Renesas product, such as safety desi hardware and software including but not limited to redundancy, fire control and malfunction preventi appropriate treatment for aging degradation or any other applicable measures. Among others, since evaluation of microcomputer software alone is very difficult, please evaluate the safety of the final pr system manufactured by you.
11. In case Renesas products listed in this document are detached from the products to which the Rene products are attached or affixed, the risk of accident such as swallowing by infants and small childre high. You should implement safety measures so that Renesas products may not be easily detached 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 w approval from Renesas.
13. Please contact a Renesas sales office if you have any questions regarding the information containe document, Renesas semiconductor products, or if you have any other inquiries.

— The input pins of CMOS products are generally in the high-impedance state. I
operation with an unused pin in the open-circuit state, extra electromagnetic n
induced in the vicinity of LSI, an associated shoot-through current flows intern
malfunctions may occur due to the false recognition of the pin state as an inpu
Unused pins should be handled as described under Handling of Unused Pins
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 re
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
states of pins are not guaranteed from the moment when power is supplied un
reset process is completed.

In a similar way, the states of pins in a product that is reset by an on-chip pow
reset function are not guaranteed from the moment when power is supplied u
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 fun
not access these addresses; the correct operation of LSI is not guaranteed if
accessed.

4. Clock Signals

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

— When the clock signal is generated with an external resonator (or from an ext
oscillator) during a reset, ensure that the reset line is only released after full st
of the clock signal. Moreover, when switching to a clock signal produced with
external resonator (or by an external oscillator) while program execution is in
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 num
confirm that the change will not lead to problems.

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

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

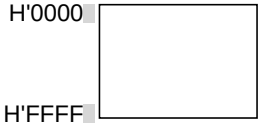
The five MCU operating modes offer a choice of expanded mode, single-chip mode, a
space size, enabling the H8/3039 Group to adapt quickly and flexibly to a variety of c

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

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

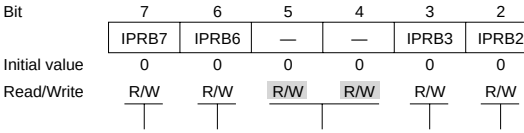
Note: F-ZTAT is a trademark of Renesas Technology Corp.

Figure 2.2 Memory Map

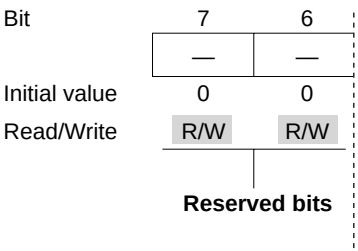


1. Normal mode (64-Kbyte mode)

5.2.2 Interrupt Priority Registers A and B (IPRA, IPRB) Interrupt Priority Register B (IPRB)	91	Description amended

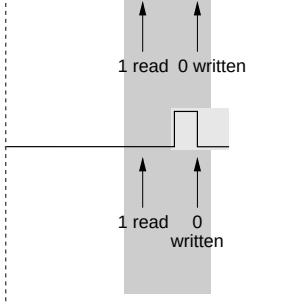


5.2.3 IRQ Status Register (ISR)	93	Description amended Bits 5, 4, 1 and 0—IRQ_5, IRQ_4, IRQ_1 and IRQ_0 Flags IRQ4F, IRQ1F, and IRQ0F): These bits indicate the IRQ_5, IRQ_4, IRQ_1, and IRQ_0 interrupt requests.
5.2.4 IRQ Enable Register (IER)	94	Description amended



Bits 5, 4, 1, and 0—IRQ_5, IRQ_4, IRQ_1, and IRQ_0 Ena
IRQ4E, IRQ1E, IRQ0E): These bits enable or disab
IRQ4, IRQ1, IRQ0 interrupts.

Exception Handling is not Executed



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 individual channels, so different values may be set fo
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 or an external clock input from the SCK pin can be s clearing or setting the CKE1 and CKE0 bits in SCR a bit in SMR. See table 11.9.

$\overline{RES}$ pulse width	t_{RESW}	10	—	10	—	10	—	tcy
Mode programming setup time (MD_{ss} MD_{1}, MD_{0})	t_{MDS}	200	—	200	—	200	—	ns

18.1.4 A/D Conversion Characteristics	535	Newly added
18.2.2 DC Characteristics Table 18.10 Permissible Output Currents	541	Table amended **Item** Permissible output low current (total): Total of 27 pins including ports 1, 2, 5 and B
A.1 Instruction List 8. Block transfer instructions	576	Table amended **Mnemonic**: EEPMOV. W **Operand Size**: — **Operation**: if R4 ≠ 0 then repeat @R5 → @R6 R5+1 → R5 R6+1 → R6 R4–1 → R4 until R4=0 else next
A.3 Number of States Required for Execution Table A.4 Number of Cycles per Instruction	584	Table amended

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.

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

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

The H8/3039 Group consists of four models: the H8/3039 with 128 kbytes of ROM a
of RAM, the H8/3038 with 64 kbytes of ROM and 2 kbytes of RAM, the H8/3037 wi
of ROM and 1 kbytes of RAM, and the H8/3036 with 16 kbytes of ROM and 512 byt

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

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

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

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

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 bi
- Signed and unsigned divide instructions (16 bits ÷ 8 bits, 32 bit
- Bit accumulator function
- Bit manipulation instructions with register-indirect specification positions

	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 indep 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 pul or 10 pulse inputs • 16-bit timer counter (channels 0 to 4) • Two multiplexed output compare/input capture pins (channels • 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><td></td><td>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>

HD6433038F	HD6433038VF	80-pin QFP (FP-80A)	Mas
HD6433038TE	HD6433038VTE	80-pin TQFP (TFP-80C)	
HD6433037F	HD6433037VF	80-pin QFP (FP-80A)	Mas
HD6433037TE	HD6433037VTE	80-pin TQFP (TFP-80C)	
HD6433036F	HD6433036VF	80-pin QFP (FP-80A)	Mas
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
and one with V_{cc} = 3.0 V to 5.5 V and ϕ = 2 to 10 MHz. Howeve
one flash memory version, with V_{cc} = 3.0 to 5.5 V and ϕ = 2 to 1

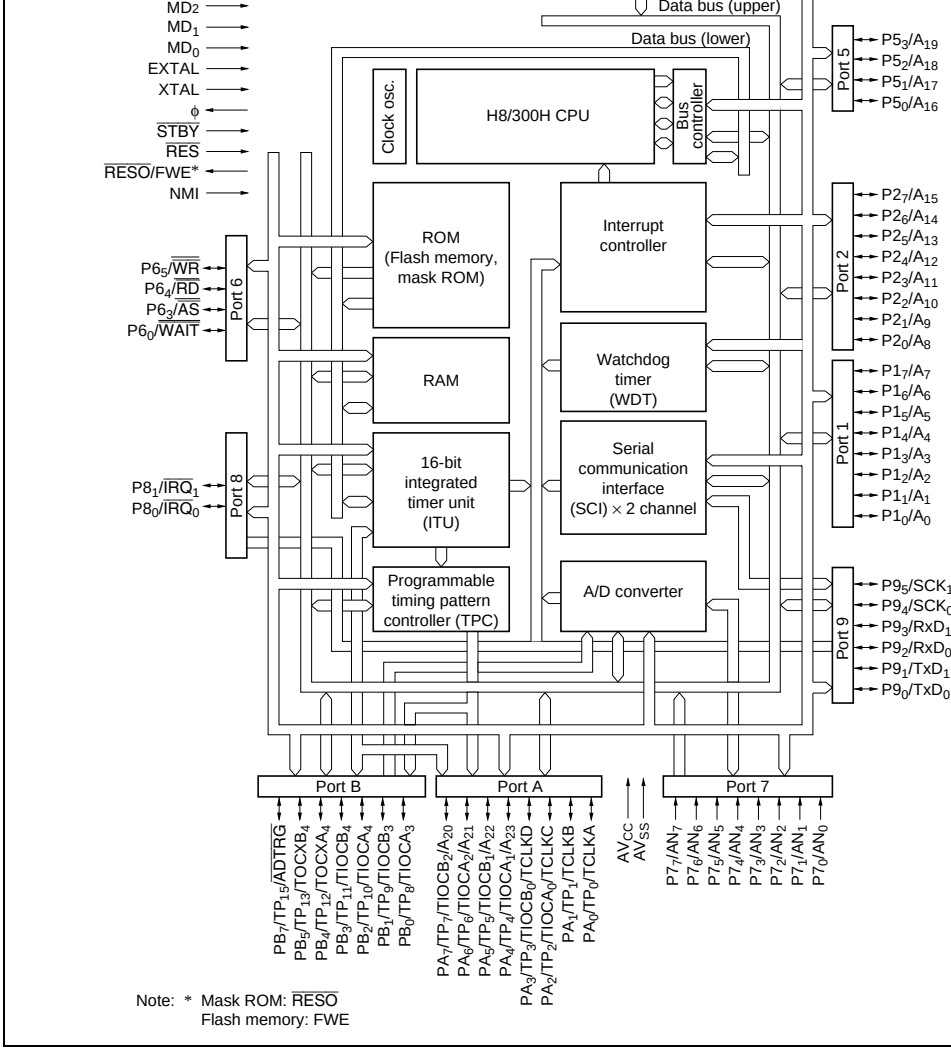


Figure 1.1 Block Diagram

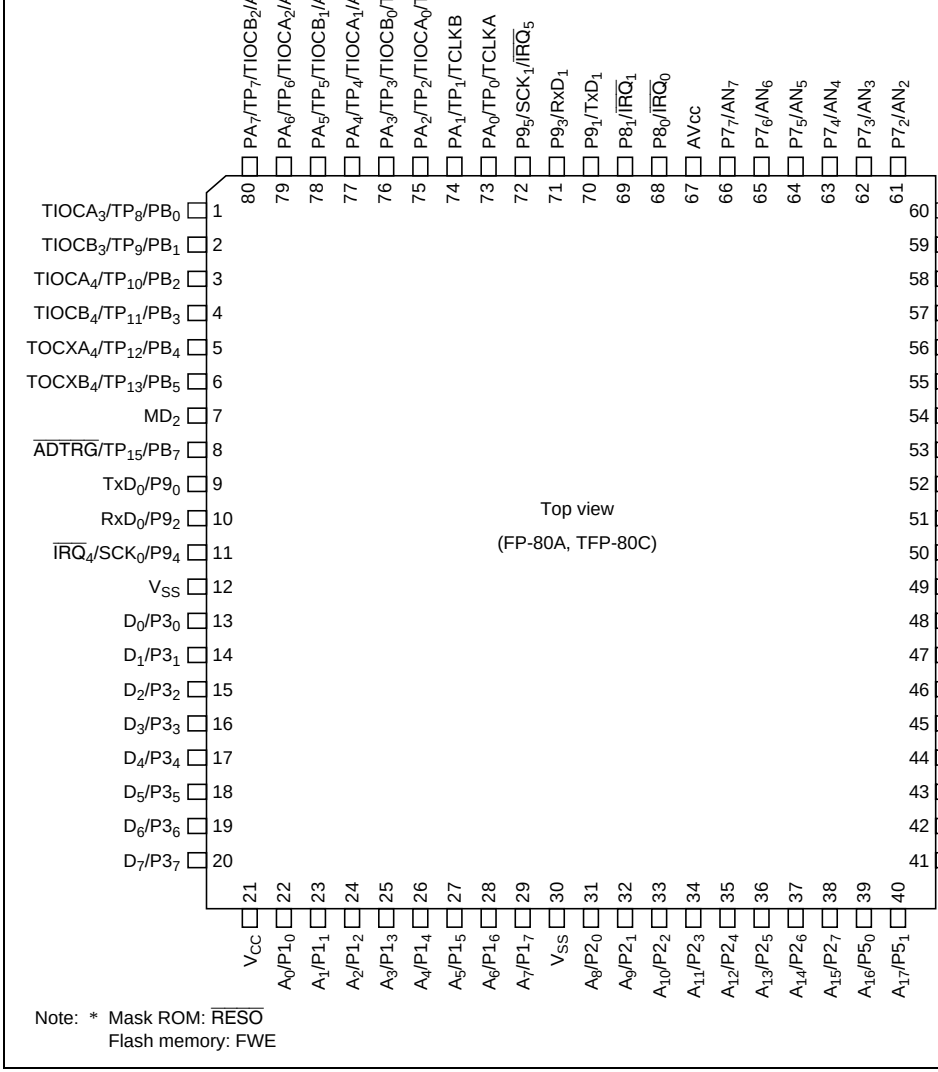


Figure 1.2 Pin Arrangement (FP-80A, TFP-80C Top View)

Pin No.	Mode 1	Mode 3	Mode 5	Mode 6	Mode 7	PROM Flash
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	$PB_7/TP_{15}/\overline{ADTRG}$	$PB_7/TP_{15}/\overline{ADTRG}$	$PB_7/TP_{15}/\overline{ADTRG}$	$PB_7/TP_{15}/\overline{ADTRG}$	$PB_7/TP_{15}/\overline{ADTRG}$	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	$P9_4/SCK_0/\overline{IRQ_4}$	$P9_4/SCK_0/\overline{IRQ_4}$	$P9_4/SCK_0/\overline{IRQ_4}$	$P9_4/SCK_0/\overline{IRQ_4}$	$P9_4/SCK_0/\overline{IRQ_4}$	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

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

51	EXTAL	EXTAL	EXTAL	EXTAL	EXTAL	EXTA
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	$\overline{CE}$

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: $\overline{\text{RESO}}$
Flash Memory: FWE

				supply.
	V_{SS}	12, 30, 50	Input	**Ground:** For connection to ground Connect all V_{SS} pins to the 0-V syst supply.
Clock	XTAL	52	Input	For connection to a crystal resonat For examples of crystal resonator a external clock input, see section 16 Pulse Generator.
	EXTAL	51	Input	For connection to a crystal resonat of an external clock signal. For exa crystal resonator and external cloc section 16, Clock Pulse Generator.
	ϕ	46	Output	System clock: Supplies the system external devices
Operating mode control	MD_2, MD_1, MD_0	7, 45, 44	Input	**Mode 2 to mode 0:** For setting the mode, as follows. These pins shou changed during operation.

MD_2	**MD_1**	**MD_0**	**Operati**
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

	$\overline{STBY}$	47	Input	**Standby:** When driven low, this p transition to hardware standby mo
Interrupts	NMI	49	Input	**Nonmaskable interrupt:** Reques 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:** Mask 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 s
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 ind address output on the address bu
	$\overline{RD}$	55	Output	**Read:** Goes low to indicate readi external address space.
	$\overline{WR}$	56	Output	**Write:** Goes low to indicate writin external address space indicates the data bus.
	$\overline{WAIT}$	43	Input	**Wait:** Requests insertion of wait s cycles during access to the extern space.
16-bit integrated timer unit (ITU)	TCLKD to TCLKA	76 to 73	Input	**Clock input A to D:** External cloc
	$TIOCA_4$ to $TIOCA_0$	3, 1, 79, 77, 75	Input/ Output	**Input capture/output compare A** GRA4 to GRA0 output compare o capture, or PWM output
	$TIOCB_4$ to $TIOCB_0$	4, 2, 80, 78, 76	Input/ output	**Input capture/output compare B** GRB4 to GRB0 output compare o capture, or PWM output
	$TOCXA_4$	5	Output	**Output compare XA4:** PWM outp
	$TOCXB_4$	6	Output	**Output compare XB4:** PWM outp

	RxD_0			
	SCK_1, SCK_0	72, 11	Input/ output	**Serial clock:**(channels 0 and 1): S 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 A/D conversion
	AV_{CC}	67	Input	Power supply pin and reference vo pin for the A/D converter. Connect system power supply when not usi converter.
	AV_{SS}	58	Input	Ground pin for the A/D converter. C 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 of each pin can be selected in the direction register (P1DDR).
	$P2_7$ to $P2_0$	38 to 31	Input/ output	**Port 2:** Eight input/output pins. The of each pin can be selected in the direction register (P2DDR).
	$P3_7$ to $P3_0$	20 to 13	Input/ output	**Port 3:** Eight input/output pins. The of each pin can be selected in the direction register (P3DDR).
	$P5_3$ to $P5_0$	42 to 39	Input/ output	**Port 5:** Four input/output pins. The each pin can be selected in the por direction register (P5DDR).
	$P6_5$ to $P6_3$, $P6_0$	56 to 54, 43	Input/ output	**Port 6:** Four input/output pins. The each pin can be selected in the por 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 each pin can be selected in the por direction register (P8DDR).

	to PB_0	output	of each pin can be selected in the direction register (PBDDR).

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-bi
- Sixty-two basic instructions
  - 8/16/32-bit arithmetic and logic instructions
  - Multiply and divide instructions
  - Powerful bit-manipulation instructions
- Eight addressing modes
  - Register direct [Rn]
  - Register indirect [@ERn]
  - Register indirect with displacement [@(d:16, ERn) or @(d:24, ERn)]
  - Register indirect with post-increment or pre-decrement [@ERn+ or @–ERn]
  - Absolute address [@aa:8, @aa:16, or @aa:24]
  - Immediate [#xx:8, #xx:16, or #xx:32]
  - Program-counter relative [@(d:8, PC) or @(d:16, PC)]
  - Memory indirect [@@aa:8]
- 16-Mbyte linear address space
- 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

2.1.2 Differences from H8/300 CPU

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

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

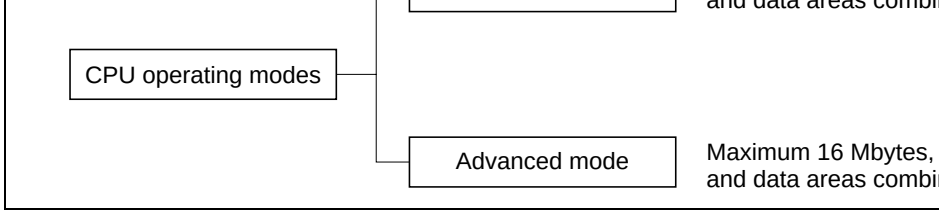


Figure 2.1 CPU Operating Modes

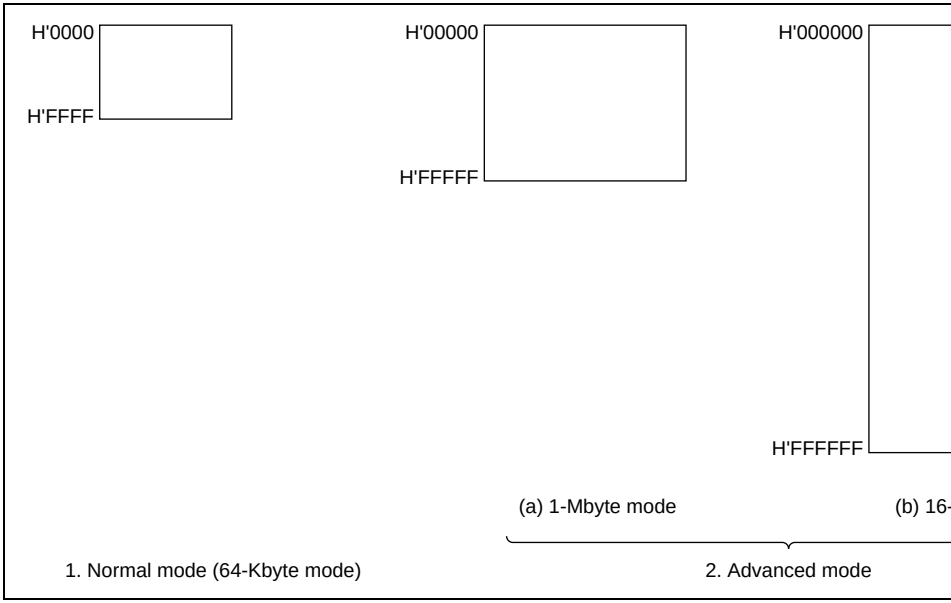
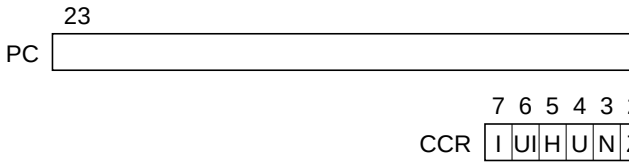


Figure 2.2 Memory Map

General Registers (ERn)

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

Control Registers (CR)



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

Figure 2.3 CPU Registers

(R0 to R7). These registers are functionally equivalent, providing a maximum sixteen 1
registers. The E registers (E0 to E7) are also referred to as extended registers.

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

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

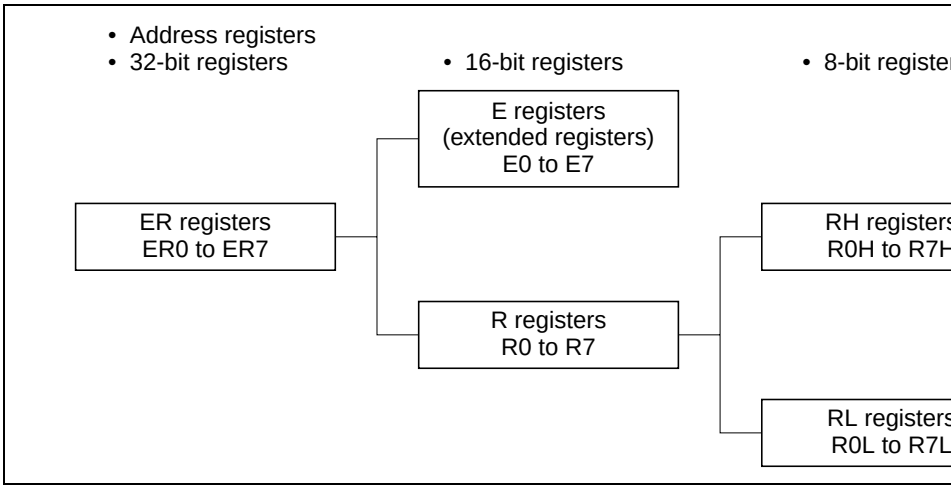


Figure 2.4 Usage of General Registers

Figure 2.5 Stack

2.4.3 Control Registers

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

Program Counter (PC)

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

Condition Code Register (CCR)

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

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

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

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-ze

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

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

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

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

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

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

2.4.4 Initial CPU Register Values

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

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

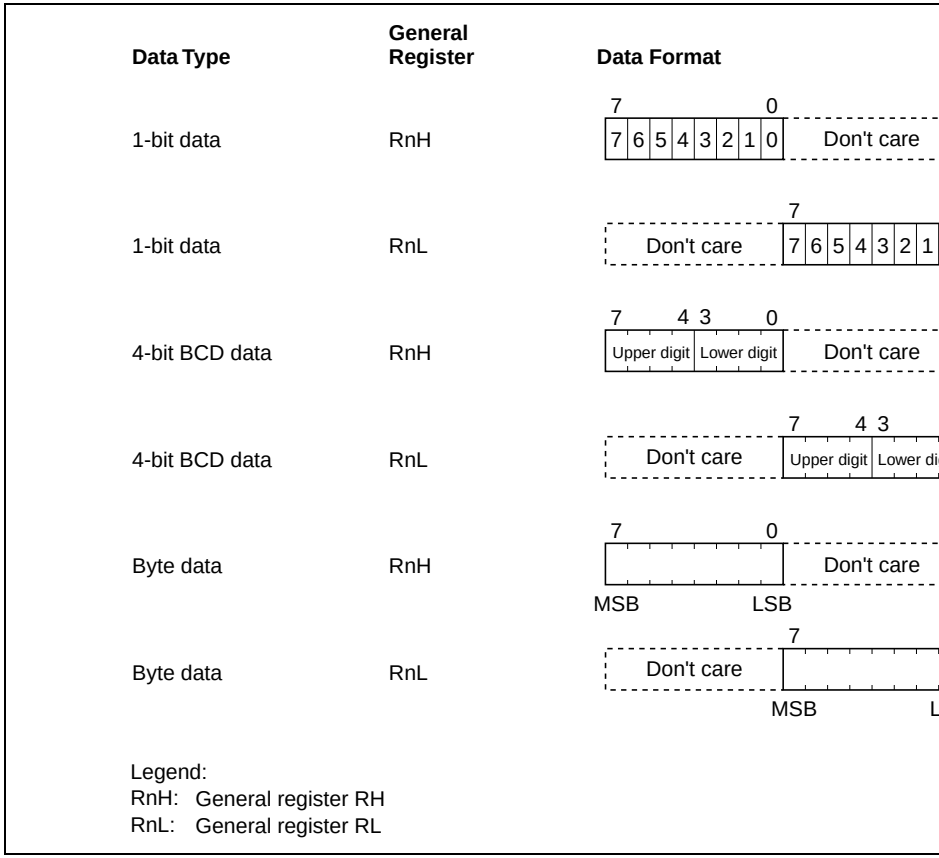
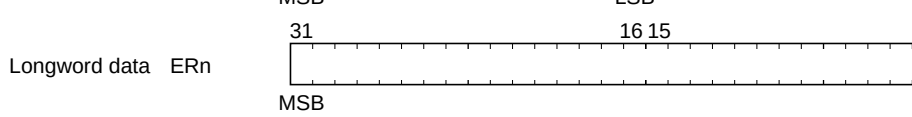


Figure 2.6 General Register Data Formats



Legend:
ERn: General register
En: General register E
Rn: General register R
MSB: Most significant bit
LSB: Least significant bit

Figure 2.7 General Register Data Formats

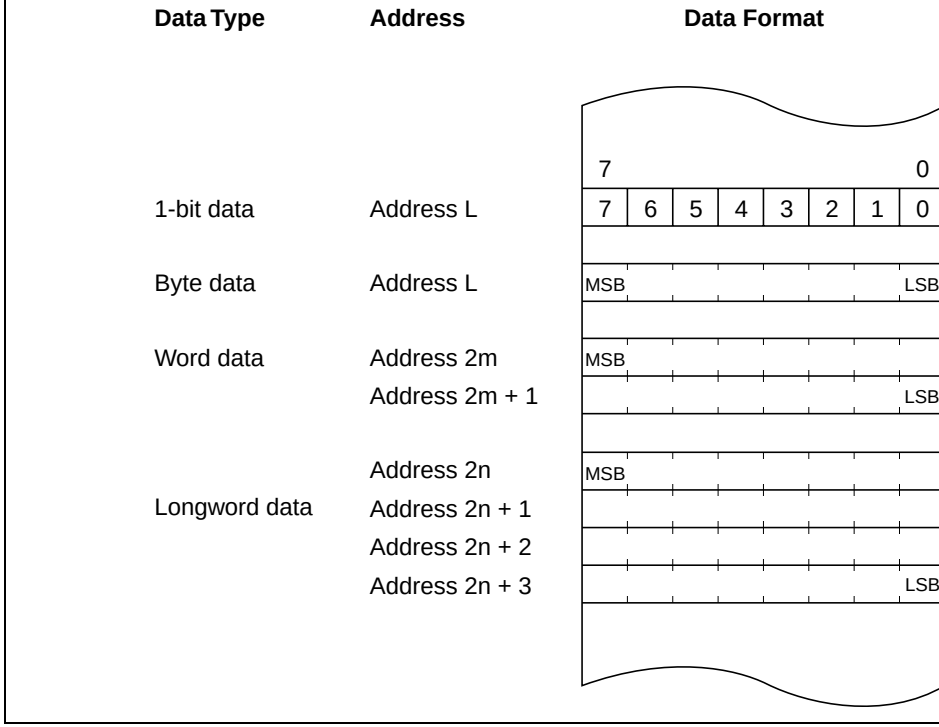


Figure 2.8 Memory Data Formats

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

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

To

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.

Function	Instruction	#xx	Rn	@ERn	@(d:16,ERn)	@(d:24, ERn)	@ERn+/@-ERn	@aa:8	@aa:16	@aa:24	@(d:8, PC)	@(d:16,PC)
Data transfer	MOV	BWL	BWL	BWL	BWL	BWL	BWL	B	BWL	BWL	—	—
	POP, PUSH	—	—	—	—	—	—	—	—	—	—	—
	MOVFPE, MOVTPE	—	—	—	—	—	—	—	B	—	—	—
Arithmetic operations	ADD, CMP	BWL	BWL	—	—	—	—	—	—	—	—	—
	SUB	WL	BWL	—	—	—	—	—	—	—	—	—
	ADDX, SUBX	B	B	—	—	—	—	—	—	—	—	—
	ADDS, SUBS	—	L	—	—	—	—	—	—	—	—	—
	INC, DEC	—	BWL	—	—	—	—	—	—	—	—	—
	DAA, DAS	—	B	—	—	—	—	—	—	—	—	—
	MULXU, MULXS, DIVXU, DIVXS	—	BW	—	—	—	—	—	—	—	—	—
	NEG	—	BWL	—	—	—	—	—	—	—	—	—
	EXTU, EXTS	—	WL	—	—	—	—	—	—	—	—	—
Logic operations	AND, OR, XOR	BWL	BWL	—	—	—	—	—	—	—	—	—
	NOT	—	BWL	—	—	—	—	—	—	—	—	—
Shift instructions		—	BWL	—	—	—	—	—	—	—	—	—
Bit manipulation		—	B	B	—	—	—	B	—	—	—	—
Branch	Bcc, BSR	—	—	—	—	—	—	—	—	—	○	○
	JMP, JSR	—	—	○	—	—	—	—	—	○	—	—
	RTS	—	—	—	—	—	—	—	—	—	—	—

	SLEEP	—	—	—	—	—	—	—	—	—	—	—
	LDC	B	B	W	W	W	W	—	W	W	—	—
	STC	—	B	W	W	W	W	—	W	W	—	—
	ANDC, ORC, XORC	B	—	—	—	—	—	—	—	—	—	—
	NOP	—	—	—	—	—	—	—	—	—	—	—
Block data transfer		—	—	—	—	—	—	—	—	—	—	—

Legend:

B: Byte

W: Word

L: Longword

Rn	General register*
ERn	General register (32-bit register 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 r
to R7, E0 to E7), and 32-bit data or address registers (ER0 to ER7).

MOVTPE	B	Rs → (EAs) Cannot be used in the H8/3039 Group.
POP	W/L	@SP+ → Rn Pops a general register from the stack. POP.W Rn is identi MOV.W @SP+, Rn. Similarly, POP.L ERn is identical to MO @SP+, ERn.
PUSH	W/L	Rn → @–SP Pushes a general register onto the stack. PUSH.W Rn is id MOV.W Rn, @–SP. Similarly, PUSH.L ERn is identical to M @–SP.

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

SUBX		Performs addition or subtraction with carry on data in two ge registers, or on immediate data and data in a general registe
INC, DEC	B/W/L	Rd ± 1 → Rd, Rd ± 2 → Rd Increments or decrements a general register by 1 or 2. (Byte 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
DAA, DAS	B	Rd decimal adjust → Rd Decimal-adjusts an addition or subtraction result in a genera referring to CCR to produce 4-bit BCD data.
MULXU	B/W	Rd × Rs → Rd Performs unsigned multiplication on data in two general regi 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 registe 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

		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 ge register or with immediate data, and sets CCR according to
NEG	B/W/L	0 – Rd → Rd Takes the two's complement (arithmetic complement) of dat general register.
EXTS	W/L	Rd (sign extension) → Rd Extends byte data in the lower 8 bits of a 16-bit register to w or extends word data in the lower 16 bits of a 32-bit register 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 w or extends word data in the lower 16 bits of a 32-bit register longword data, by padding with zeros.

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

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

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

Table 2.6 Shift Instructions

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

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

		Clears a specified bit in a general register or memory operan bit number is specified by 3-bit immediate data or the lower general register.
BNOT	B	¬ (<bit-No.> of <EAd>) → (<bit-No.> of <EAd>) Inverts a specified bit in a general register or memory opera number is specified by 3-bit immediate data or the lower 3 b general register.
BTST	B	¬ (<bit-No.> of <EAd>) → Z Tests a specified bit in a general register or memory operan or clears the Z flag accordingly. The bit number is specified 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 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 ge register or memory operand and stores the result in the carr 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 o 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 ger register or memory operand and stores the result in the carr The bit number is specified by 3-bit immediate data.

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

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

BHI	High	$C \vee Z = 0$
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	$Z \vee (N \oplus V)$
BLE	Less or equal	$Z \vee (N \oplus V)$

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

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

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

repeat @ER5+ → @ER6+, R4 – 1 → R4

until R4 = 0

else next;

Transfers a data block according to parameters set in gener
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 trans
completed.

2.6.4 Basic Instruction Formats

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

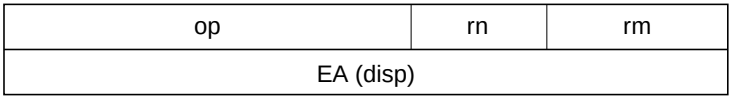
Operation Field: Indicates the function of the instruction, the addressing mode, and th
to be carried out on the operand. The operation field always includes the first 4 bits of t
instruction. Some instructions have two operation fields.

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

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

Condition Field: Specifies the branching condition of Bcc instructions.

Operation field, register fields, and effective address extension



Operation field, effective address extension, and condition field

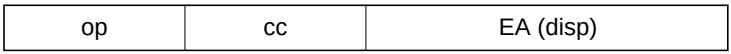


Figure 2.9 Instruction Formats

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 b byte, then write the byte back. Care is required when these instructions are used to acc with write-only bits, or to access ports.

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

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 containi operand. R0H to R7H and R0L to R7L can be specified as 8-bit registers. R0 to R7 and can be specified as 16-bit registers. ER0 to ER7 can be specified as 32-bit registers.

2. Register Indirect—@ERn

The register field of the instruction code specifies an address register (ERn), the lower 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 conten address register (ERn) specified by the register field of the instruction, and the lower 24 sum specify the address of a memory operand. A 16-bit displacement is sign-extended added.

- Register indirect with pre-decrement—@–ERn

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

5. Absolute Address—@aa:8, @aa:16, or @aa:24

The instruction code contains the absolute address of a memory operand. The absolute may be 8 bits long (@aa:8), 16 bits long (@aa:16), or 24 bits long (@aa:24). For an 8- address, the upper 16 bits are all assumed to be 1 (H'FFFF). For a 16-bit absolute addr upper 8 bits are a sign extension. A 24-bit absolute address can access the entire addre 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
24 bits (@aa:24)	H'00000 to H'FFFFF (0 to 1,048,575)	H'000000 to H'FFFFFF (0 to 16,777,215)

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

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

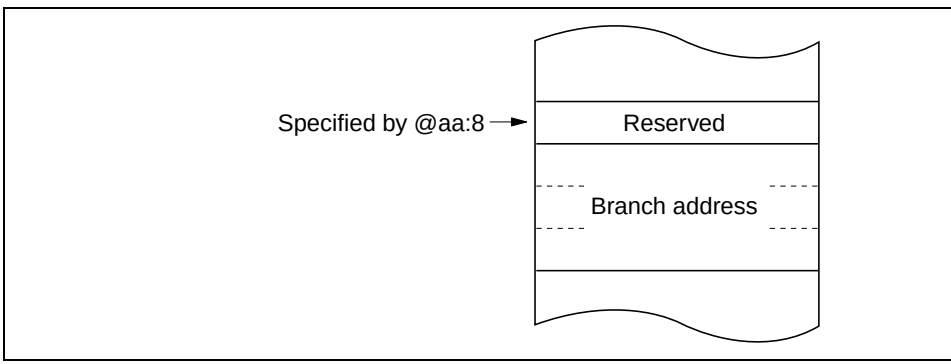
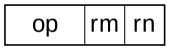
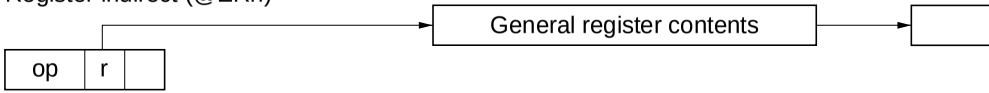
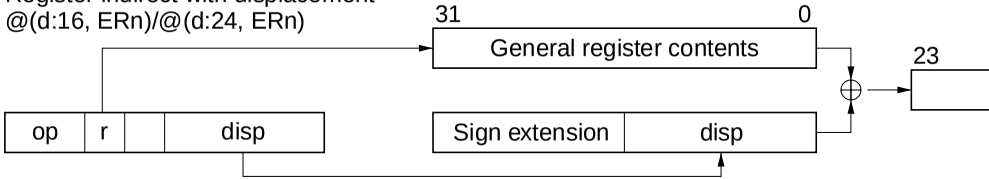
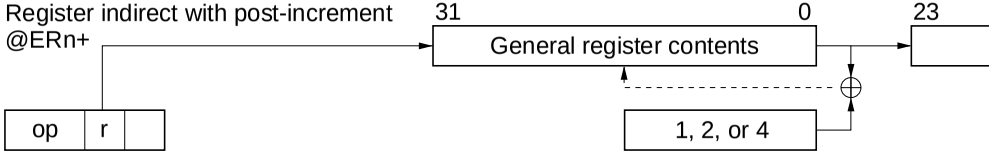
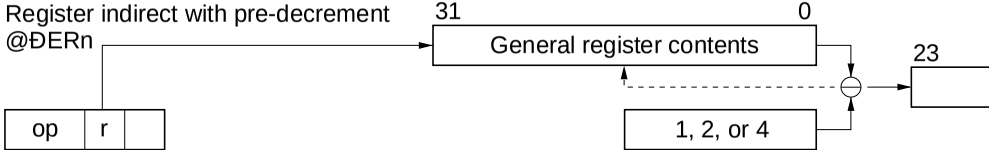
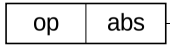
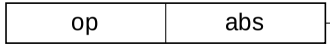
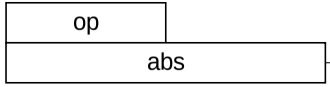
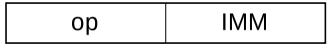
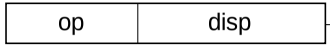
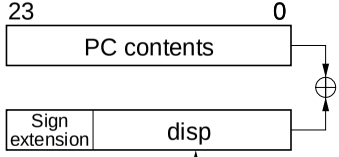


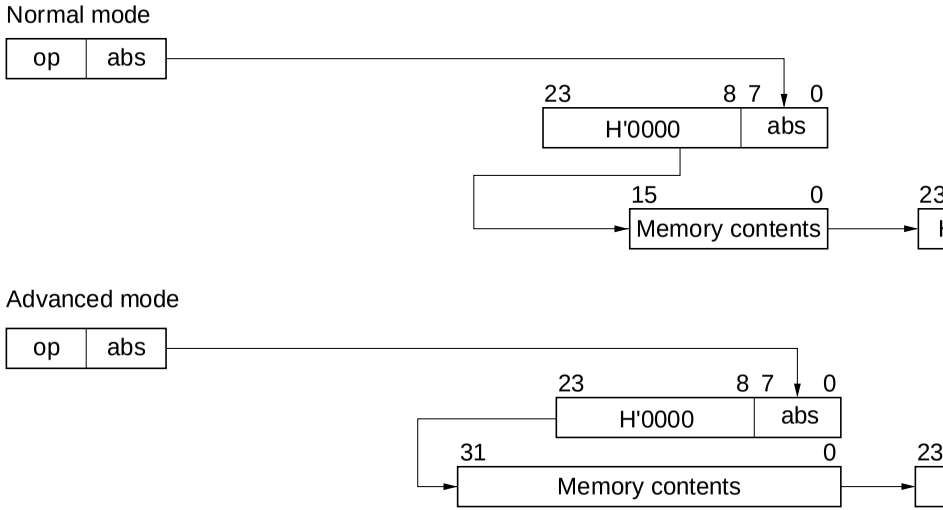
Figure 2.10 Memory-Indirect Branch Address Specification

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

No.	Addressing Mode and Instruction Format	Effective Address Calculation	Effective A
1	Register direct (Rn) 		Operand is register co
2	Register indirect (@ERn) 		
3	Register indirect with displacement @(d:16, ERn)/@(d:24, ERn) 		
4	Register indirect with post-increment or pre-decrement Register indirect with post-increment @ERn+  Register indirect with pre-decrement @ÐERn 		

No.	Addressing Mode and Instruction Format	Effective Address Calculation	Effe
5	Absolute address @aa:8  @aa:16  @aa:24 		23 23 ext 23
6	Immediate #xx:8, #xx:16, or #xx:32 		Oper
7	Program-counter relative @(d:8, PC) or @(d:16, PC) 		23

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



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

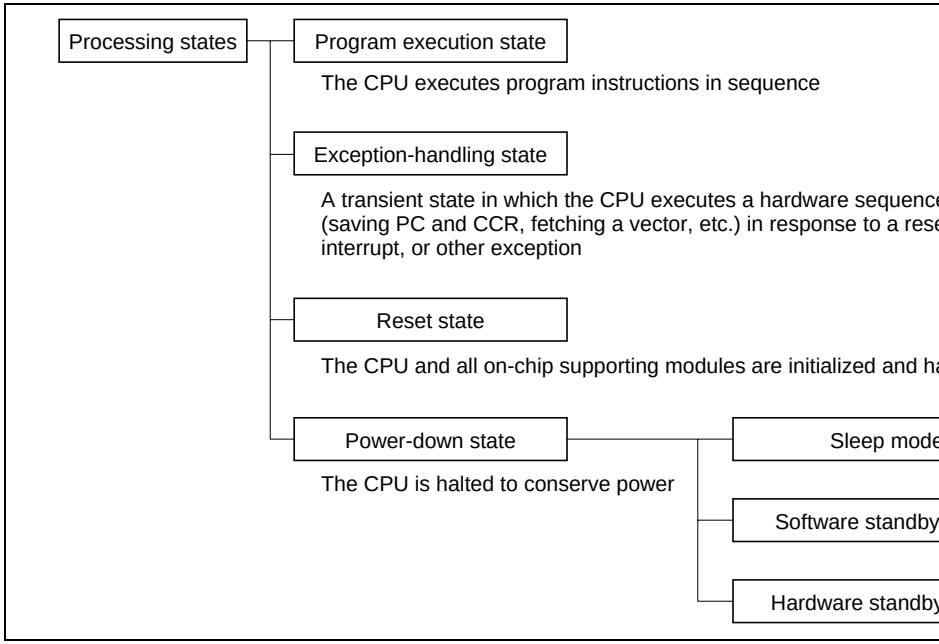


Figure 2.11 Processing States

2.8.2 Program Execution State

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

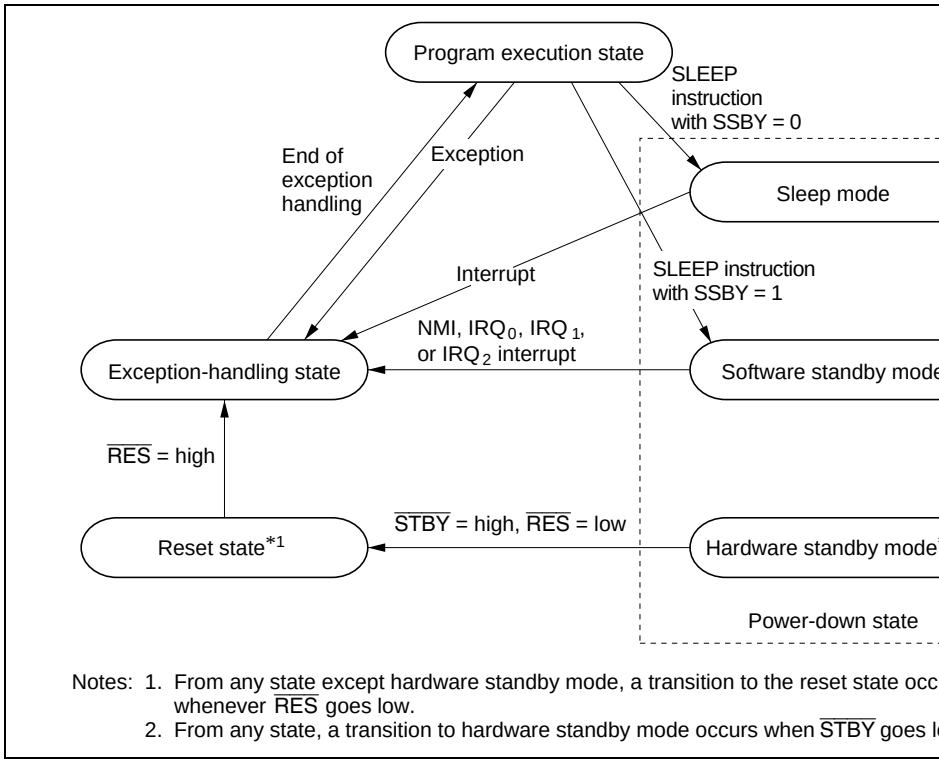
interrupts, and trap instructions. Table 2.14 indicates the types of exception handling an
priority. Trap instruction exceptions are accepted at all times in the program execution

Table 2.14 Exception Handling Types and Priority

Priority	Type of Exception	Detection Timing	Start of Exception H
High ▲	Reset	Synchronized with clock	Exception handling st immediately when RE from low to high
	Interrupt	End of instruction execution or end of exception handling*	When an interrupt is exception handling st end of the current ins current exception-han sequence
	Trap instruction	When TRAPA instruction is executed	Exception handling st a trap (TRAPA) instru executed
Low			

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

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



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

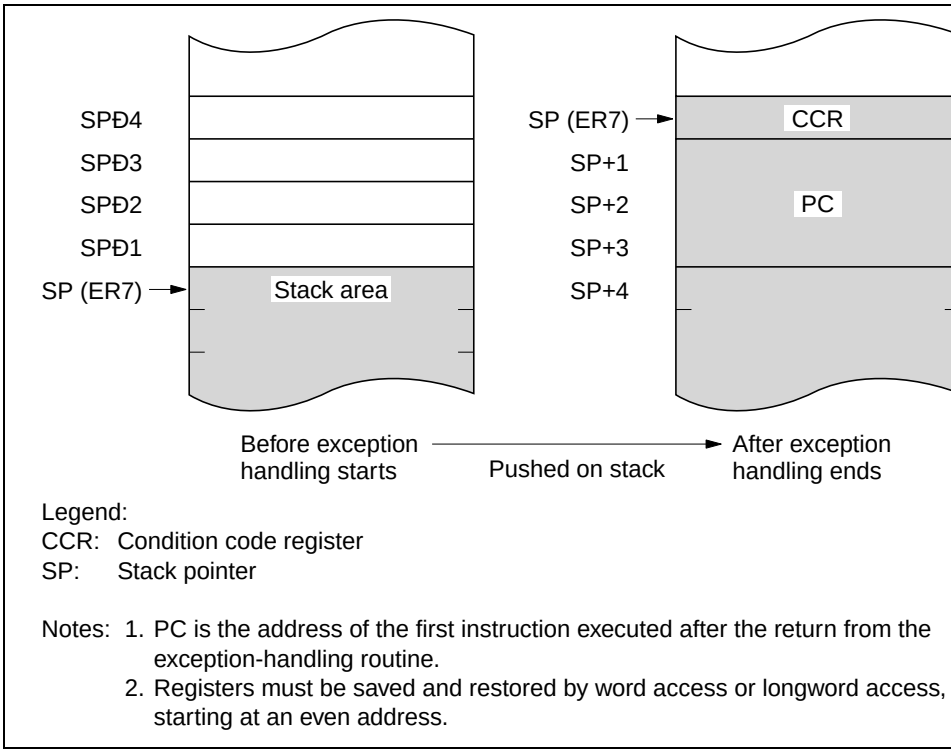
Figure 2.13 State Transitions

2.8.4 Exception-Handling Sequences

Reset Exception Handling: Reset exception handling has the highest priority. The re
entered when the $\overline{RES}$ signal goes low. Reset exception handling starts after that, whe
changes from low to high. When reset exception handling starts the CPU fetches a sta
from the exception vector table and starts program execution from that address. All in

execution branches to that address.

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



Legend:
CCR: Condition code register
SP: Stack pointer

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

Figure 2.14 Stack Structure after Exception Handling

2.8.6 Power-Down State

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

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

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

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

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

controlled by the bus controller.

2.9.2 On-Chip Memory Access Timing

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

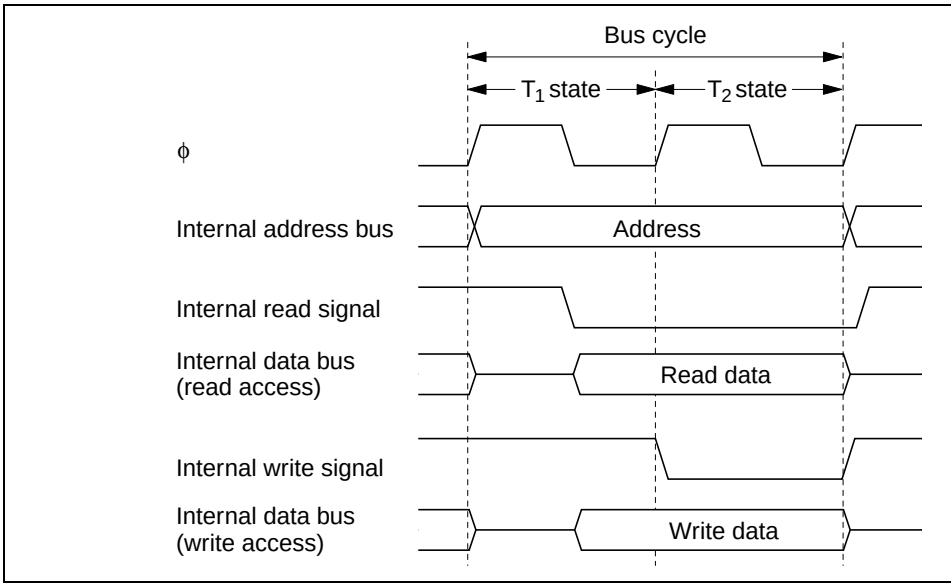


Figure 2.15 On-Chip Memory Access Cycle

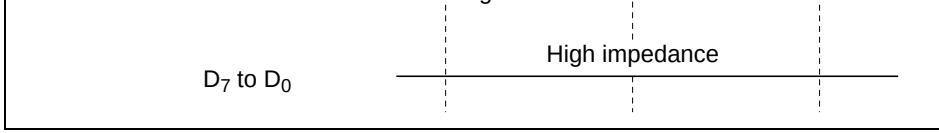


Figure 2.16 Pin States during On-Chip Memory Access

2.9.3 On-Chip Supporting Module Access Timing

The on-chip supporting modules are accessed in three states. The data bus is 8 or 16 b
depending on the register being accessed. Figure 2.17 shows the on-chip supporting m
timing. Figure 2.18 indicates the pin states.

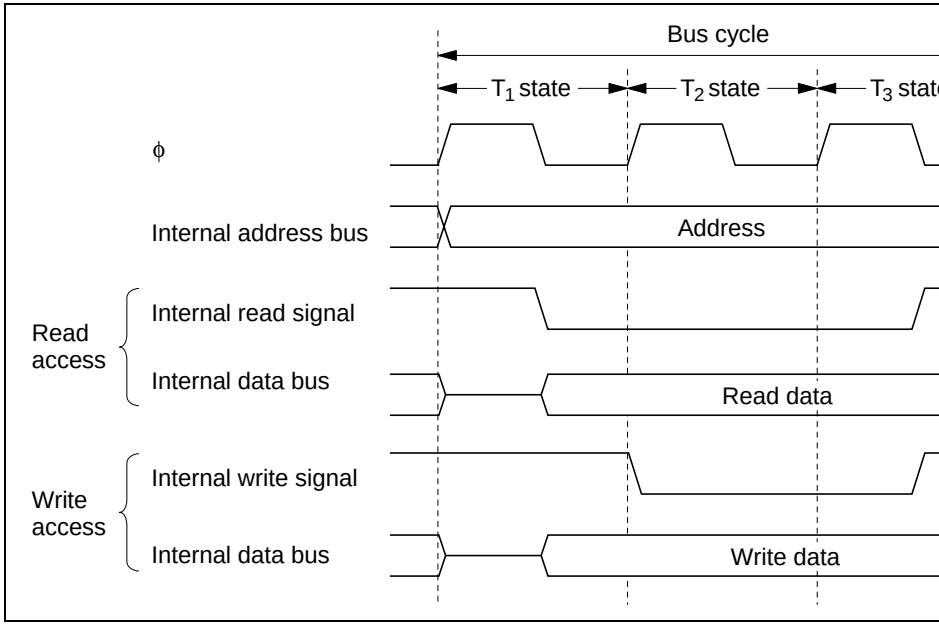


Figure 2.17 Access Cycle for On-Chip Supporting Modules

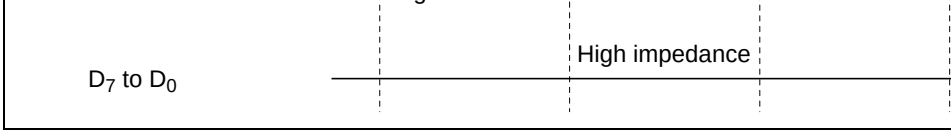


Figure 2.18 Pin States during Access to On-Chip Supporting Modules

2.9.4 Access to External Address Space

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

pins (MD_2 to MD_0) as indicated in table 3.1. The input at these pins determines expand
single-chip mode.

Table 3.1 Operating Mode Selection

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

Notes: 1. If the RAM enable bit (RAME) in the system control register (SYSCR) is cle
these addresses become external addresses.
2. In mode 6 and 7, clearing bit RAME in SYSCR to 0 and reading the on-chip
always return H'FF, and write access is ignored. For details, see section 14
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 extern
and peripheral devices.

Mode 1 supports a maximum address space of 1 Mbyte.

Mode 3 supports a maximum address space of 16 Mbytes.

select one of these seven modes. The inputs at the mode pins must not be changed durin
operation.

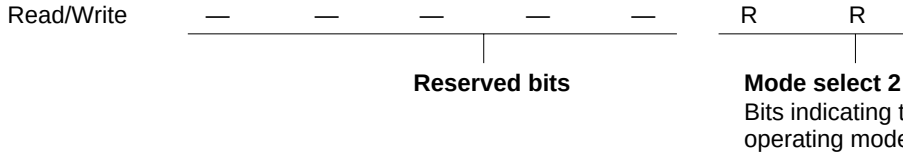
3.1.2 Register Configuration

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

Table 3.2 Registers

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

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

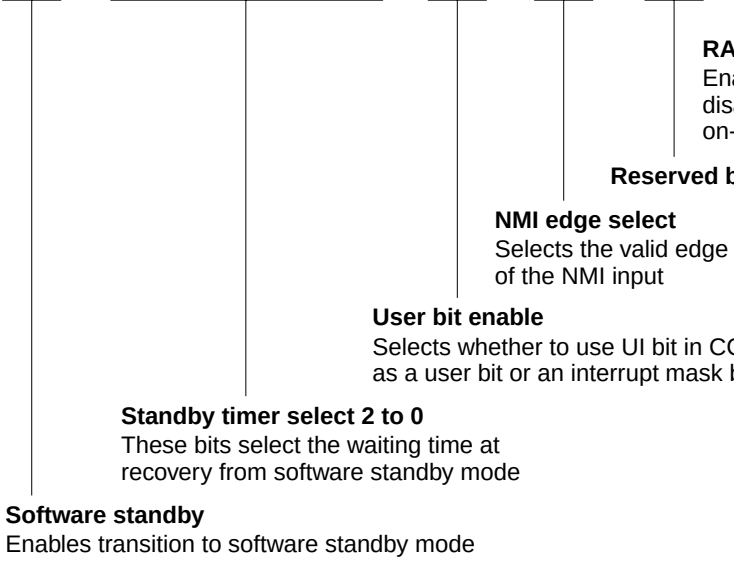


Note: Determined by pins MD_2 to MD_0.

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 MD_2 to MD_0 (the current operating mode). MDS_2 to MDS_0 correspond to MD_2 to MD_0 MDS_0 are read-only bits. The mode pin (MD_2 to MD_0) levels are latched when MDCR



Bit 7—Software Standby (SSBY): Enables transition to software standby mode. (For 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 this bit, write 0.

Bit 7 SSBY	Description	
0	SLEEP instruction causes transition to sleep mode	(Ini
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 tir and on-chip supporting modules wait for the internal clock oscillator to settle when sof standby mode is exited by an external interrupt. Set these bits so that the waiting time w

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 user bit or an interrupt mask bit.

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

Bit 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	(In
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 bi initialized by the rising edge of the $\overline{\text{RES}}$ signal. It is not initialized in software standby

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

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

3.4.3 Mode 5

Ports 1, 2, and 5 can function as address pins A_{19} to A_0, permitting access to a maximun address space, but following a reset they are input ports. To use ports 1, 2, and 5 as an a the corresponding bits in their data direction registers (P1DDR, P2DDR, and P5DDR) r to 1. The address bus width can be selected freely by setting DDR of ports 1, 2, and 5. T 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 avai 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 ROM, RAM, and registers. All I/O ports are available.

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

Port 2	A_{15} to A_8	—	A_{15} to A_8	—	$P2_7$ to $P2_0$*[2]	$P2_7$ to $P2_0$
Port 3	D_7 to D_0	—	D_7 to D_0	—	D_7 to D_0	$P3_7$ to $P3_0$
Port 5	A_{19} to A_{16}	—	A_{19} to A_{16}	—	$P5_3$ to $P5_0$*[2]	$P5_3$ to $P5_0$
Port A	PA_7 to PA_4	—	PA_6 to PA_4*[3], A_{20}	—	PA_7 to PA_4	PA_7 to PA_4

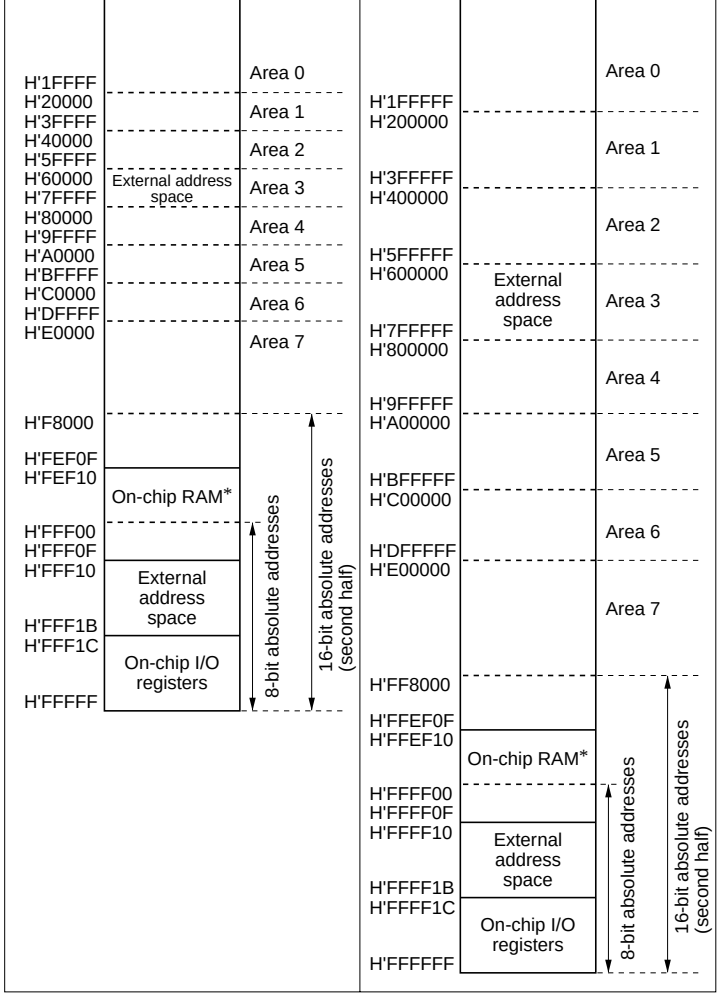
Notes: 1. H8/3039 Group cannot be used in these modes.
2. Initial state. These pins become address output pins when the correspondi data direction registers (P1DDR, P2DDR, P5DDR) are set to 1.
3. Initial state A_{20} is always an address output pin. PA_6 to PA_4 are switched ov A_{21} output by writing 0 in bits 7 to 5 of 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 Figure 3.3 shows a memory map of the H8/3037. Figure 3.4 shows a memory map of The address space is divided into eight areas.

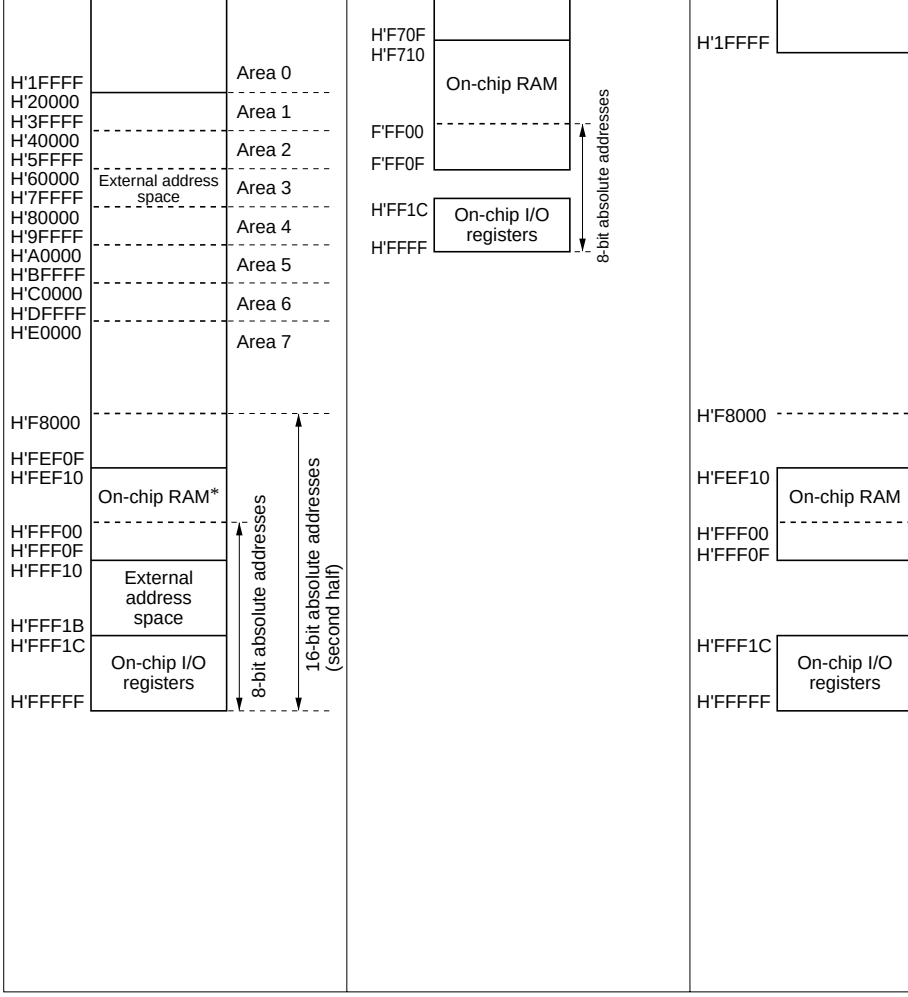
Modes 1, 3 and 5 are the 8-bit bus mode.

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



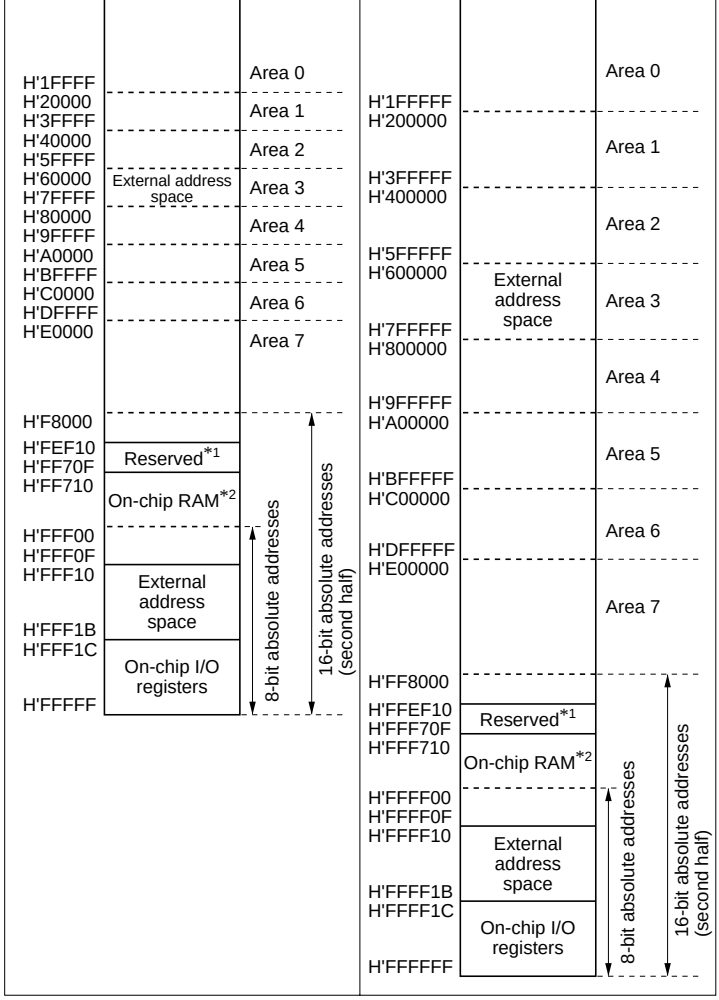
Note: * External addresses can be accessed by disabling on-chip RAM.

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



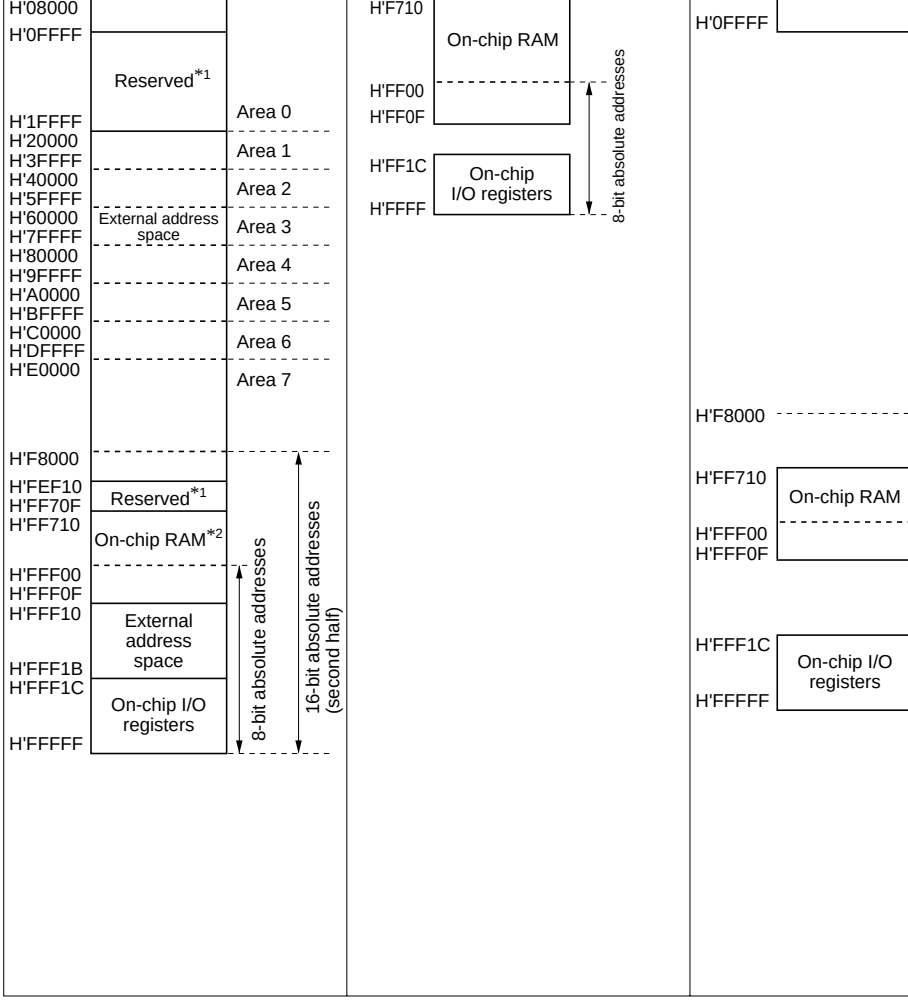
Note: * External addresses can be accessed by disabling on-chip RAM.

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



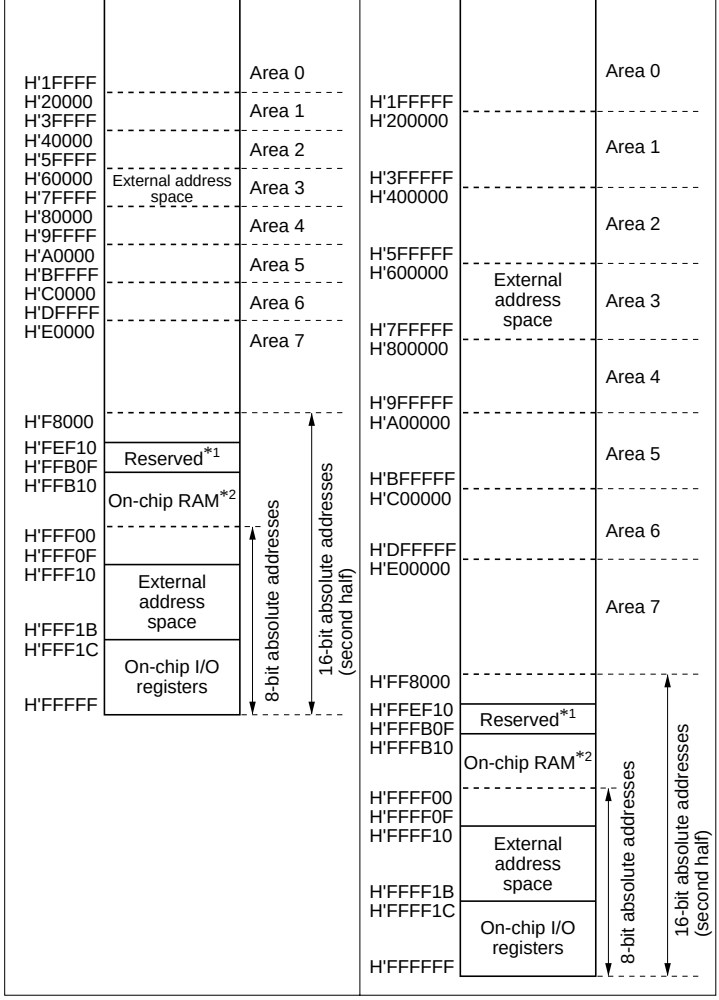
Notes: 1. Do not access the reserved area.
2. External addresses can be accessed by disabling on-chip RAM.

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



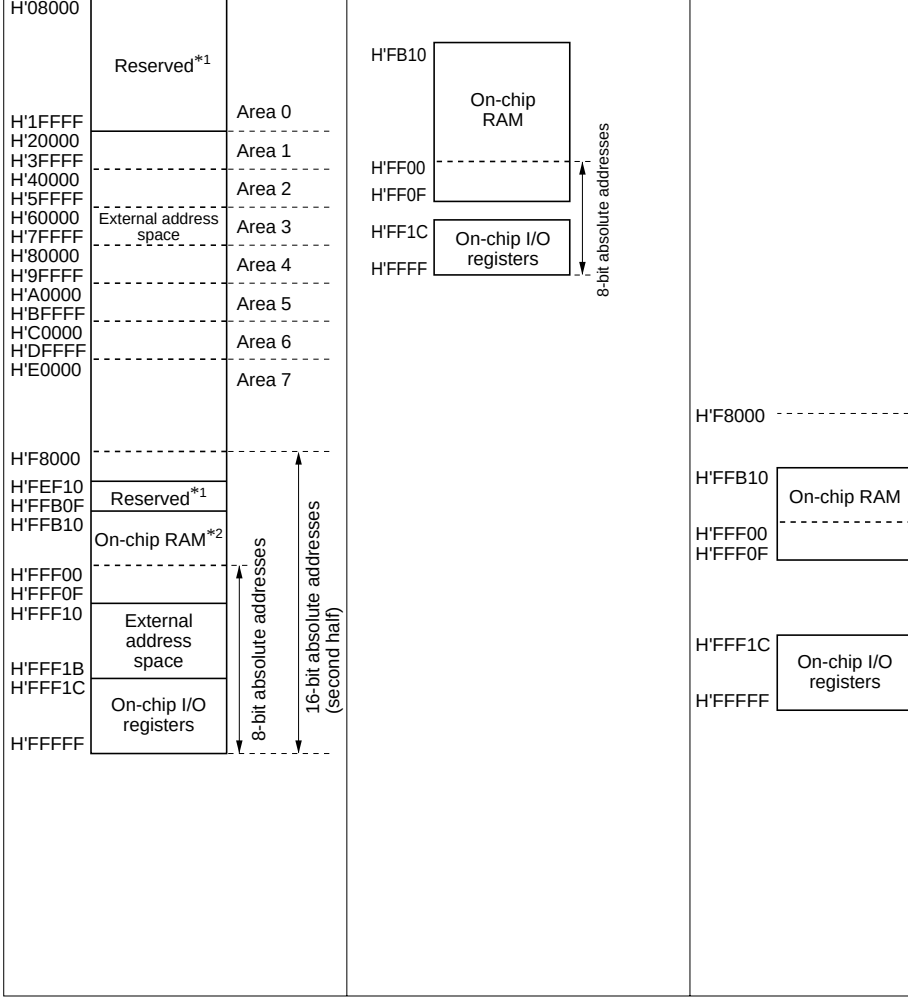
Notes: 1. Do not access the reserved area.
2. External addresses can be accessed by disabling on-chip RAM.

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



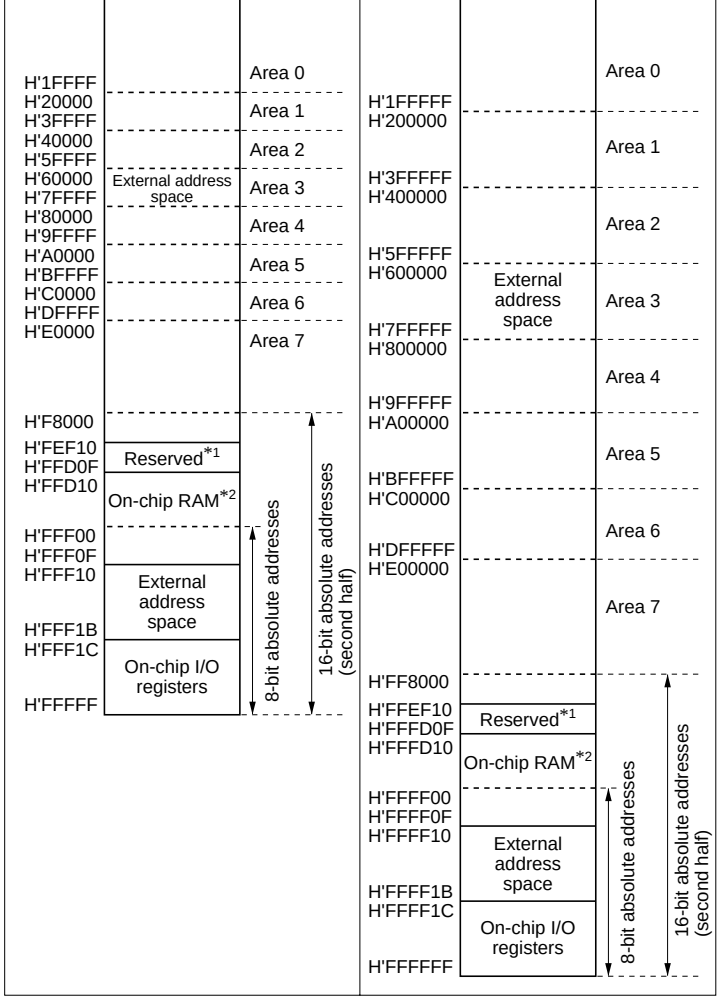
Notes: 1. Do not access the reserved area.
2. External addresses can be accessed by disabling on-chip RAM.

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



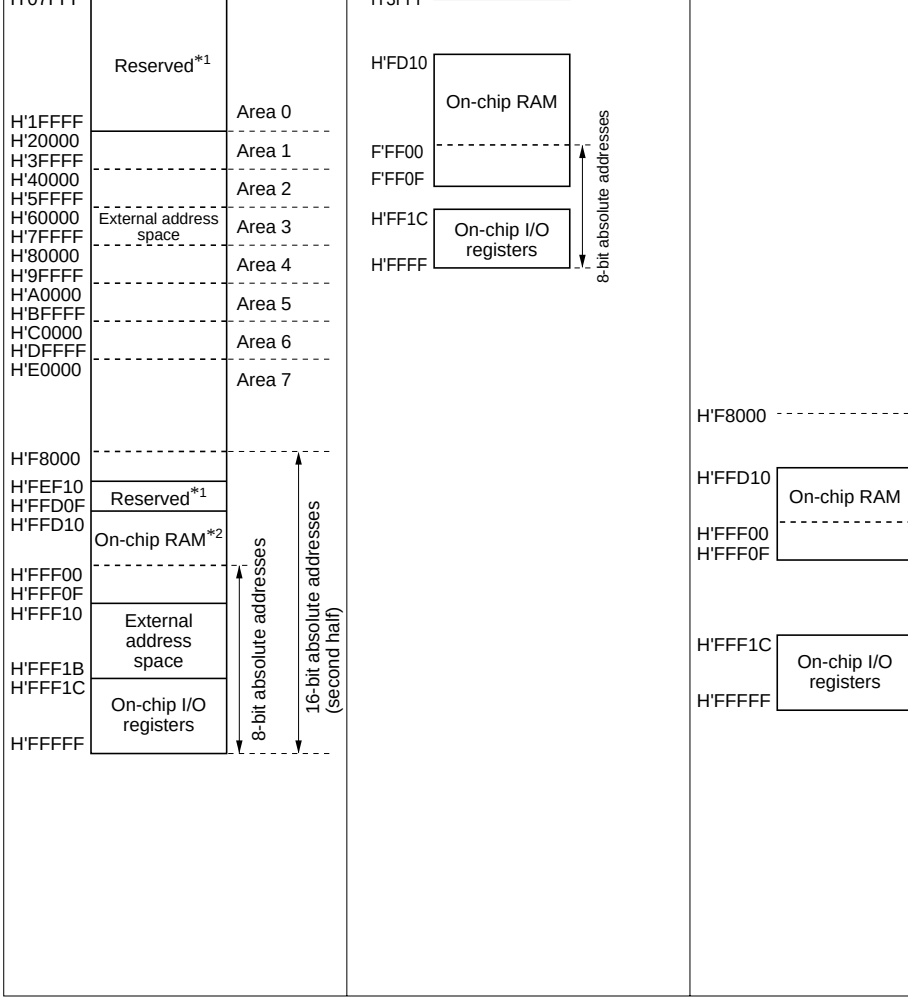
Notes: 1. Do not access the reserved area.
2. External addresses can be accessed by disabling on-chip RAM.

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)

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 to H'0000; or, same data as H'00000–H'0F H'10000–H'1F 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 or above as a address; or, w data as in H'0 H'0FFFF to H H'1FFFF in on 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 acce addresses ir shown unde or, write sam H'00000–H'0 H'10000–H'1 on-chip ROM
Memory indirect (@@aa:8)	—	—	No problem	—

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

Table 4.1 Exception Types and Priority

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

4.1.2 Exception Handling Operation

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

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

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

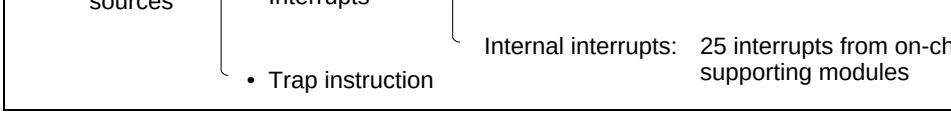


Figure 4.1 Exception Sources

		4	H'0008 to H'0009	H'0010 to
		5	H'000A to H'000B	H'0014 to
		6	H'000C to H'000D	H'0018 to
External interrupt (NMI)		7	H'000E to H'000F	H'001C to
Trap instruction (4 sources)		8	H'0010 to H'0011	H'0020 to
		9	H'0012 to H'0013	H'0024 to
		10	H'0014 to H'0015	H'0028 to
		11	H'0016 to H'0017	H'002C to
External interrupt	IRQ_0	12	H'0018 to H'0019	H'0030 to
	IRQ_1	13	H'001A to H'001B	H'0034 to
Reserved for system use		14	H'001C to H'001D	H'0038 to
		15	H'001E to H'001F	H'003C to
External interrupt	IRQ_4	16	H'0020 to H'0021	H'0040 to
	IRQ_5	17	H'0022 to H'0023	H'0044 to
Reserved for system use		18	H'0024 to H'0025	H'0048 to
		19	H'0026 to H'0027	H'004C to
Internal interrupts*[2]		20 to 60	H'0028 to H'0029 to H'0078 to H'0079	H'0050 to to H'00F0 to

Notes: 1. Lower 16 bits of the address.

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

The chip can also be reset by overflow of the watchdog timer. For details see section 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 chip during operation, hold the $\overline{\text{RES}}$ pin low for at least 10 system clock (ϕ) cycles. Wh the flash memory version, hold at "Low" level for a least 1usec. See appendix D.2, Pin 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 G starts reset exception handling as follows.

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

4.3 Interrupts

Interrupt exception handling can be requested by five external sources (NMI, IRQ_0, IRIRQ_5) and 25 internal sources in the on-chip supporting modules. Figure 4.3 classifies t sources and indicates the number of interrupts of each type.

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

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

For details on interrupts see section 5, Interrupt Controller.

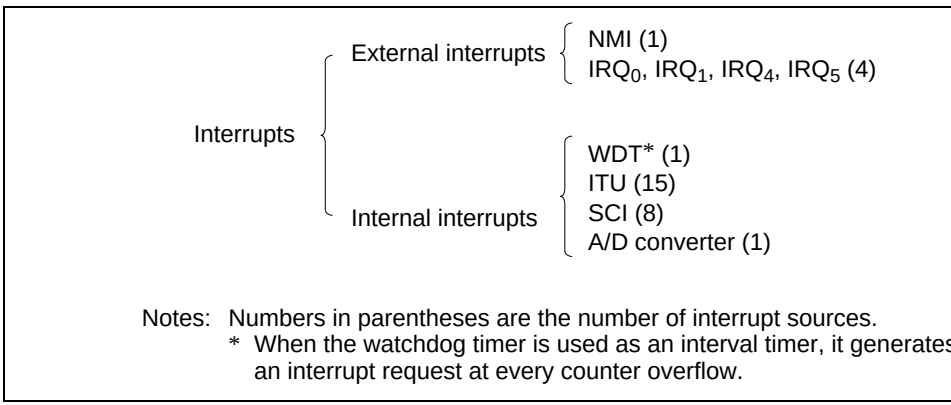
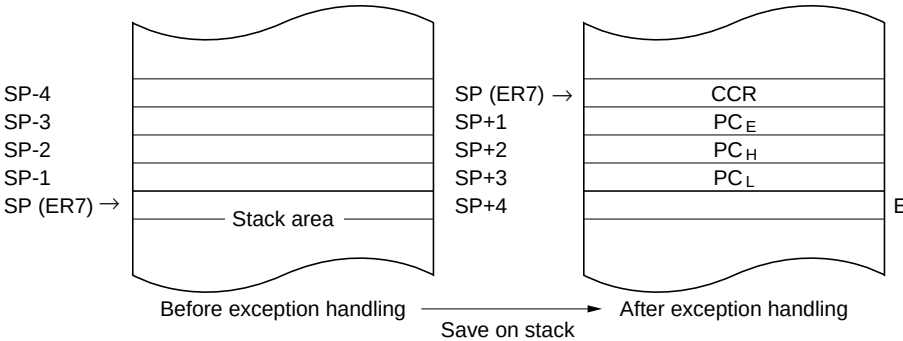


Figure 4.3 Interrupt Sources and Number of Interrupts

4.5 Stack Status after Exception Handling

Figure 4.4 shows the stack after completion of trap instruction exception handling and exception handling.



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

Notes: 1. PC indicates the address of the first instruction that will be executed after return.
2. Saving and restoring of registers must be conducted at even addresses in word-size or longword-size units.

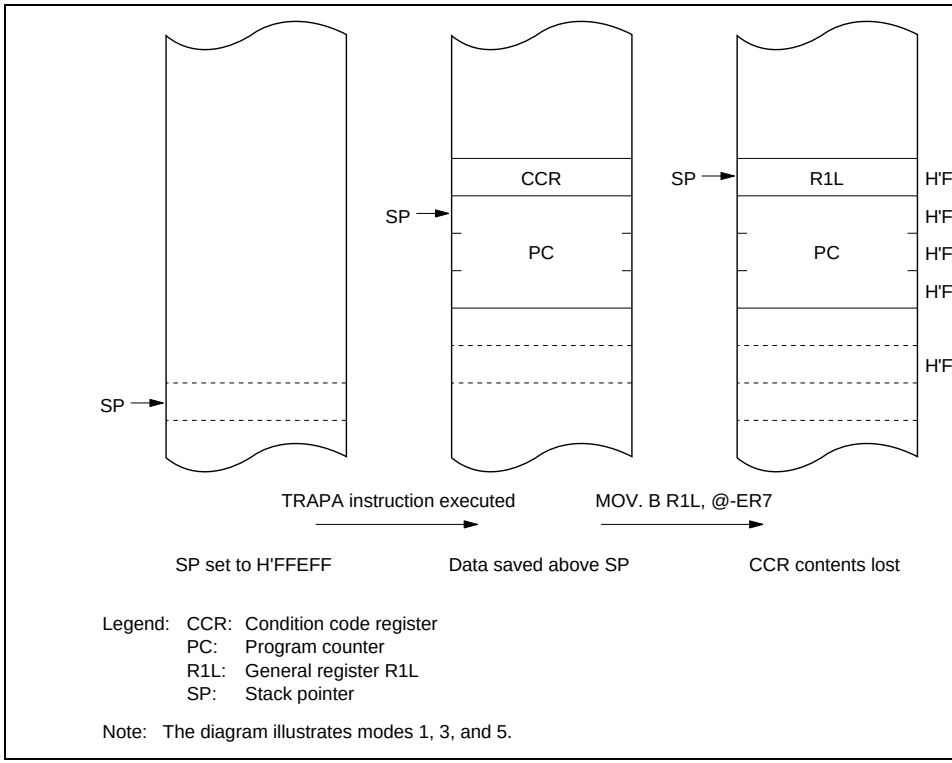
Figure 4.4 Stack after Completion of Exception Handling (Advanced M

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
happens when the SP value is odd.



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

Note: The diagram illustrates modes 1, 3, and 5.

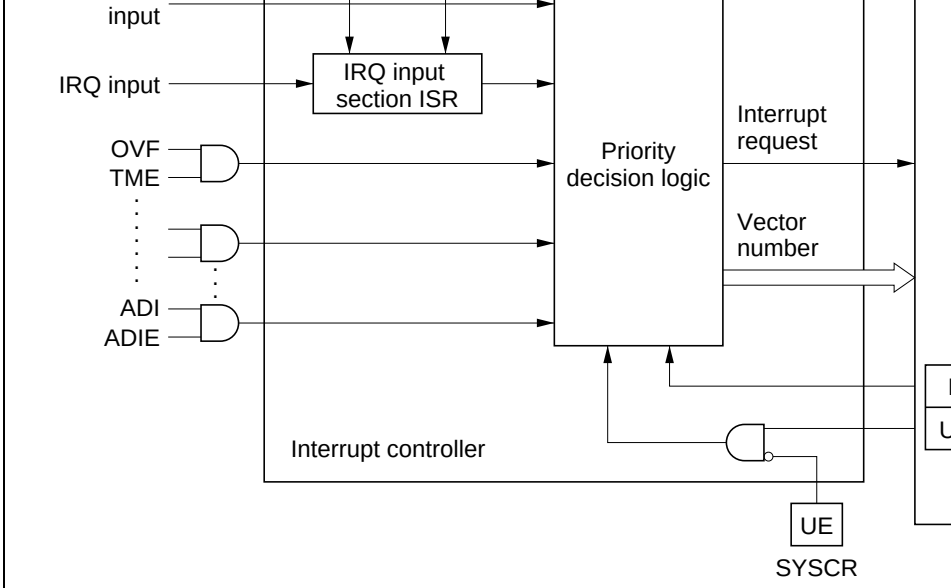
Figure 4.5 Operation when SP Value is Odd

- Interrupt priority registers (IPRs) for setting interrupt priorities

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

  All interrupts are independently vectored; the interrupt service routine does not ha
  identify the interrupt source.
- Five external interrupt pins

  NMI has the highest priority and is always accepted; either the rising or falling edg
  selected. For each of IRQ_0, IRQ_1, IRQ_4, and IRQ_5, sensing of the falling edge or lev
  can be selected independently.



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

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 e sensing selectable

5.1.4 Register Configuration

Table 5.2 lists the registers of the interrupt controller.

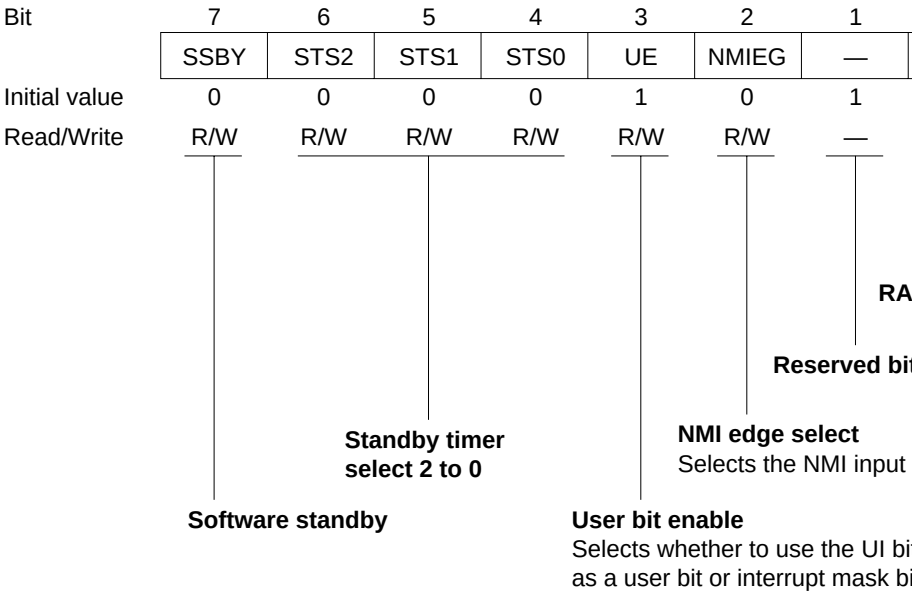
Table 5.2 Interrupt Controller Registers

Address*[1]	Name	Abbreviation	R/W	Ini
H'FFF2	System control register	SYSCR	R/W	H'0
H'FFF4	IRQ sense control register	ISCR	R/W	H'0
H'FFF5	IRQ enable register	IER	R/W	H'0
H'FFF6	IRQ status register	ISR	R/(W)*[2]	H'0
H'FFF8	Interrupt priority register A	IPRA	R/W	H'0
H'FFF9	Interrupt priority register B	IPRB	R/W	H'0

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

(SYSCR).

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



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

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

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

Pri
lev
Sel
pri
of I
cha
inte
req

Priority level
Selects the pr
of ITU channe
interrupt reque

Priority level A2
Selects the priority leve
ITU channel 0 interrupt

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

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

Bit6 IPRA6	Description
0	IRQ1 interrupt requests have priority level 0 (low priority)
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

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

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

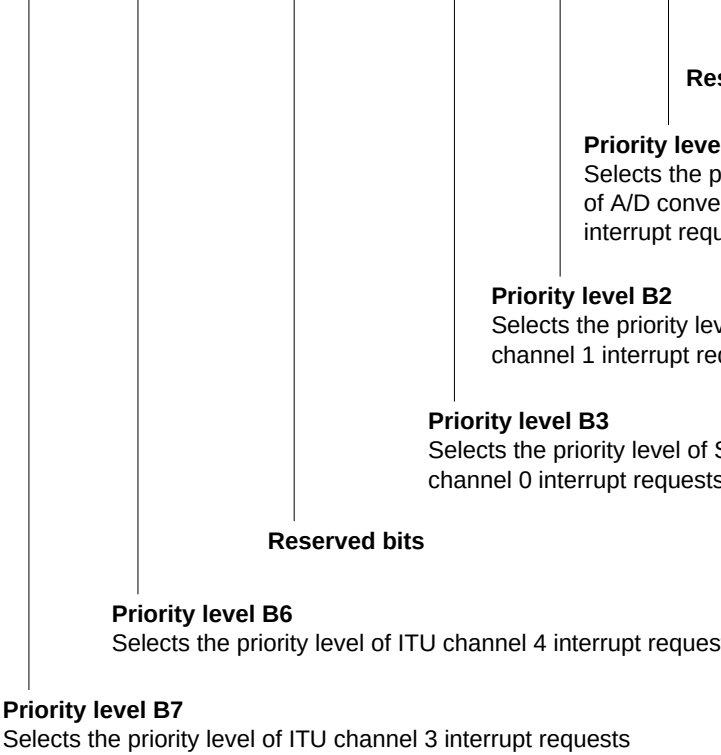
Bit 3—Priority Level A3 (IPRA3): Selects the priority level of WTD interrupt reque

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

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

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

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



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

Bit6 IPRB6	Description	
0	ITU channel 4 interrupt requests have priority level 0 (low priority)	(In
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 interrup

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

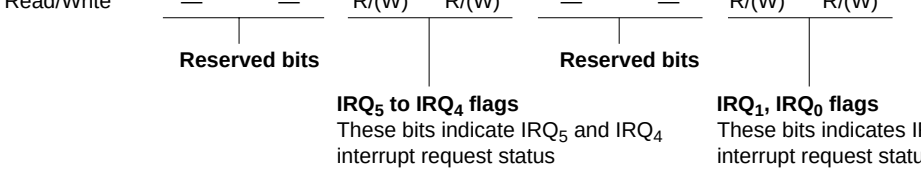
Bit 2—Priority Level B2 (IPRB2): Selects the priority level of SCI channel 1 interrup

Bit2 IPRB2	Description	
0	SCI channel 1 interrupt requests have priority level 0 (low priority)	(In
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 interrup

Bit1 IPRB1	Description	
0	A/D converter interrupt requests have priority level 0 (low priority)	(In
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.



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

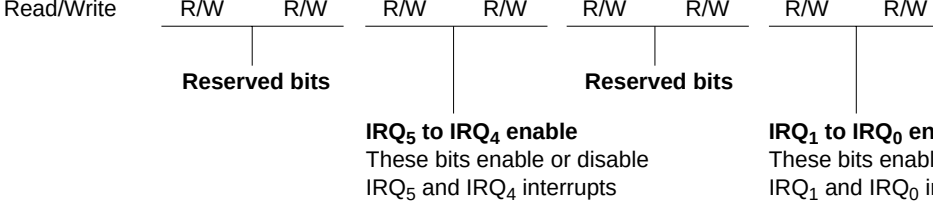
ISR is initialized to H'00 by a reset and in hardware standby mode.

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, an
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] (I • 0 is written in IRQnF after reading the IRQnF flag when IRQ • IRQnSC = 0, $\overline{IRQ}$n input is high, and interrupt exception har carried out. • IRQnSC = 1 and IRQn interrupt exception handling is carrie
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

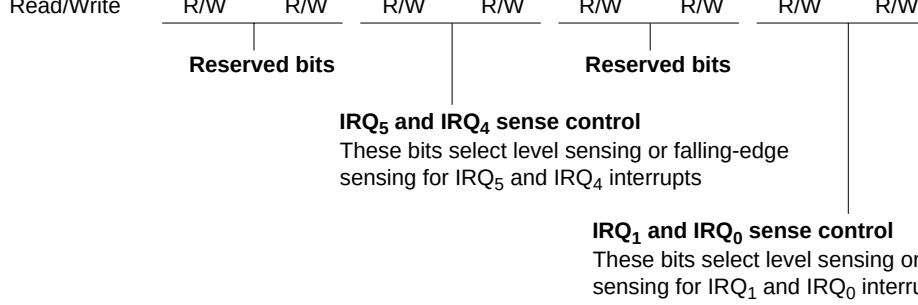


IER is initialized to H'00 by a reset and in hardware standby mode.

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, 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	(Ini
1	IRQ_5, IRQ_4, IRQ_1, IRQ_0 interrupts are enabled	



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 sele 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 IRQ0SC): These bits selects whether interrupts IRQ_5, IRQ_4, IRQ_1, IRQ_0 are requested 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 (I
1	Interrupts are requested by falling-edge input at $\overline{IRQ_5}$, $\overline{IRQ_4}$, $\overline{IRQ}$

IRQ1, can be used to exit software standby mode.

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

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

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

Figure 5.2 shows a block diagram of interrupts $\overline{IRQ_5}$, $\overline{IRQ_4}$, $\overline{IRQ_1}$, $\overline{IRQ_0}$.

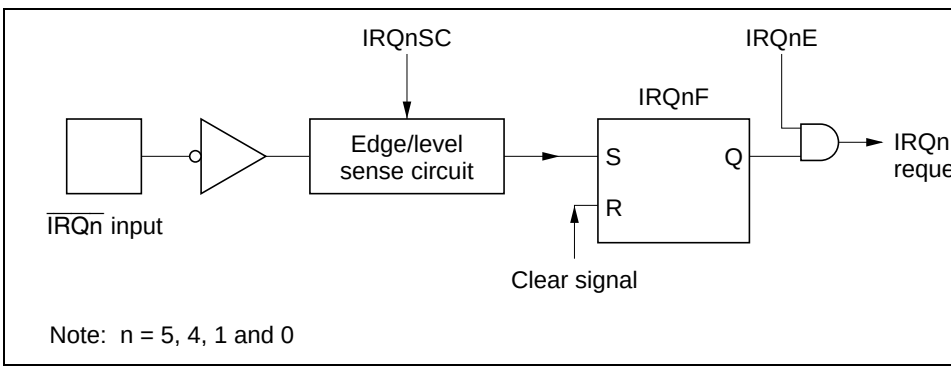
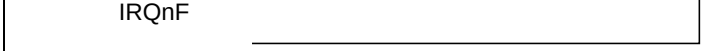


Figure 5.2 Block Diagram of Interrupts $\overline{IRQ_5}$, $\overline{IRQ_4}$, $\overline{IRQ_1}$, and $\overline{IRQ_0}$



Note: n = 5, 4, 1 and 0

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 interrupt
detected regardless of whether the corresponding pin is set for input or output. When u
for external interrupt input, clear its DDR bit to 0 and do not use the pin for SCI input

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
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 or
default priority order, smaller vector numbers have higher priority. The priority of inte
than NMI can be changed in IPRA and IPRB. The priority order after a reset is the def
shown in table 5.3.

		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	IPR
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	IPR
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	IPR
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	IPR
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	IPR
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	

IMIA4 (compare match/ input capture A4)	ITU channel 4	40	H'0050 to H'0051	H'00A0 to H'00A3	IP
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	IP
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	IP
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	IP

Note: * Lower 16 bits of the address.

NMI interrupts are always accepted except in the reset and hardware standby states. IR
and interrupts from the on-chip supporting modules have their own enable bits. Interrup
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 leve higher priority.
	1	—	No interrupts are accepted except NMI.
0	0	—	All interrupts are accepted. Interrupts with priority leve 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 modu
be masked by the I bit in the CPU's CCR. Interrupts are masked when the I bit is set to
unmasked when the I bit is cleared to 0. Interrupts with priority level 1 have higher prio
5.4 is a flowchart showing how interrupts are accepted when UE = 1.

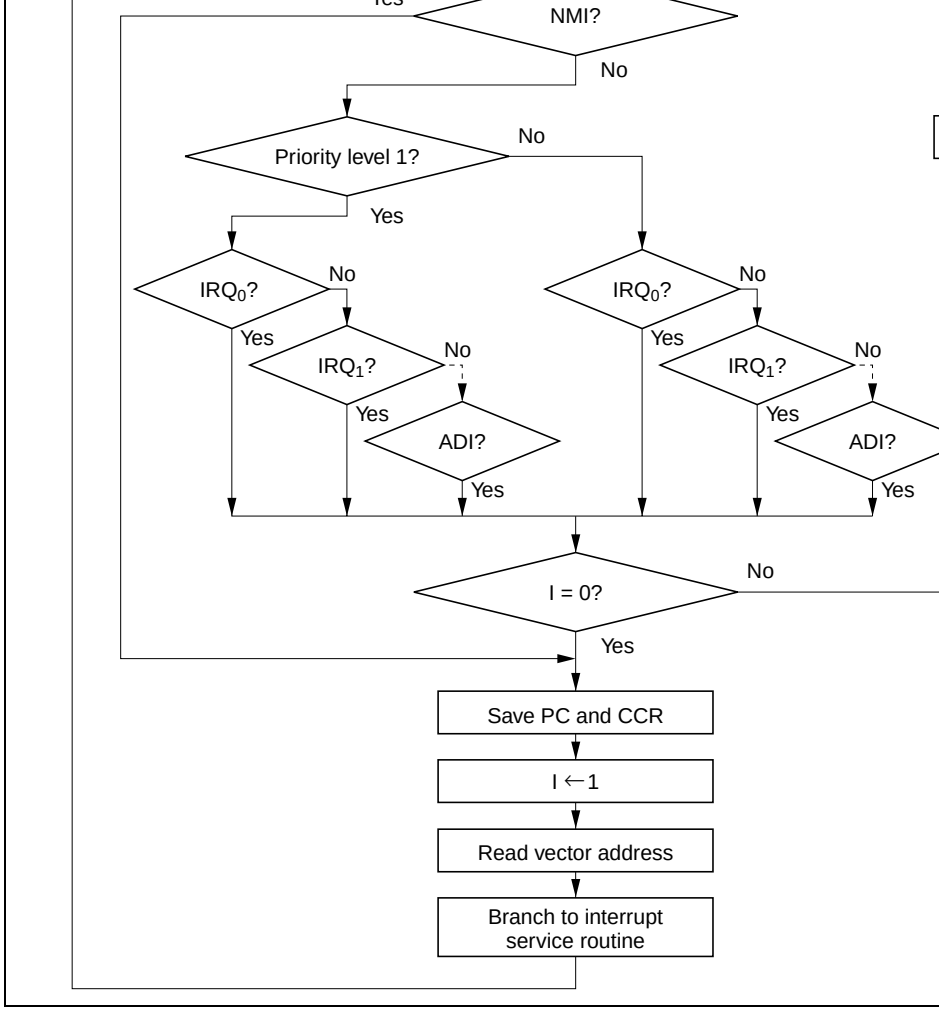


Figure 5.4 Process Up to Interrupt Acceptance when UE = 1

pending.

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

UE = 0

The I and UI bits in the CPU's CCR and the IPR bits enable three-level masking of IRQ 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 when the I bit is cleared to 0.
- Interrupt requests with priority level 1 are masked when the I and UI bits are both s are unmasked when either the I bit or the UI bit is cleared to 0.

  For example, if the interrupt enable bits of all interrupt requests are set to 1, IPRA i H'10, and IPRB is set to H'00 (giving IRQ_4 and IRQ_5 interrupt requests priority over 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.

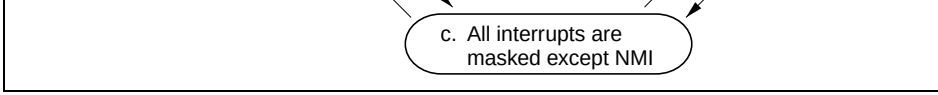


Figure 5.5 Interrupt Masking State Transitions (Example)

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

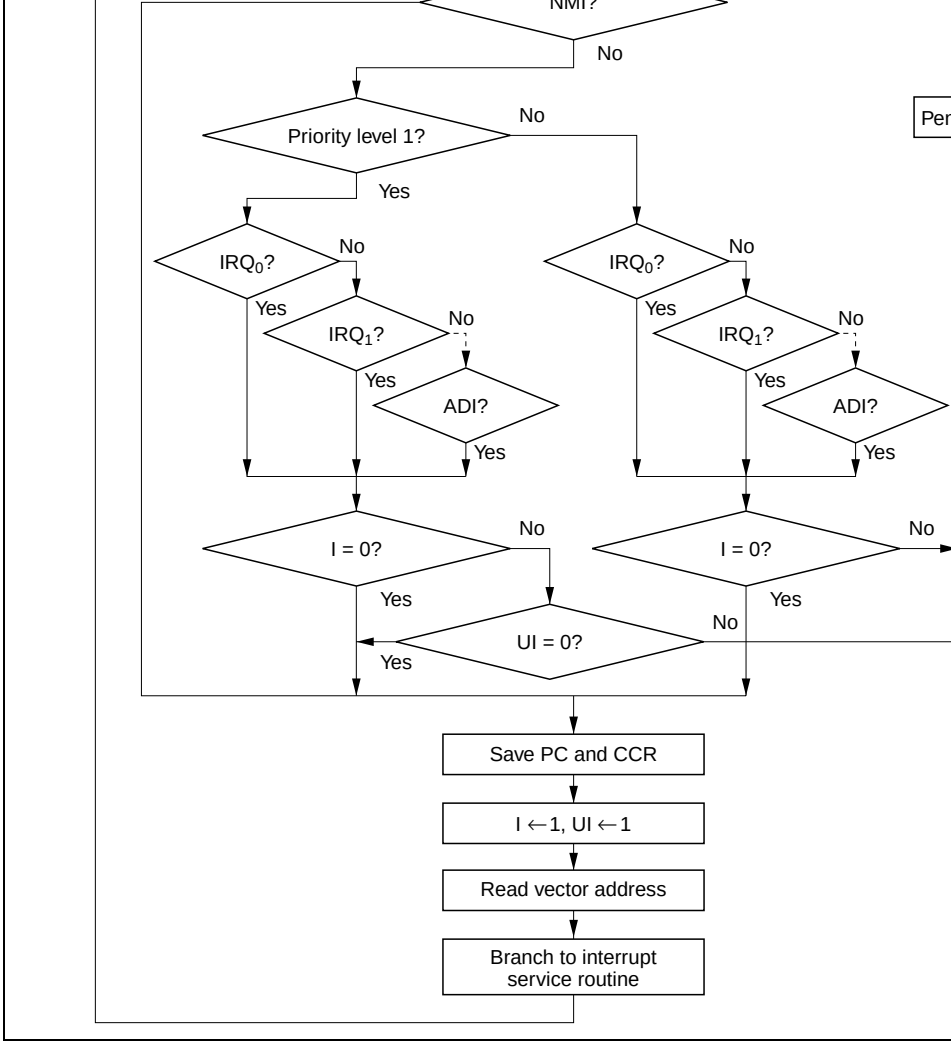


Figure 5.6 Process Up to Interrupt Acceptance when UE = 0

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

RENESAS

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

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

handling is carried out. If a higher-priority interrupt is also requested, however, interru
handling for the higher-priority interrupt is carried out, and the lower-priority interrup
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

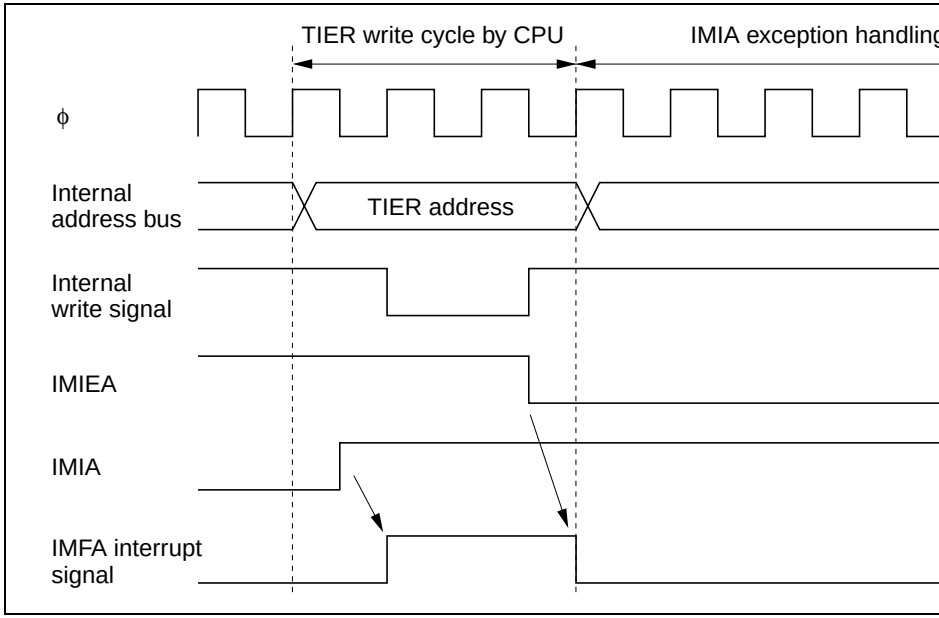


Figure 5.8 Contention between Interrupt and Interrupt-Disabling Instru

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

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

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

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

```
L1: 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 ha while set to 1. However, it is possible for the IRQnF flag to be cleared by mistake simp writing 0 to it, irrespective of whether it has been read while set to 1, with the result tha exception handling is not executed. This will occur when the following conditions are

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 clearing by hardware for the IRQbF flag.

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

when the ISR write in generation condition (2) is performed the IRQbF flag will be cl
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 t
once between generation conditions (1) and (2).

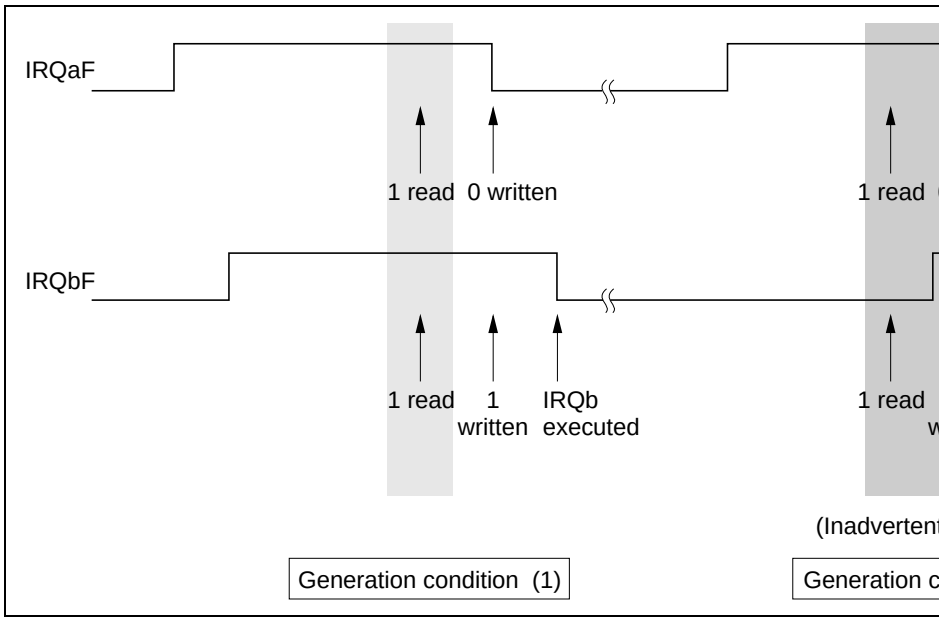


Figure 5.9 IRQnF Flag when Interrupt Exception Handling is not Exec

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

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

Example: When b = 1

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

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
  - Zero to three wait states can be inserted automatically.

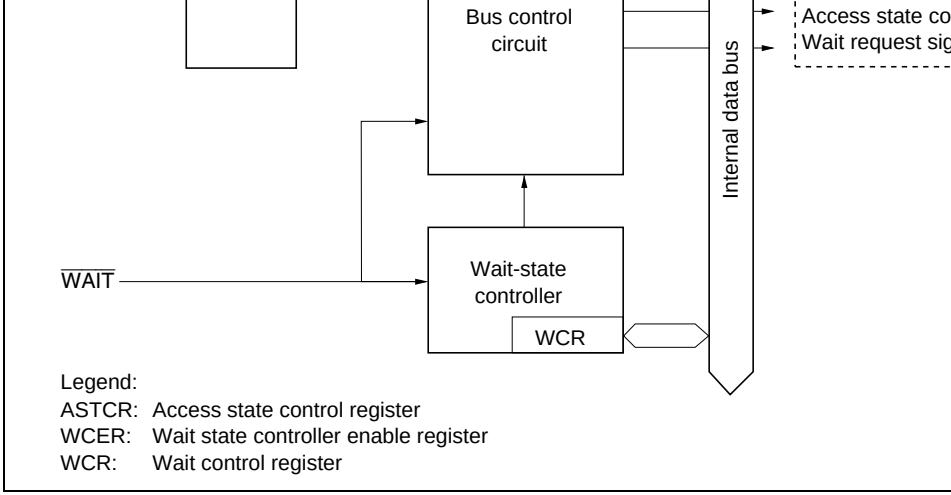


Figure 6.1 Block Diagram of Bus Controller

Read	$\overline{RD}$	Output	Strobe signal indicating reading from address space
Write	$\overline{WR}$	Output	Strobe signal indicating writing to the address space, with valid data on the bus(D7 to D0)
Wait	$\overline{WAIT}$	Input	Wait request signal for access to external 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	Init
H'FFED	Access state control register	ASTCR	R/W	H'F
H'FFEE	Wait control register	WCR	R/W	H'F
H'FFEF	Wait state controller enable register	WCER	R/W	H'F
H'FFF3	Address control register	ADRCR	R/W	H'F

Note: * Lower 16 bits of the address.

	AST7	AST6	AST5	AST4	AST3	AST2	AST1
Initial value	1	1	1	1	1	1	1
Read/Write	R/W	R/W	R/W	R/W	R/W	R/W	R/W

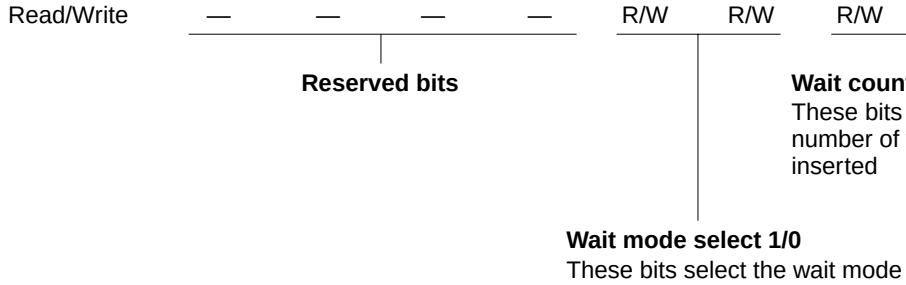
Bits selecting number of states for access to each area

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

Bits 7 to 0—Area 7 to 0 Access State Control (AST7 to AST0): These bits select wh
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 (In

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



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

Bit3 WMS1	Bit2 WMS0	Description
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

1	3 states inserted	(In

6.2.3 Wait State Controller Enable Register (WCER)

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

Bit	7	6	5	4	3	2	1	
	WCE7	WCE6	WCE5	WCE4	WCE3	WCE2	WCE1	
Initial value	1	1	1	1	1	1	1	
Read/Write	R/W	R/W	R/W	R/W	R/W	R/W	R/W	

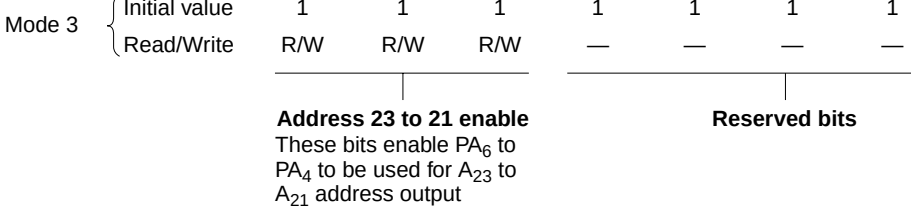
Wait state controller enable 7 to 0
These bits enable or disable wait-state control

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

Bits 7 to 0—Wait-State Controller Enable 7 to 0 (WCE7 to WCE0): These bits ena 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	(Ini

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



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

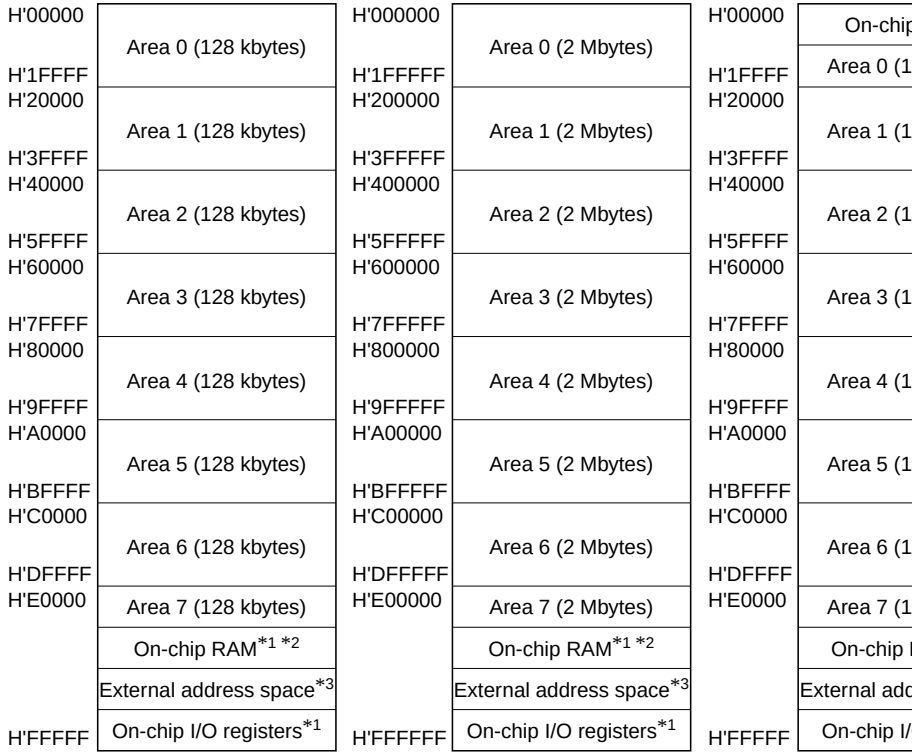
Bit 7—Address 23 Enable ($A_{23}E$): Enables PA4 to be used as the A_{23} address output p 0 in this bit enables A_{23} address output from PA4. In modes other than 3 this bit canno modified and PA4 has its ordinary input/output functions

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 (

Bit 6—Address 22 Enable ($A_{22}E$): Enables PA_5 to be used as the A_{22} address output p in this bit enables A_{22} address output from PA_5. In modes other than 3 this bit cannot b 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 (

Bits 4 to 0—Reserved



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 m
on-chip R
(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 fi
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)

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

Note: n = 0 to 7

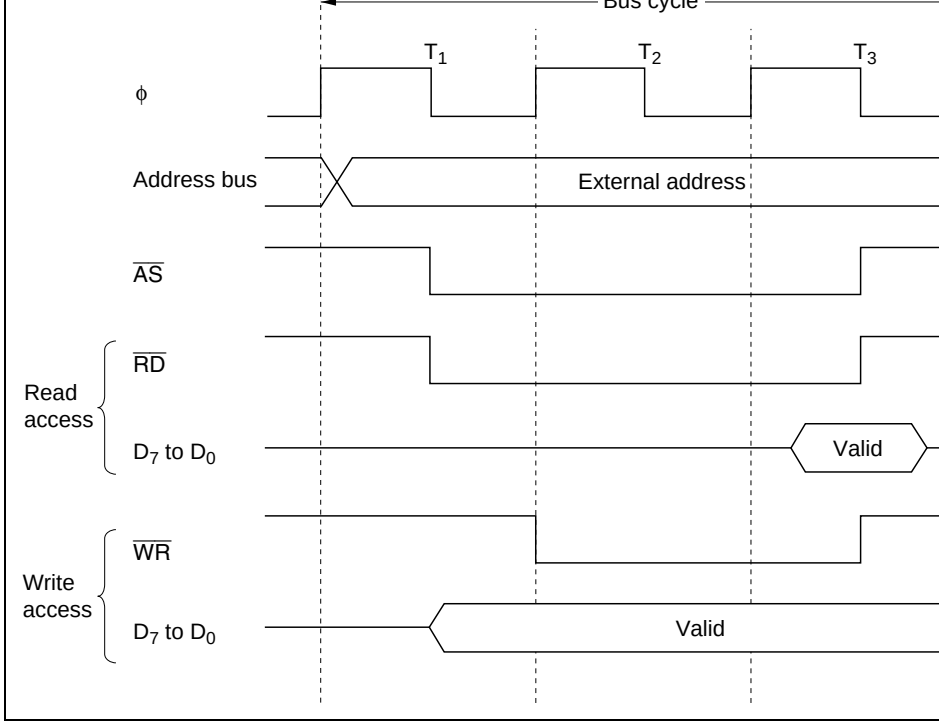


Figure 6.3 Bus Control Signal Timing for 8-Bit, Three-State-Access Ar

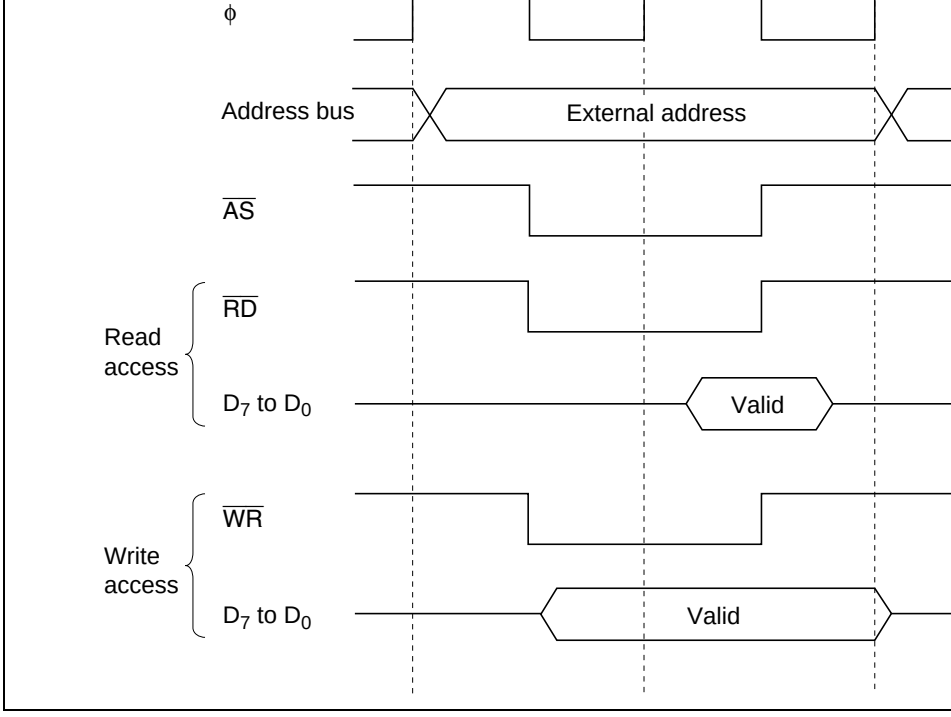


Figure 6.4 Bus Control Signal Timing for 8-Bit, Two-State-Access Area

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

Note: n = 0 to 7

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

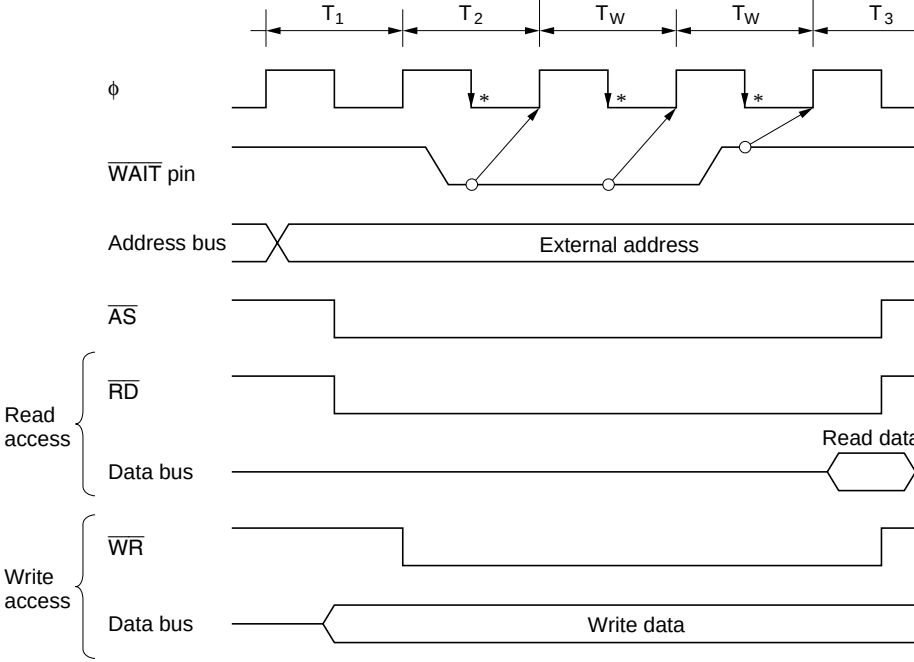


Figure 6.5 Pin Wait Mode 0

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 ac
state is inserted by $\overline{\text{WAIT}}$ input.

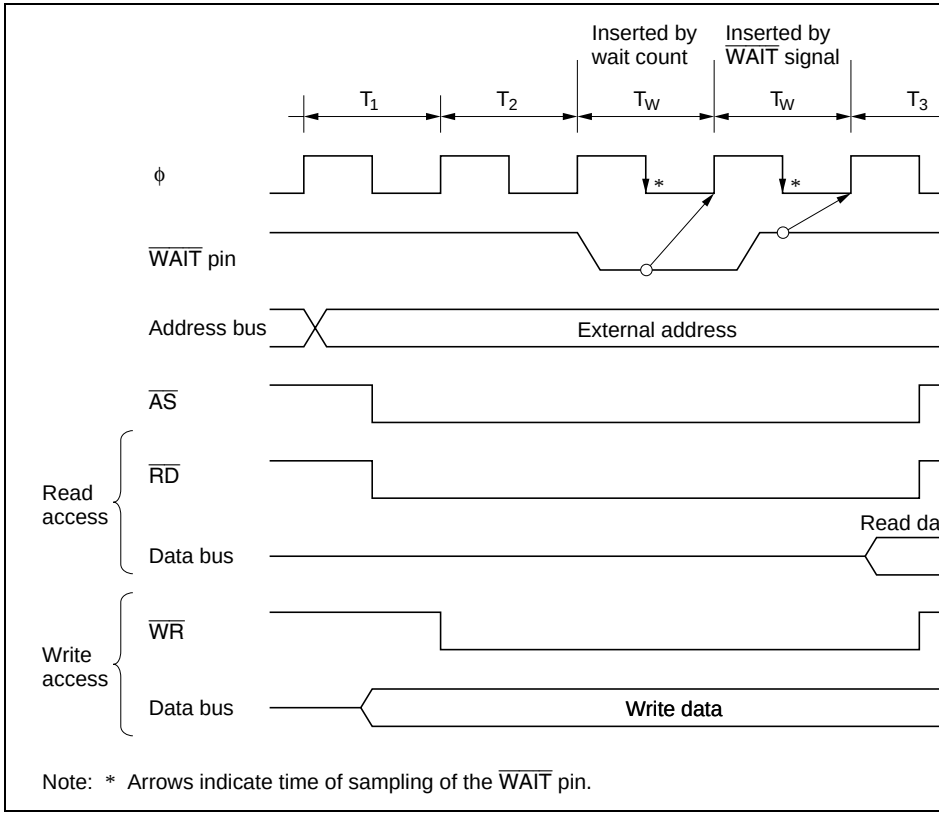


Figure 6.6 Pin Wait Mode 1

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

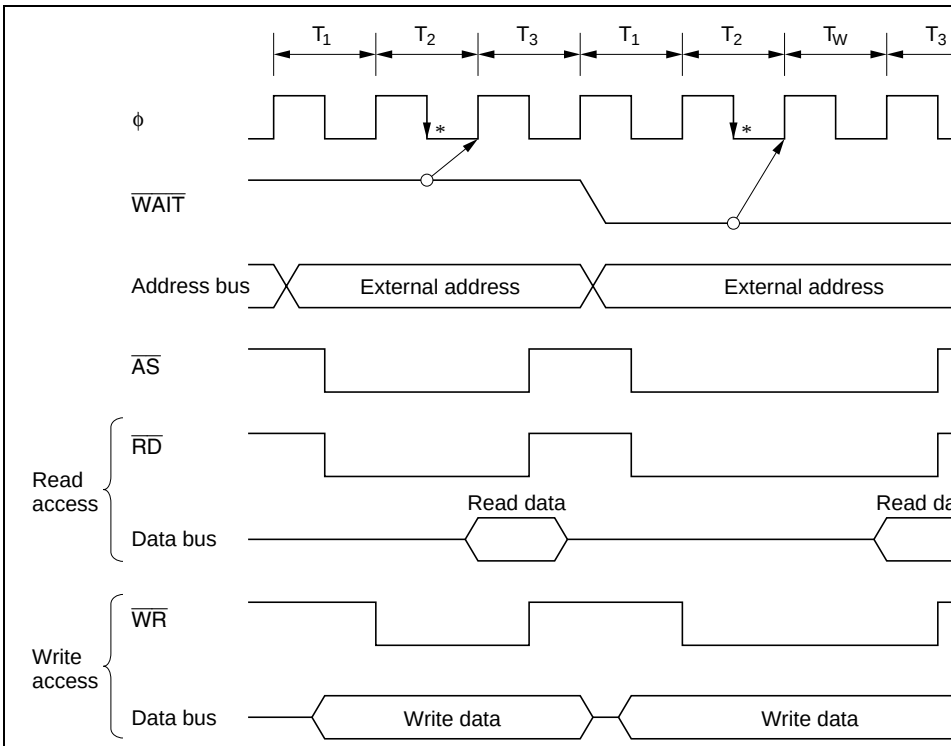


Figure 6.7 Pin Auto-Wait Mode

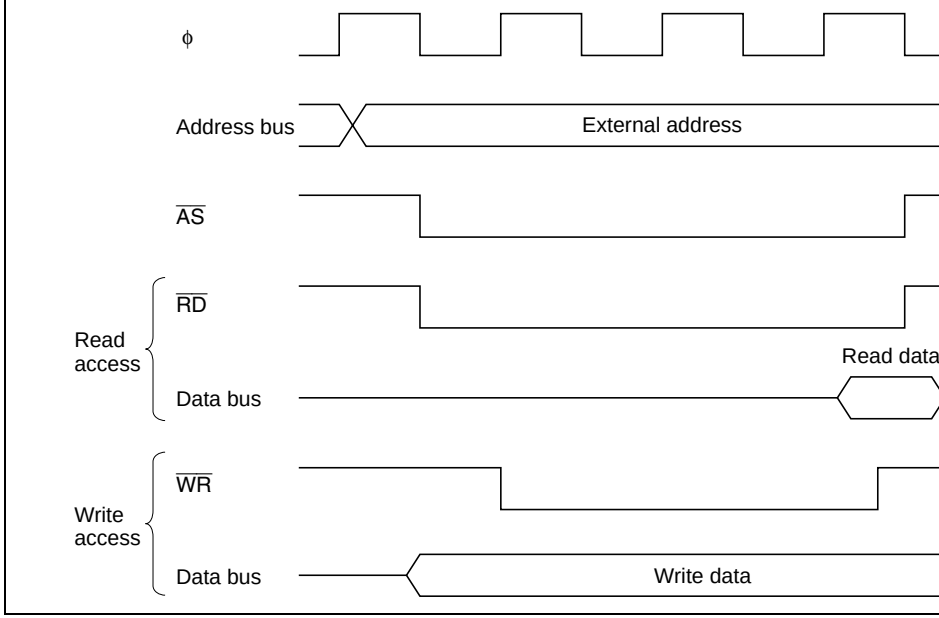
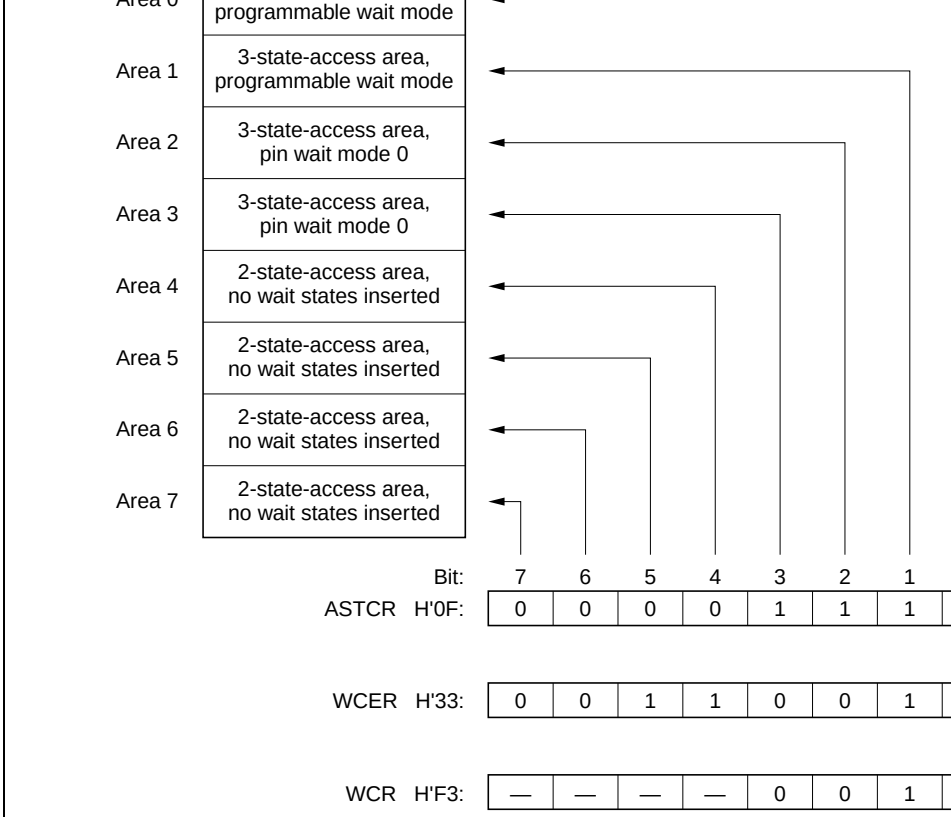


Figure 6.8 Programmable Wait Mode



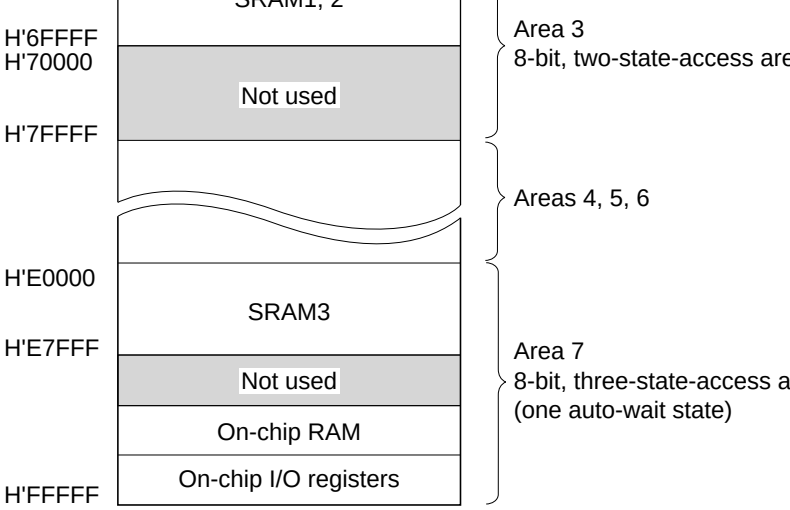
Note: Wait states cannot be inserted in areas designated for two-state access by ASTCR.

Figure 6.9 Wait Mode Settings (Example)

8-bit bus.

Two 32-kword × 8-bit SRAM devices (SRAM1 and SRAM2) are connected to area 3. 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 accesse bus, using three-state access with an additional wait state inserted in pin auto-wait mo



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

Figure 6.10 Memory Map (H8/3039 Mode 5)

state access.

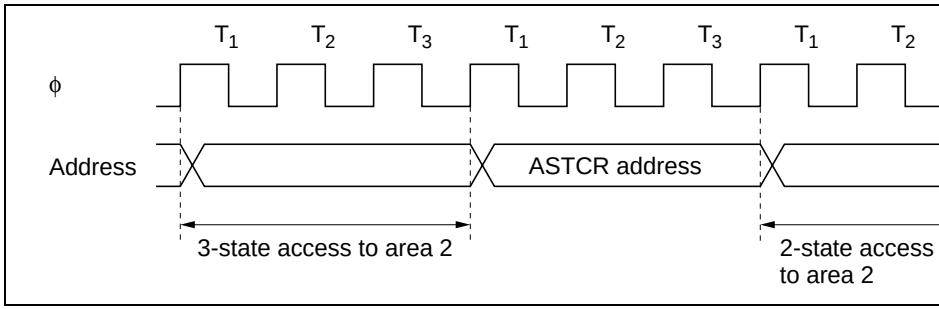


Figure 6.11 ASTCR Write Timing

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
on-chip ROM access cycle for H8/3039 Group can be emulated using the evaluation c
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
onto this LSI. For detailed features of the ABWCR, see the H8/3048 Group
H8/3048F-ZTAT™ Hardware Manual.

Each port has a data direction register (DDR) for selecting input or output, and a data
(DR) for storing output data. In addition to these registers, ports 2, and 5 have an inpu
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.
and B can drive one TTL load and a 30-pF capacitive load. Ports 1 to 3 and ports 5, 6,
B can drive a Darlington pair. Ports 1, 2, 5, and B can drive LEDs (with 10-mA curren
$P8_1$, $P8_0$, 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.

				address 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 in
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 in
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 in
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 in
		$P6_0$/$\overline{WAIT}$	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 inp		
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 I and gener output
		$P8_0$/$\overline{IRQ}_0$	$\overline{IRQ}_0$ input and 1-bit generic input/output		

	inputs		timing pattern controller (TPC), input or output ($TIOCB_2$) for 16-bit integrated timer unit (ITU), 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), IT output ($TIOCA_2$, $TIOCB_1$, TI 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, TC TCLKB, TCLKA, $TIOCB_0$, $TIOCA_0$), and generic input/outp		

		$TIOCA_4$, PB_1/TP_9/ $TIOCB_3$, PB_0/TP_8/ $TIOCA_3$	

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

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

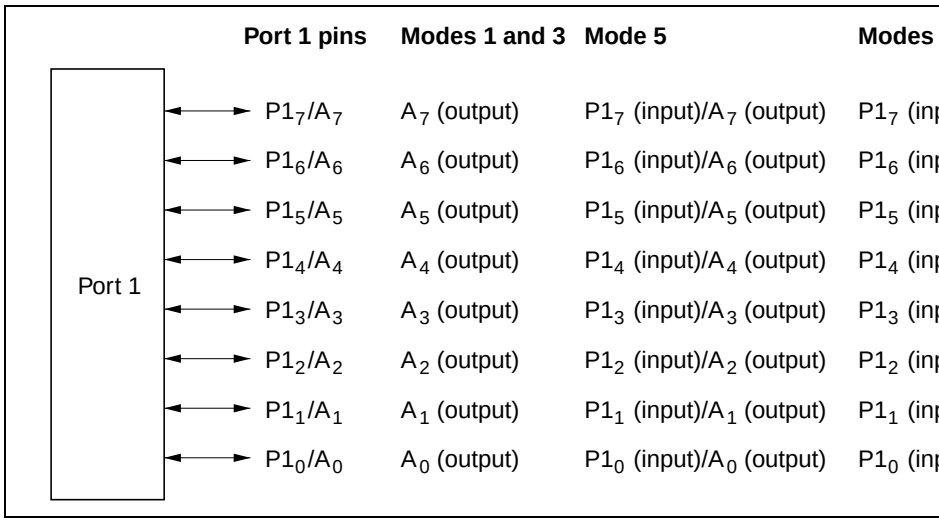


Figure 7.1 Port 1 Pin Configuration

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

Note: * Lower 16 bits of the address.

Port 1 Data Direction Register (P1DDR)

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

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

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

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

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 returned directly, regardless of the actual state of the pin. When a bit in P1DDR is clea 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 stan retains its previous setting.

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

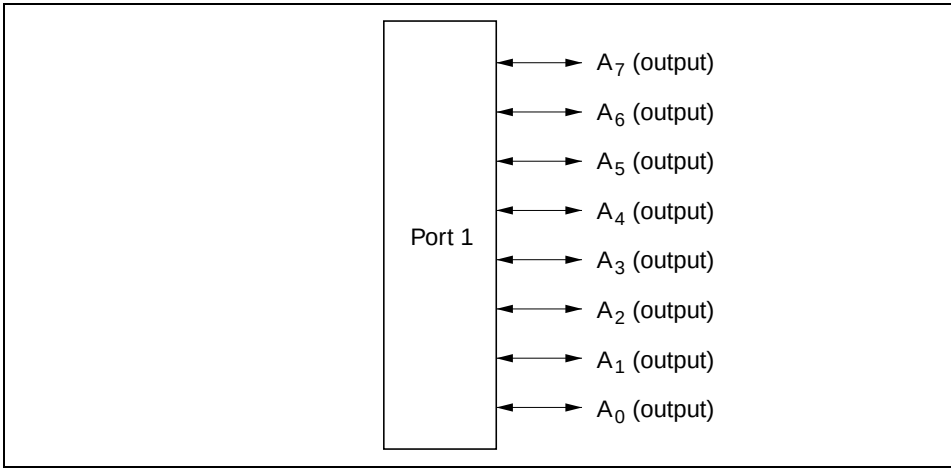


Figure 7.2 Pin Functions in Modes 1 and 3 (Port 1)

Mode 5

Address output or generic input can be selected for each pin in port 1. A pin becomes a
output pin if the corresponding P1DDR bit is set to 1, and a generic input pin if this bit
to 0. Following a reset, all pins are input pins. To use a pin for address output, its P1DD
be set to 1. Figure 7.3 shows the pin functions in mode 5.

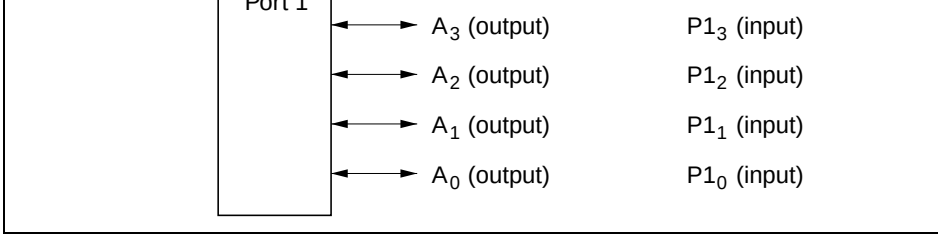


Figure 7.3 Pin Functions in Mode 5 (Port 1)

Modes 6 and 7 (Single-Chip Mode)

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

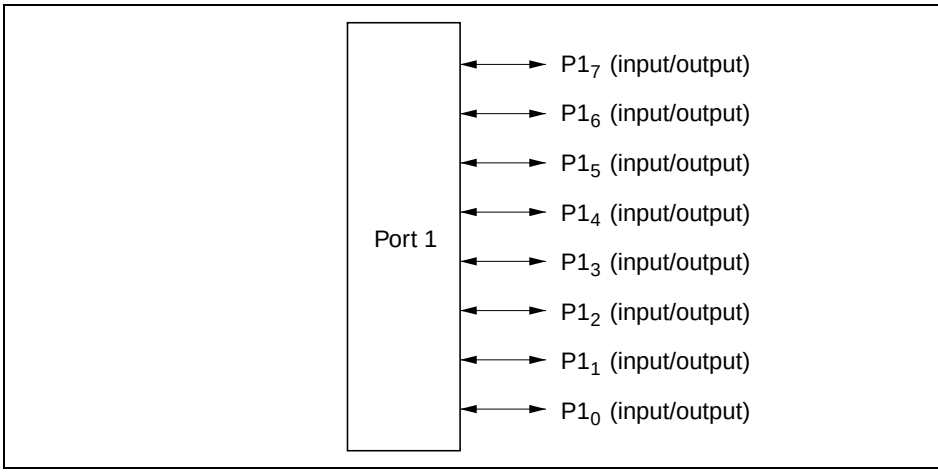


Figure 7.4 Pin Functions in Modes 6 and 7 (Port 1)

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

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

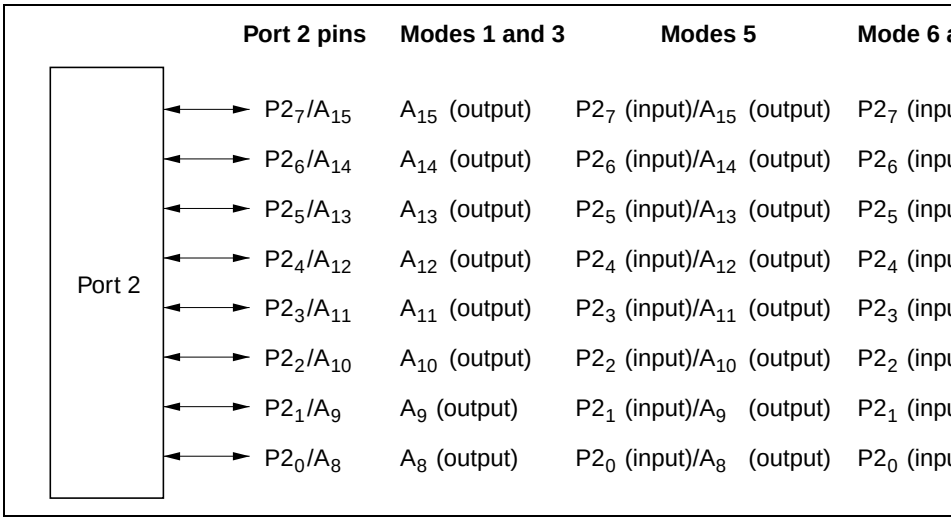


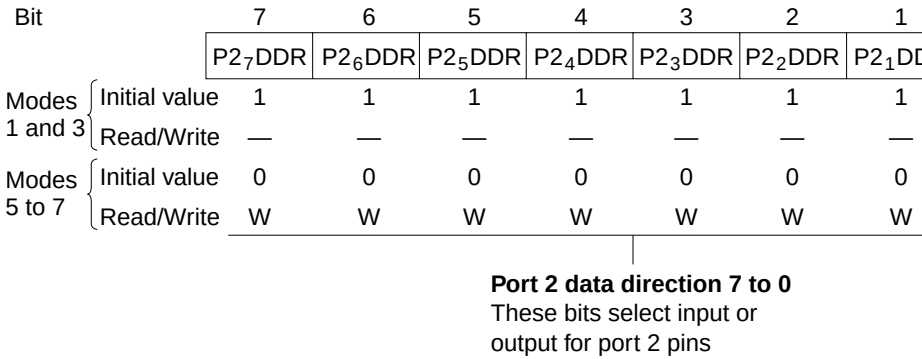
Figure 7.5 Port 2 Pin Configuration

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

Note: * Lower 16 bits of the address.

Port 2 Data Direction Register (P2DDR)

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



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

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 is returned directly, regardless of the actual state of the pin. When a bit in P2DDR is clear 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 stand 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 transis 2.

Bit	7	6	5	4	3	2	1
	$P2_7PCR$	$P2_6PCR$	$P2_5PCR$	$P2_4PCR$	$P2_3PCR$	$P2_2PCR$	$P2_1PCR$
Initial value	0	0	0	0	0	0	0
Read/Write	R/W	R/W	R/W	R/W	R/W	R/W	R/W

Port 2 input pull-up 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 corre bit from $P2_7PCR$ to $P2_0PCR$ 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 stan retains its previous setting.

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

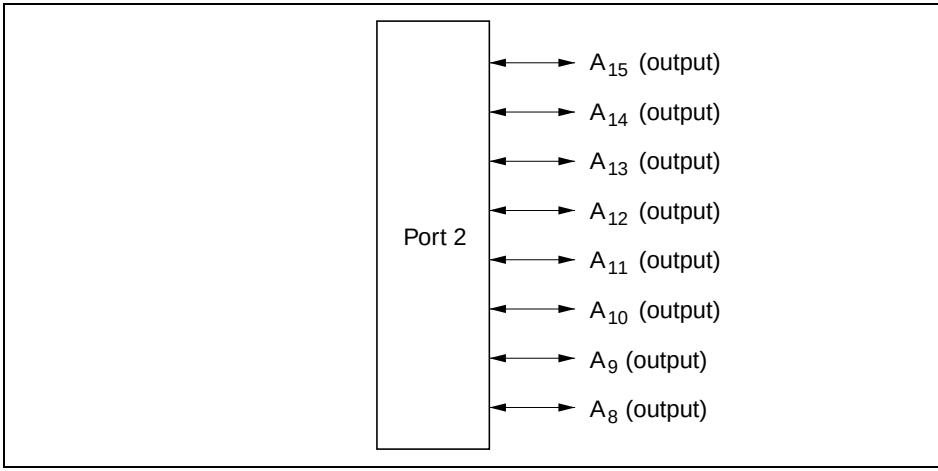


Figure 7.6 Pin Functions in Modes 1 and 3 (Port 2)

Mode 5

Address output or generic input can be selected for each pin in port 2. A pin becomes
output pin if the corresponding P2DDR bit is set to 1, and a generic input pin if this bi
to 0. Following a reset, all pins are input pins. To use a pin for address output, its P2D
be set to 1. Figure 7.7 shows the pin functions in modes 5.

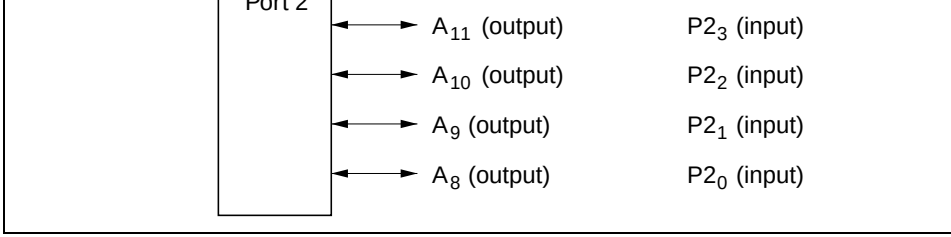


Figure 7.7 Pin Functions in Modes 1 and 3 (Port 2)

Modes 6 and 7

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

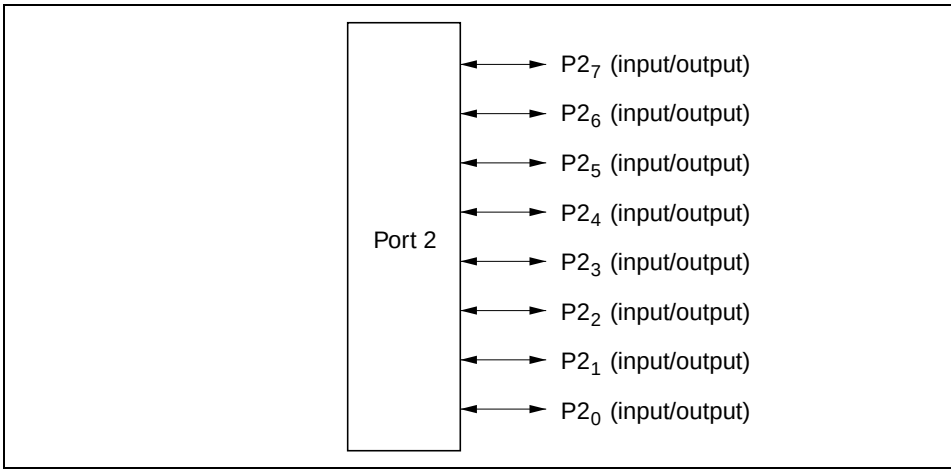


Figure 7.8 Pin Functions in Modes 6 and 7 (Port 2)

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 Mo
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 i

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

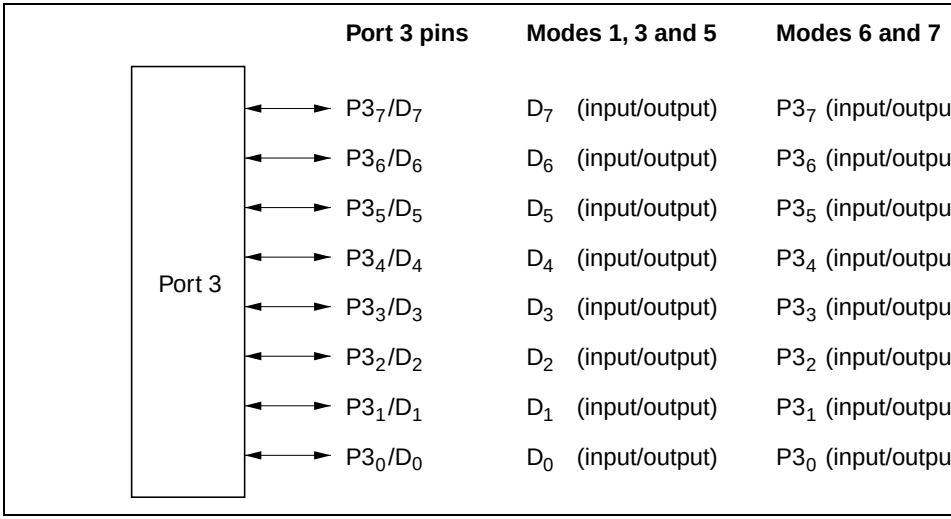


Figure 7.9 Port 3 Pin Configuration

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 V
H'FFC4	Port 3 data direction register	P3DDR	W	H'00
H'FFC6	Port 3 data register	P3DR	R/W	H'00

Note: * Lower 16 bits of the address.

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

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

Port 3 Data Register (P3DR)

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

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

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

When a bit in P3DDR is set to 1, if port 3 is read the value of the corresponding P3DR returned directly, regardless of the actual state of the pin. When a bit in P3DDR is clea 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 stan retains its previous setting.

functions in modes 1, 3 and 5.

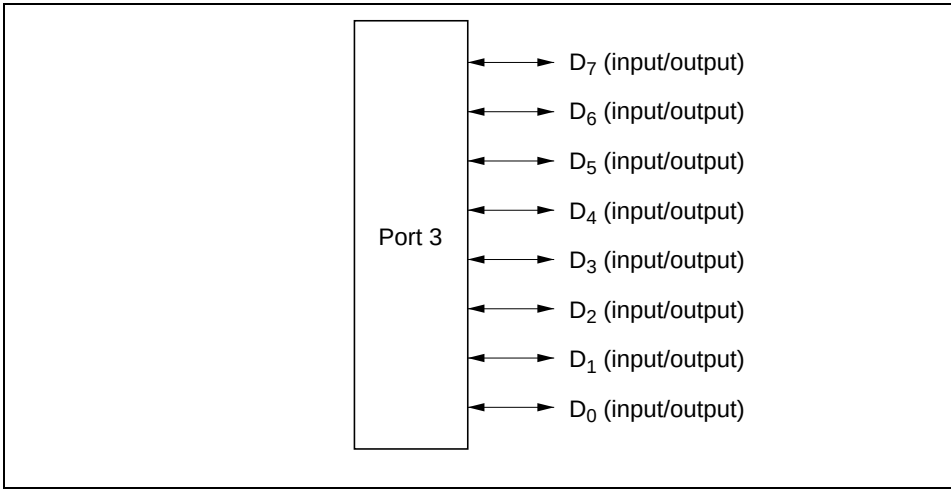


Figure 7.10 Pin Functions in Modes 1, 3 and 5 (Port 3)

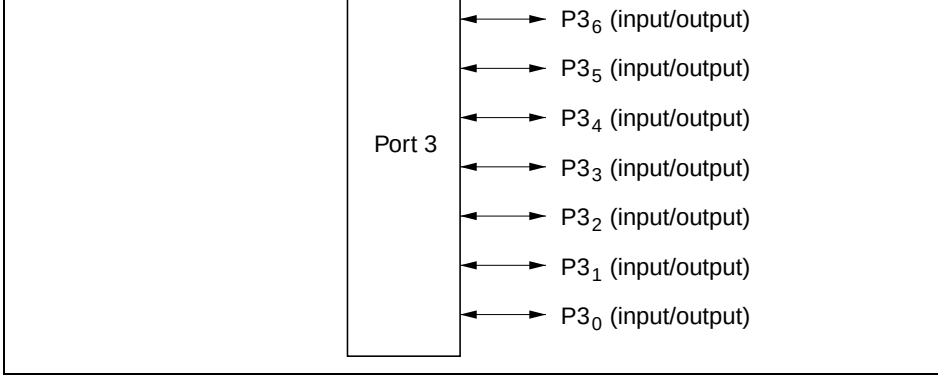
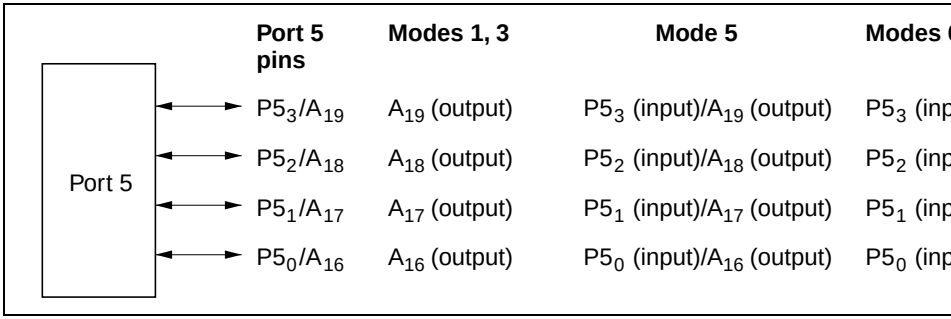


Figure 7.11 Pin Functions in Modes 6 and 7 (Port 3)

pins (A19 to A16). In modes 5 (expanded modes with on-chip ROM enabled), settings
5 data direction register (P5DDR) designate pins for address bus output (A_{19} to A_{16}) or g
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
90-pF capacitive load. They can also drive an LED or a Darlington transistor pair.



Port 5 pins	Modes 1, 3	Mode 5	Modes
$P5_3/A_{19}$	A_{19} (output)	$P5_3$ (input)/A_{19} (output)	$P5_3$ (inp
$P5_2/A_{18}$	A_{18} (output)	$P5_2$ (input)/A_{18} (output)	$P5_2$ (inp
$P5_1/A_{17}$	A_{17} (output)	$P5_1$ (input)/A_{17} (output)	$P5_1$ (inp
$P5_0/A_{16}$	A_{16} (output)	$P5_0$ (input)/A_{16} (output)	$P5_0$ (inp

Figure 7.12 Port 5 Pin Configuration

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

Note: * Lower 16 bits of the address.

Port 5 Data Direction Register (P5DDR)

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

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

Reserved bits

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

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

Reserved bits

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

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

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

Reserved bits

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

When a P5DDR bit is cleared to 0 (selecting generic input) in modes 5 to 7, if the corre bit from $P5_3PCR$ to $P5_0PCR$ 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 stan it retains its previous setting.

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

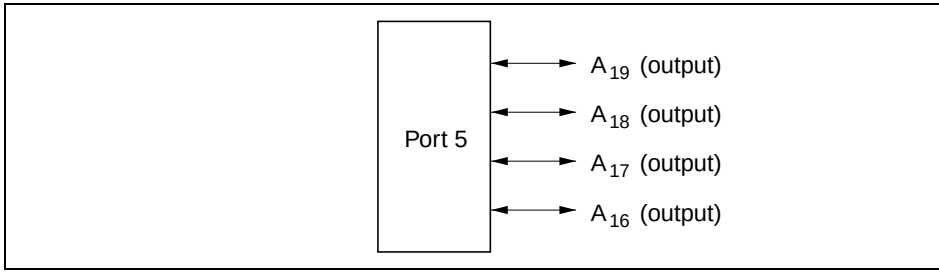


Figure 7.13 Pin Functions in Modes 1 and 3 (Port 5)

Mode 5

Address output or generic input can be selected for each pin in port 5. A pin becomes
output pin if the corresponding P5DDR bit is set to 1, and a generic input pin if this bi
to 0. Following a reset, all pins are input pins. To use a pin for address output, its P5D
set to 1. Figure 7.14 shows the pin functions in mode 5.

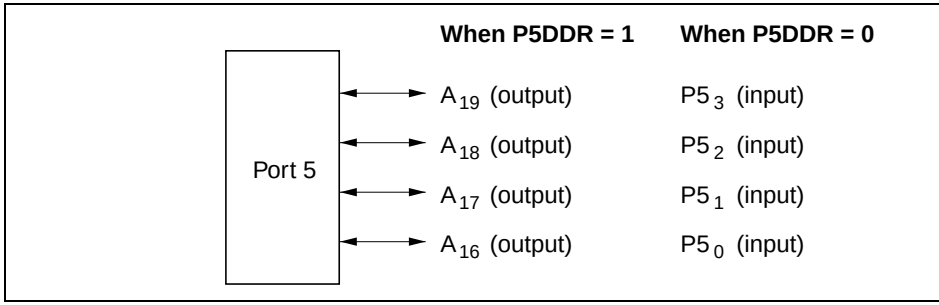


Figure 7.14 Pin Functions in Mode 5 (Port 5)

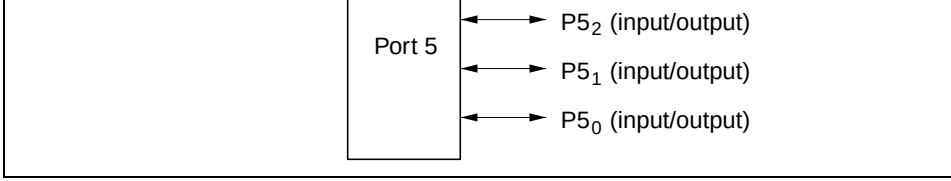


Figure 7.15 Pin Functions in Mode 6 and 7 (Port 5)

7.5.4 Input Pull-Up Transistors

Port 5 has built-in MOS input pull-up transistors that can be controlled by software. Th
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 inpu
transistor is turned on.

The input pull-up transistors are turned off by a reset and in hardware standby mode. In
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
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

RD, AS, and $P6_0$/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
Darlington transistor pair.

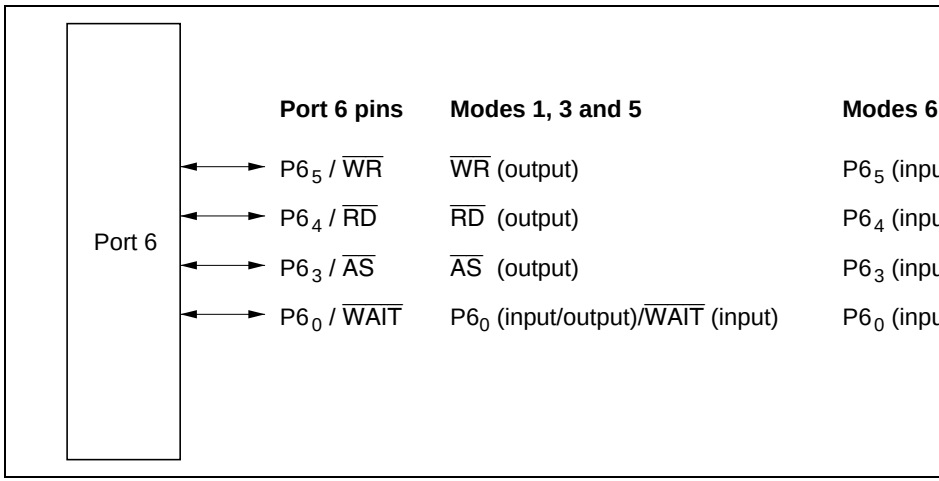


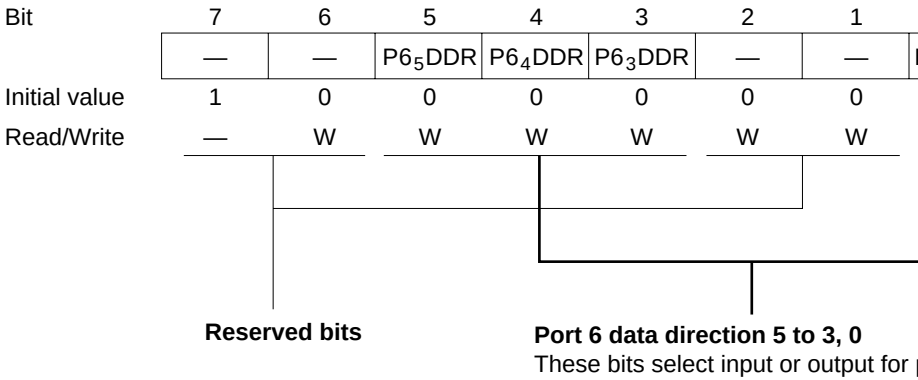
Figure 7.16 Port 6 Pin Configuration

	register				
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)

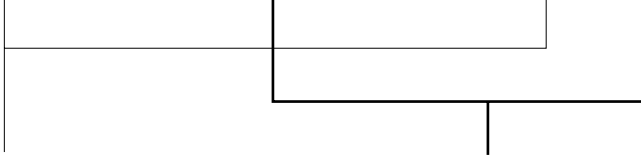
P6DDR is an 8-bit write-only register that can select input or output for each pin in por



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



Reserved bits

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

When a bit in P6DDR is set to 1, if port 6 is read the value of the corresponding P6DR returned directly. When a bit in P6DDR is cleared to 0, if port 6 is read the correspond is read. Bits 7, 6, 2, and 1 are reserved. Bit 7 cannot be modified and always reads 1. B 1 can be written and read, but cannot be used as ports. If bit 6, 2, or 1 in P6DDR is rea value is 1, the value of the corresponding bit in P6DR will be read. If bit 6, 2, or 1 in P 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 stan retains its previous setting.

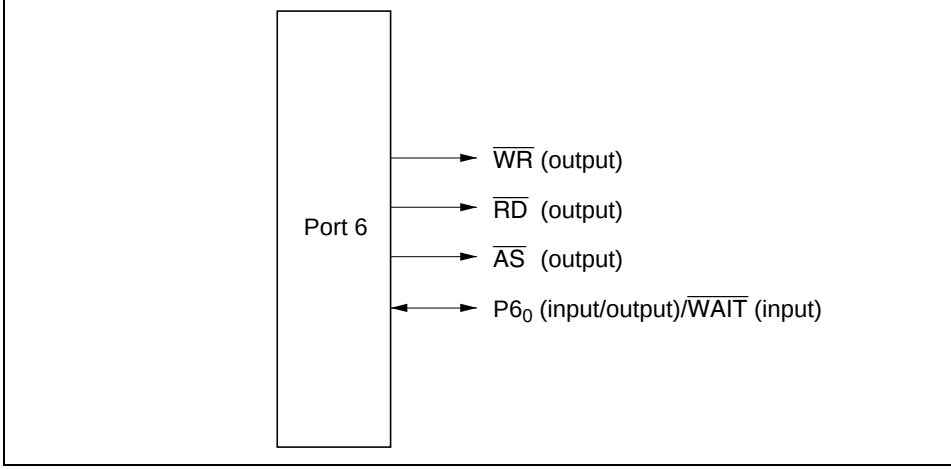


Figure 7.17 Pin Functions in Modes 1, 3, and 5 (Port 6)

$P6_4$DDR	0	1
Pin function	$\overline{RD}$ output	

$P6_3/\overline{AS}$ Functions as follows regardless of $P6_3$DDR

$P6_3$DDR	0	1
Pin function	$\overline{AS}$ output	

$P6_0/\overline{WAIT}$ Bits WCE7 to WCE0 in WCER, bit WMS1 in WCR, and bit $P6_0$DDR sele function as follows

WCER	All 1s			Not
WMS1	0		1	–
$P6_0$DDR	0	1	0*	0
Pin function	$P6_0$ input	$P6_0$ output	$\overline{WAIT}$ inp	

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

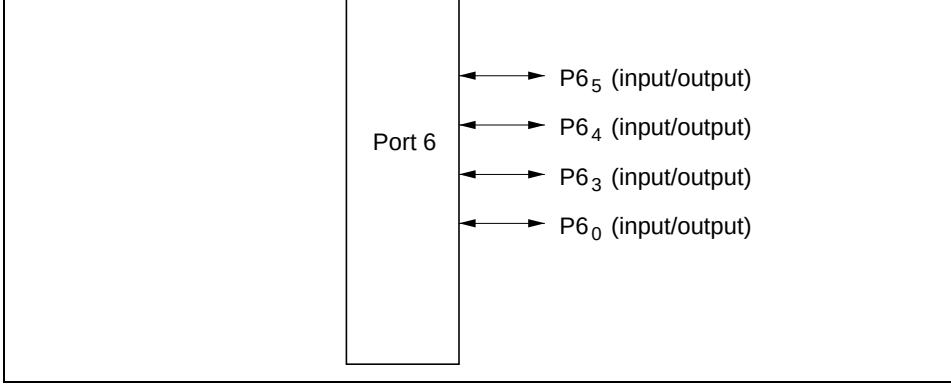


Figure 7.18 Pin Functions in Modes 6 and 7 (Port 6)

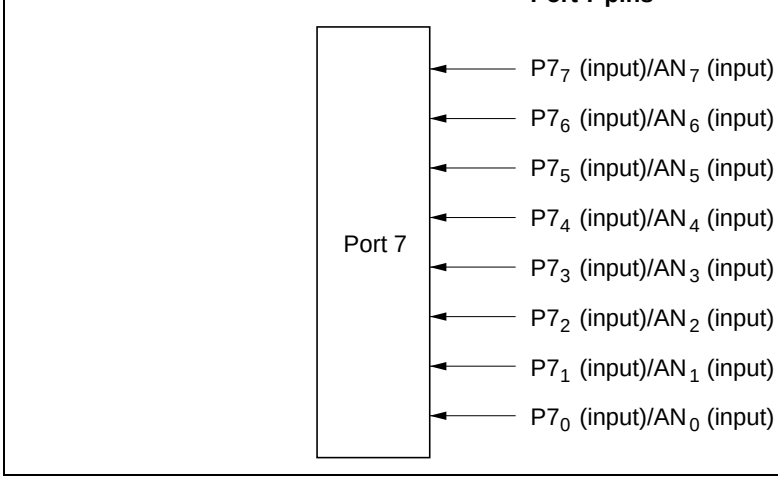


Figure 7.19 Port 7 Pin Configuration

7.7.2 Register Description

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

Table 7.10 Port 7 Data Register

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

Note: * Lower 16 bits of the address.

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 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 or $\overline{IRQ}_1$ input pins in modes 1, 3, and 5, and as input/output pins or $\overline{IRQ}_1$ input pins in m 7.

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

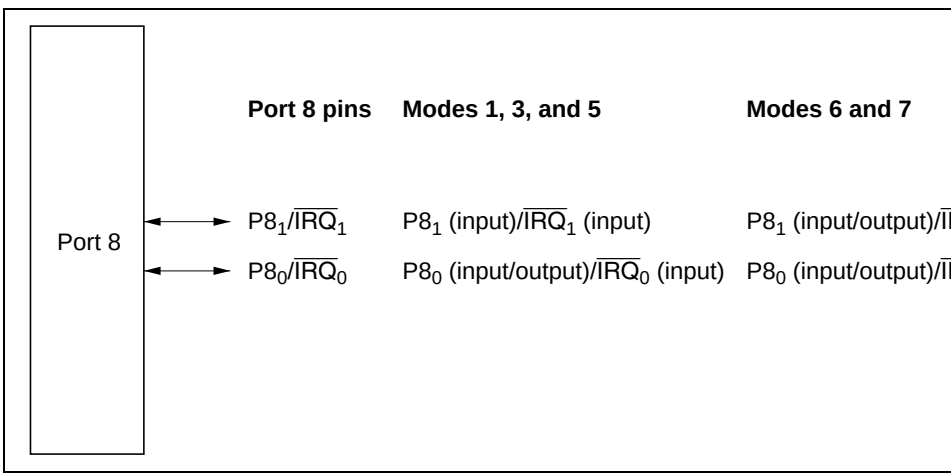
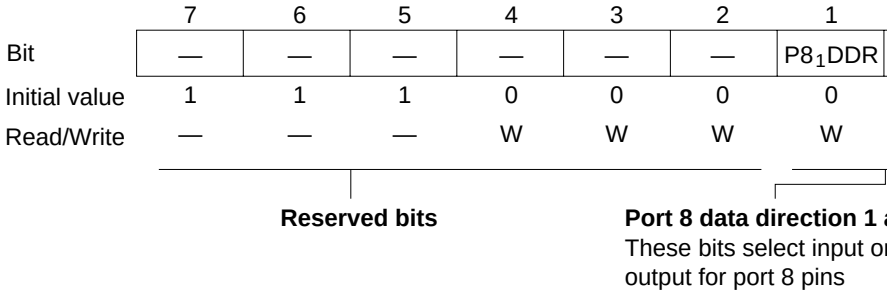


Figure 7.20 Port 8 Pin Configuration

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 po



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

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 returned directly. When a bit in P8DDR is cleared to 0, if port 8 is read the correspondi is read.

Bits 7 to 2 are reserved. Bits 7 to 5 cannot be modified and always read 1. Bit 4, 3, and written and read, but it cannot be used for port input or output. If bit 4, 3, and 2 of P8D while its value is 1, bit 4, 3 and 2 of P8DR is read directly. If bit 4, 3, and 2 of P8DDR 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 stand retains its previous setting.

$P8_1$DDR	0	1	
		Modes 1, 3, and 5	Modes
Pin function	$P8_1$ input	Illegal setting	$P8_1$
	$\overline{IRQ}_1$ 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	

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

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

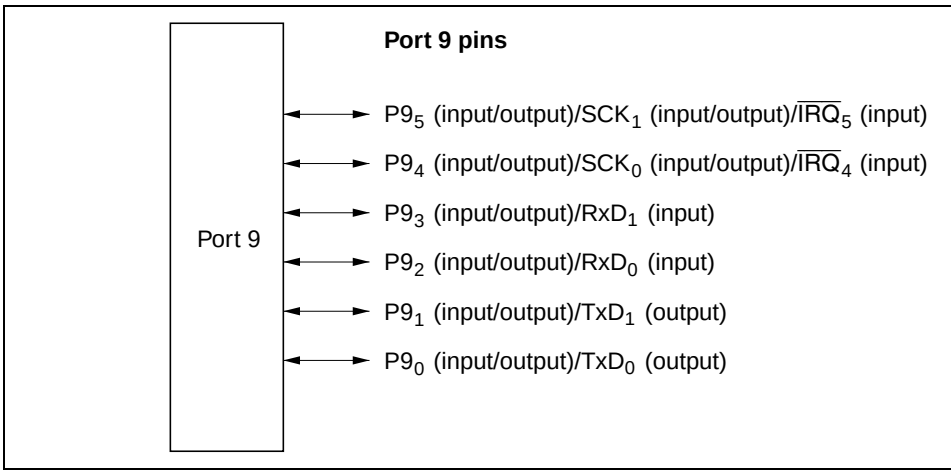


Figure 7.21 Port 9 Pin Configuration

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	Initi
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.

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 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 st it retains its previous setting. If a P9DDR bit is set to 1, the corresponding pin maintai 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
	—	—	$P9_5$	$P9_4$	$P9_3$	$P9_2$	$P9_1$
Initial value	1	1	0	0	0	0	0
Read/Write	—	—	R/W	R/W	R/W	R/W	R/W

Reserved bits

Port 9 data 5 to 0
These bits store data for port 9 pins

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

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

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 $P9_5$D the pin function as follows

CKE1	0				
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	SC
	$\overline{IRQ}_5$ 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 $P9_4$DD the pin function as follows

CKE1	0				
C/$\overline{A}$	0			1	
CKE0	0		1	—	
$P9_4$DDR	0	1	—	—	
Pin function	$P9_4$ input	$P9_4$ output	SCK_0 output	SCK_0 output	SC
	$\overline{IRQ}_4$ input				

follows

SMIF	0		
RE	0		1
$P9_2$DDR	0	1	—
Pin function	$P9_2$ input	$P9_2$ output	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
$P9_1$DDR	0	1	—
Pin function	$P9_1$ input	$P9_1$ output	TxD_1 outpu

$P9_0/TxD_0$ Bit TE in SCR of SCI0, bit SMIF in SCMR, and bit $P9_0$DDR select the pin f follows

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

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

Pins in port A can drive one TTL load and a 30-pF capacitive load. They can also drive
Darlington transistor pair. Port A has Schmitt-trigger inputs.

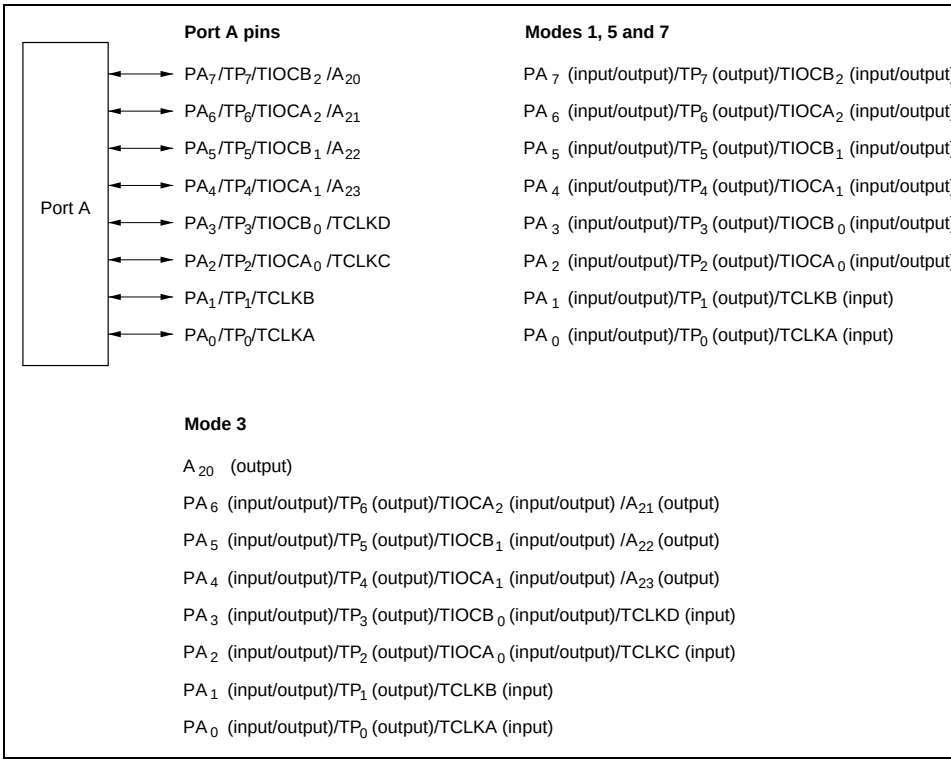


Figure 7.22 Port A Pin Configuration

	register				
H'FFD3	Port A data register	PADR	R/W	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 po
corresponding PADDR bit should also be set when a pin is used as a TPC output.

		7	6	5	4	3	2	1
	Bit	PA_7DDR	PA_6DDR	PA_5DDR	PA_4DDR	PA_3DDR	PA_2DDR	PA_1D
Modes 1, 5, and 7	Initial value	0	0	0	0	0	0	0
	Read/Write	W	W	W	W	W	W	W
Mode 3	Initial value	1	0	0	0	0	0	0
	Read/Write	—	W	W	W	W	W	W

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

A pin in port A becomes an output pin if the corresponding PADDR bit is set to 1, and
pin if this bit is cleared to 0. However, in mode 3, PA_7 DDR is fixed at 1, and PA7 fun
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
hardware standby mode. In software standby mode it retains its previous setting. If a P
is set to 1, the corresponding pin maintains its output state in software standby mode.

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 PADI returned directly. When a bit in PADDR is cleared to 0, if port A is read the correspond level is read.

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

When port A pins are used for TPC output, PADR stores output data for TPC output gr 1. If a bit in the next data enable register (NDERA) is set to 1, the corresponding PADI be written. In this case, PADR can be updated only when data is transferred from NDR

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

The mode setting, ITU channel 2 settings (bit PWM2 in TMDR and bits IOI in TIOR2), bit NDER7 in NDERA, and bit PA7DDR in PADDR select the p as follows

Mode	1, 5 to 7			
ITU channel 2 settings	(1) in table below	(2) in table below		
PA_7DDR	—	0	1	1
NDER7	—	—	0	1
Pin function	$TIOCB_2$ output	PA_7 input	PA_7 output	TP_7 outpu
		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	—	—

settings	below				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	T out
		$TIOCA_2$ 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	—	—	–

settings	below				below			
PA_5DDR	—	0	1	1	—	0	1	
NDER5	—	—	0	1	—	—	0	
Pin function	$TIOCB_1$ output	PA_5 input	PA_5 output	TP_5 output	$TIOCB_1$ output	PA_5 input	PA_5 output	o
		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	—	—

settings	below				below			
PA_4DDR	—	0	1	1	—	0	1	
NDER4	—	—	0	1	—	—	0	
Pin function	$TIOCA_1$ output	PA_4 input	PA_4 output	TP_5 output	$TIOCA_1$ output	PA_4 input	PA_4 output	T ou
		$TIOCA_1$ input*				$TIOCA_1$ input*		

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

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

NDER3	—	—	0
Pin function	$TIOCB_0$ output	PA_3 input	PA_3 output
		$TIOCB_0$ input	
	TCLKD input*2		

Notes: 1. TIOCB0 input when IOB2 = 1 and PWM0 = 0.

2. TCLKD input when TPSC2 = TPSC1 = TPSC0 = 1 in any of TC TCR0.

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

NDER2	—			—	0
Pin function	$TIOCA_0$ output			PA_2 input	PA_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 TCR0.

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

Note: TCLKB input when MDF = 1 in TMDR, or when TPSC2 = 1, TP
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 a

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

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

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

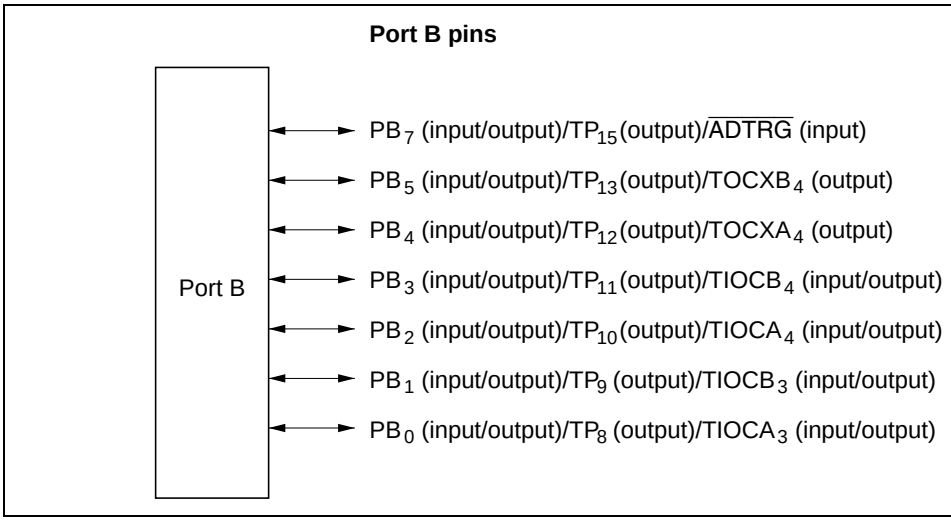


Figure 7.23 Port B Pin Configuration

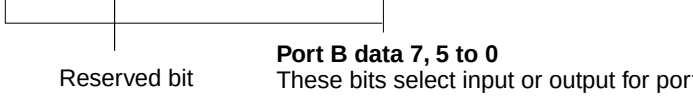
7.11.2 Register Descriptions

Table 7.17 summarizes the registers of port B.

Table 7.17 Port B Registers

Address*	Name	Abbreviation	R/W	Ini
H'FFD4	Port B data direction register	PBDDR	W	H'(
H'FFD6	Port B data register	PBDR	R/W	H'(

Note: * Lower 16 bits of the address.



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

Bit 6 is reserved.

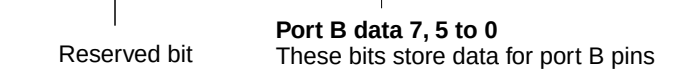
PBDDR is a write-only register. Its value cannot be read. All bits return 1 when read.

PBDDR is initialized to H'00 by a reset and in hardware standby mode. In software st it retains its previous setting. If a PBDDR bit is set to 1, the corresponding pin maintai 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
	PB_7	—	PB_5	PB_4	PB_3	PB_2	PB_1
Initial value	0	0	0	0	0	0	0
Read/Write	R/W	R/W	R/W	R/W	R/W	R/W	R/W



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

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 (TIO
$TIOCB_3$, $TIOCA_4$, $TIOCA_3$) and output ($TOCXB_4$, $TOCXA_4$), and $\overline{ADTRG}$ input. Tabl
describes the selection of pin functions.

				$\overline{\text{ADTRG}}$ input*

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

PB_5/ TP_{13}/ $TOCXB_4$

ITU channel 4 settings (bit CMD1 in TFCR and bit EXB4 in TOER), bit ND
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	$TOCXB_4$ o

PB_4/ TP_{12}/ $TOCXA_4$

ITU channel 4 settings (bit CMD1 in TFCR and bit EXA4 in TOER), bit ND
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	$TOCXA_4$ o

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

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

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

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

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

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

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

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

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

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

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

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

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 outpu input capture functions
- Selection of eight counter clock sources for each channel:
  Internal clocks: ϕ, ϕ/2, ϕ/4, ϕ/8
  External clocks: TCLKA, TCLKB, TCLKC, TCLKD
- Five operating modes selectable in all channels:
  — Waveform output by compare match
    Selection of 0 output, 1 output, or toggle output (only 0 or 1 output in channel
  — Input capture function
    Rising edge, falling edge, or both edges (selectable)
  — Counter clearing function
    Counters can be cleared by compare match or input capture
  — Synchronization
    Two or more timer counters (TCNTs) can be preset simultaneously, or cleared simultaneously by compare match or input capture. Counter synchronization e synchronous register input and output.
  — PWM mode
    PWM output can be provided with an arbitrary duty cycle. With synchronizatio five-phase PWM output is possible
- Phase counting mode selectable in channel 2
  Two-phase encoder output can be counted automatically.

Buffering

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

- High-speed access via internal 16-bit bus

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

- Fifteen interrupt sources

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

- Output triggering of programmable pattern controller (TPC)

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

Table 8.1 summarizes the ITU functions.

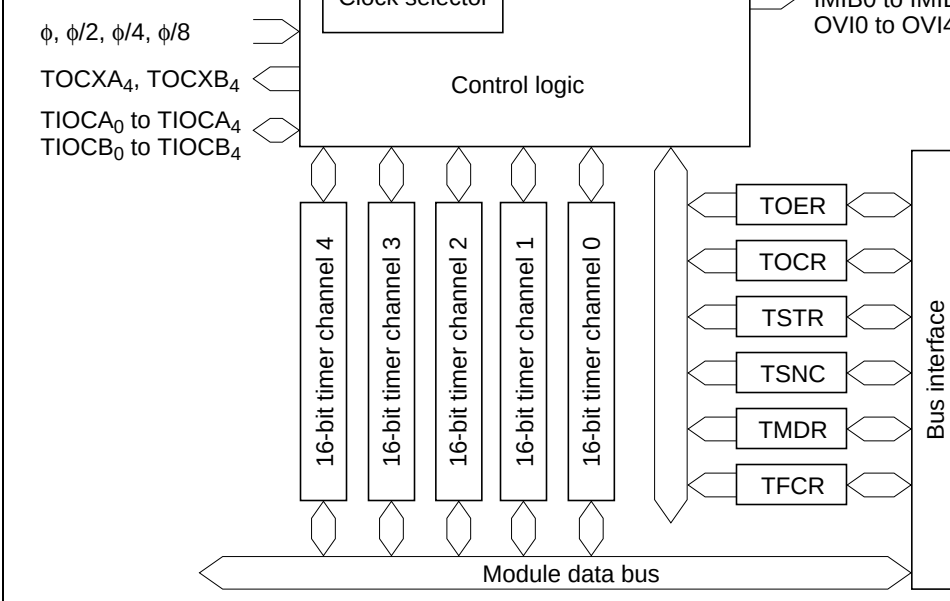
Input/output pins		$TIOCA_0$, $TIOCB_0$	$TIOCA_1$, $TIOCB_1$	$TIOCA_2$, $TIOCB_2$	$TIOCA_3$, $TIOCB_3$	TI
Output pins		—	—	—	—	TC TC
Counter clearing function		GRA0/GRB0 compare match or input capture	GRA1/GRB1 compare match or input capture	GRA2/GRB2 compare match or input capture	GRA3/GRB3 compare match or input capture	GI co or
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	Th • r c • r c •

Legend:

O: Available

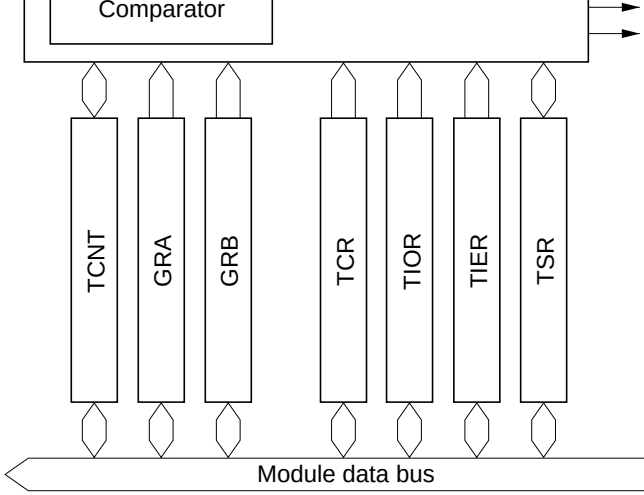
—: Not available

RENESAS



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

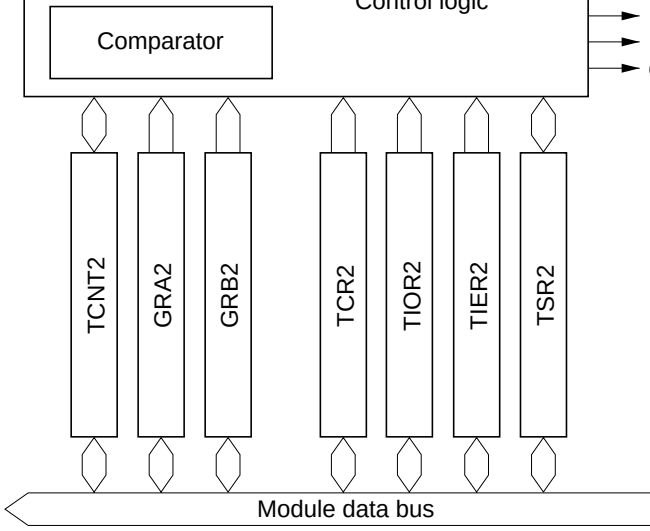
Figure 8.1 ITU Block Diagram (Overall)



Legend:

TCNT:	Timer counter (16 bits)
GRA, GRB:	General registers A and B (input capture/output compare registers) (16
TCR:	Timer control register (8 bits)
TIOR:	Timer I/O control register (8 bits)
TIER:	Timer interrupt enable register (8 bits)
TSR:	Timer status register (8 bits)

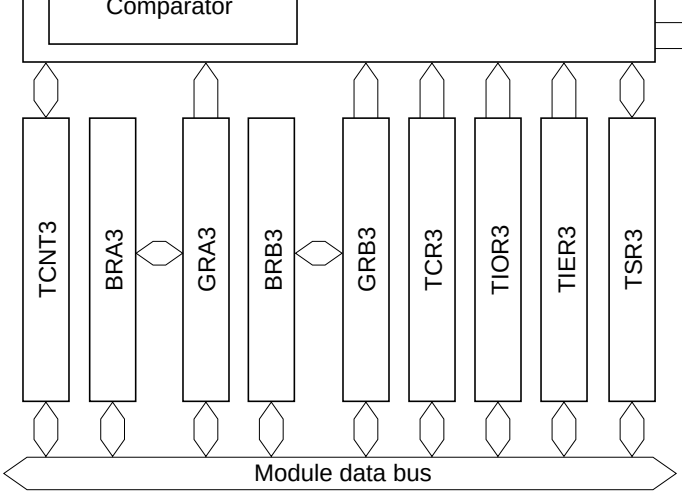
Figure 8.2 Block Diagram of Channels 0 and 1 (for Channel 0)



Legend:

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

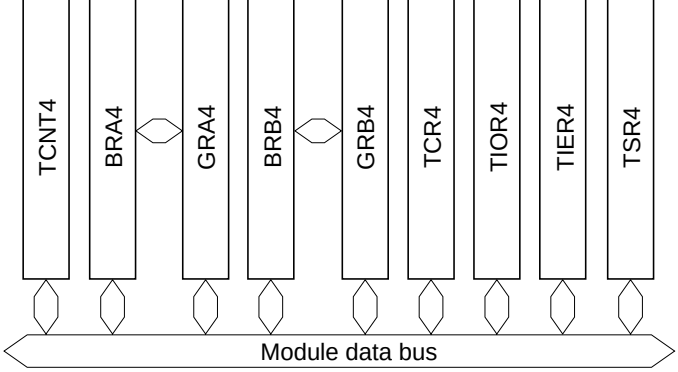
Figure 8.3 Block Diagram of Channel 2



Legend:

TCNT3:	Timer counter 3 (16 bits)
GRA3, GRB3:	General registers A3 and B3 (input capture/output compare registe (16 bits × 2)
BRA3, BRB3:	Buffer registers A3 and B3 (input capture/output compare buffer re (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 regis (16 bits × 2)
TCR4:	Timer control register 4 (8 bits)
TIOR4:	Timer I/O control register 4 (8 bits)
TIER4:	Timer interrupt enable register 4 (8 bits)
TSR4:	Timer status register 4 (8 bits)

Figure 8.5 Block Diagram of Channel 4

				(phase-A input pin in phase co mode)
	Clock input B	TCLKB	Input	External clock B input pin (phase-B input pin in phase co 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 inpu PWM output pin in PWM mod
	Input capture/output compare B0	$TIOCB_0$	Input/ output	GRB0 output compare or inpu
1	Input capture/output compare A1	$TIOCA_1$	Input/ output	GRA1 output compare or inpu PWM output pin in PWM mod
	Input capture/output compare B1	$TIOCB_1$	Input/ output	GRB1 output compare or inpu
2	Input capture/output compare A2	$TIOCA_2$	Input/ output	GRA2 output compare or inpu PWM output pin in PWM mod
	Input capture/output compare B2	$TIOCB_2$	Input/ output	GRB2 output compare or inpu
3	Input capture/output compare A3	$TIOCA_3$	Input/ output	GRA3 output compare or inpu PWM output pin in PWM mod complementary PWM mode, o synchronized PWM mode
	Input capture/output compare B3	$TIOCB_3$	Input/ output	GRB3 output compare or inpu PWM output pin in compleme mode or reset-synchronized P

Output compare XA4	$TOCXA_4$	Output	PWM output pin in complement mode or reset-synchronized PW
Output compare XB4	$TOCXB_4$	Output	PWM output pin in complement mode or reset-synchronized PW

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

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

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

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

	—	—	—	STR4	STR3	STR2	STR1
Initial value	1	1	1	0	0	0	0
Read/Write	—	—	—	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	(Ini
1	TCNT4 is counting	

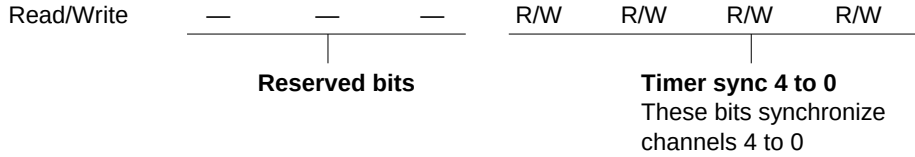
Bit 3—Counter Start 3 (STR3): Starts and stops timer counter 3 (TCNT3).

Bit 3 STR3	Description	
0	TCNT3 is halted	(Ini
1	TCNT3 is counting	

Bit 1 STR1	Description	
0	TCNT1 is halted	(In
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	(In
1	TCNT0 is counting	



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 cleared independently of other channels (Ini
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 cleared independently of other channels (Ini
1	Channel 3 operates synchronously TCNT3 can be synchronously preset and cleared

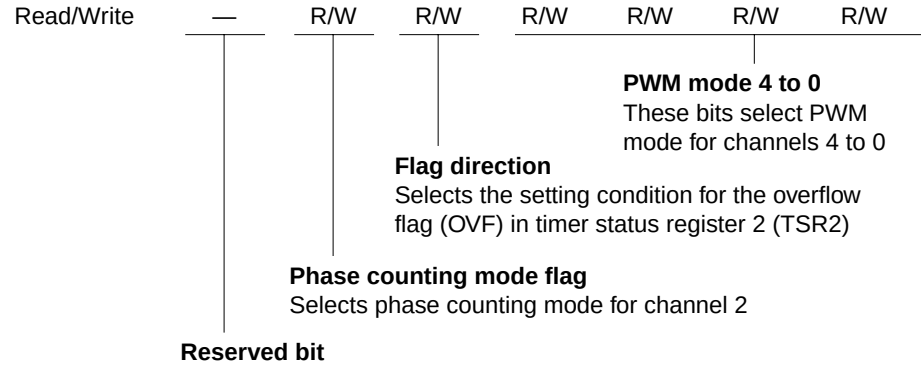
	TCNT2 can be synchronously preset and cleared

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 prese cleared independently of other channels (In
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 prese cleared independently of other channels (In
1	Channel 0 operates synchronously TCNT0 can be synchronously preset and cleared



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

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

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

Bit 6 MDF	Description	
0	Channel 2 operates normally	(Ini
1	Channel 2 operates in phase counting mode	

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

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

Bit 5—Flag Direction (FDIR): Designates the setting condition for the overflow flag timer status register 2 (TSR2). The FDIR designation is valid in all modes in channel

Bit 5 FDIR	Description	
0	OVF is set to 1 in TSR2 when TCNT2 overflows or underflows	(In
1	OVF is set to 1 in TSR2 when TCNT2 overflows	

Bit 4—PWM Mode 4 (PWM4): Selects whether channel 4 operates normally or in P

Bit 4 PWM4	Description	
0	Channel 4 operates normally	(In
1	Channel 4 operates in PWM mode	

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

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

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

Bit 3 PWM3	Description	
0	Channel 3 operates normally	(In
1	Channel 3 operates in PWM mode	

Bit 2 PWM2	Description	
0	Channel 2 operates normally	(Ini
1	Channel 2 operates in PWM mode	

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

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

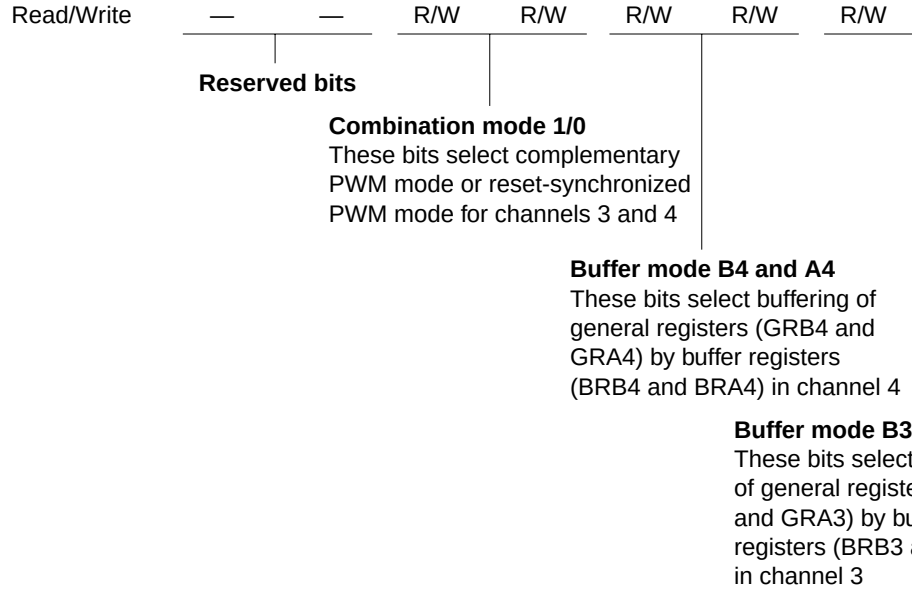
Bit 1 PWM1	Description	
0	Channel 1 operates normally	(Ini
1	Channel 1 operates in PWM mode	

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

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

Bit 0 PWM0	Description	
0	Channel 0 operates normally	(Ini
1	Channel 0 operates in PWM mode	

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



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.

1	Channels 3 and 4 operate together in reset-synchronized PWM m

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

When these bits select complementary PWM mode or reset-synchronized PWM mode, precedence over the setting of the PWM mode bits (PWM4 and PWM3) in TMDR. Set 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 comp PWM mode is selected, channels 3 and 4 must not be synchronized (do not set bits SYN SYNC4 both to 1 in TSNC).

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

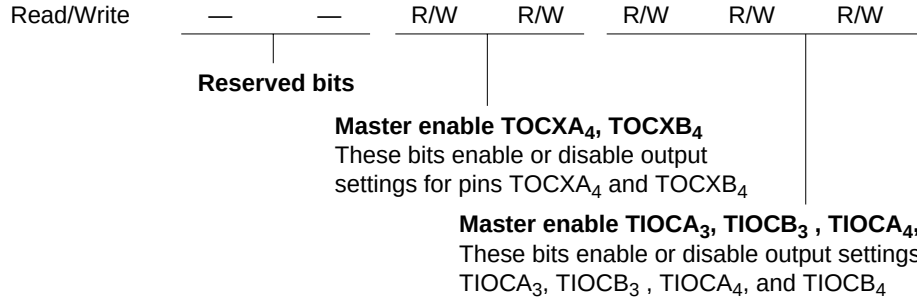
Bit 3 BFB4	Description	
0	GRB4 operates normally	(Ini
1	GRB4 is buffered by BRB4	

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

Bit 2 BFA4	Description	
0	GRA4 operates normally	(Ini
1	GRA4 is buffered by BRA4	

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

Bit 0 BFA3	Description	
0	GRA3 operates normally	(In
1	GRA3 is buffered by BRA3	



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 TOC

Bit 5 EXB4	Description
0	TOCXB$_4$ output is disabled regardless of TFCR settings (TOCXB$_4$ operates a input/output pin). If XTGD = 0, EXB4 is cleared to 0 when input capture A oc channel 1.
1	TOCXB$_4$ is enabled for output according to TFCR settings (Ini

Bit 4—Master Enable TOCXA$_4$ (EXA4): Enables or disables ITU output at pin TOC

Bit 4 EXA4	Description
0	TOCXA$_4$ output is disabled regardless of TFCR settings (TOCXA$_4$ operates a 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 (Ini

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

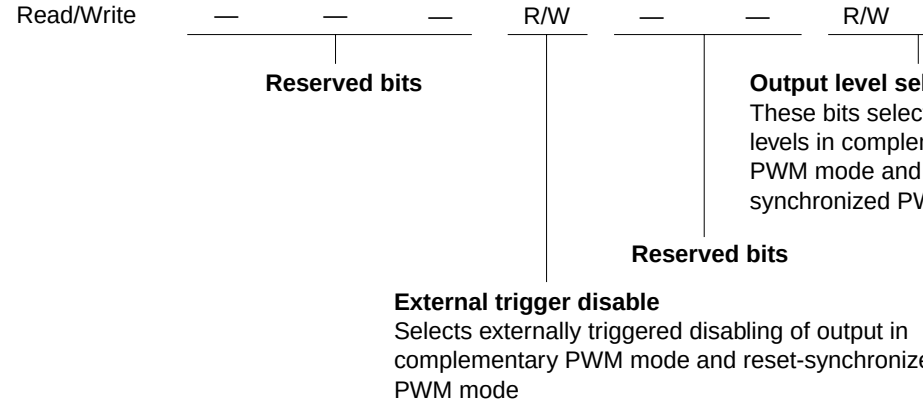
Bit 2 EB4	Description
0	TIOCB$_4$ output is disabled regardless of TIOR4 and TFCR settings (TIOCB$_4$ a generic input/output pin). If XTGD = 0, EB4 is cleared to 0 when input capture A occurs in channel 1.
1	TIOCB$_4$ is enabled for output according to TIOR4 and TFCR settings (In

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

Bit 1 EA4	Description
0	TIOCA$_4$ output is disabled regardless of TIOR4, TMDR, and TFCR settings 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 setting (In

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

Bit 0 EA3	Description
0	TIOCA$_3$ output is disabled regardless of TIOR3, TMDR, and TFCR settings 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 setting (In



The settings of the XTGD, OLS4, and OLS3 bits are valid only in complementary PWM 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 IT 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 complem PWM mode and reset-synchronized PWM mode. When an external trigger occurs, bits 5 to 0 in the timer output master enable (TOER) are cleared to 0, disabling ITU output.
1	External triggering is disabled (Ini

1	$TIOCB_3$, $TOCXA_4$, and $TOCXB_4$ pin outputs are not inverted	(In

Bit 0—Output Level Select 3 (OLS3): Selects output levels in complementary PWM 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	(In

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

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

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

TCNT can be cleared to H'0000 by compare match with general register A or B (GRA
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
the timer status register (TSR) of the corresponding channel.

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

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

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

3	GRA3, GRB3	Output compare/input capture register; can be by buffer r
4	GRA4, GRB4	BRA and BRB

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

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

When a general register is used as an output compare register, its value is constantly c with the TCNT value. When the two values match (compare match), the IMFA or IM to 1 in 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 edges of an external input capture signal are detected and the current TCNT value is s general register. The corresponding IMFA or IMFB flag in TSR is set to 1 at the same 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-sync PWM mode.

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

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

compare register, BRA or BRB can function as an outp
buffer register: the BRA or BRB value is automatically t
to GRA or GRB at compare match

- When the corresponding GRA or GRB functions as an
capture register, BRA or BRB can function as an input
buffer register: the GRA or GRB value is automatically
to BRA or BRB at input capture

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

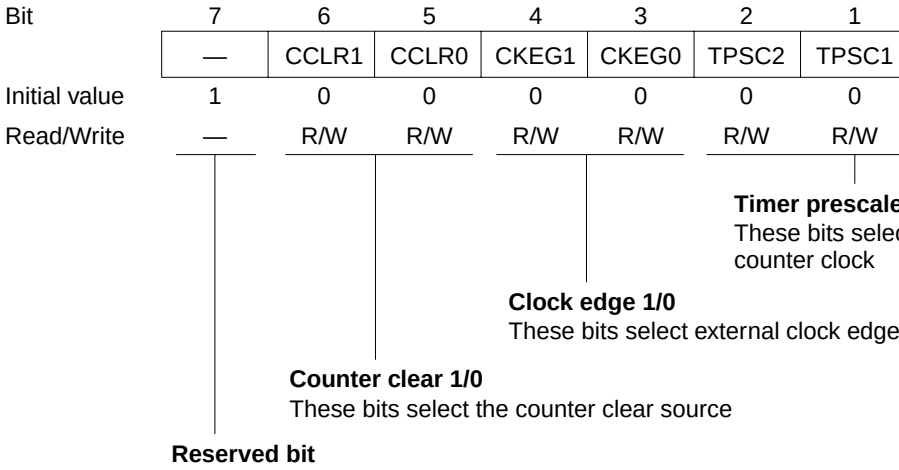
A buffer register is a 16-bit readable/writable register that is used when buffering is sel
Buffering can be selected independently by bits BFB4, BFA4, BFB3, and BFA3 in TF

The buffer register and general register operate as a pair. When the general register fun
output compare register, the buffer register functions as an output compare buffer regis
the general register functions as an input capture register, the buffer register functions a
capture buffer register.

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

Buffer registers are initialized to H'FFFF by a reset and in standby mode.

3	TCR3
4	TCR4



Each TCR is an 8-bit readable/writable register that selects the timer counter clock sou the edge or edges of external clock sources, and selects how the counter is cleared.

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

Bit 7—Reserved: 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 i

compare match register, and by input capture when the general register func
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 in
when an external clock source is used.

Bit 4 CKEG1	Bit 3 CKEG0	Description	
0	0	Count rising edges	(Initi
	1	Count falling edges	
1	—	Count both edges	

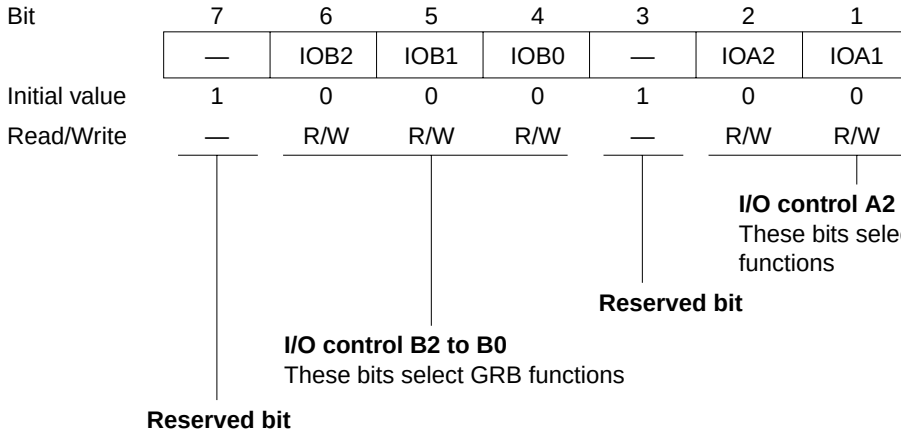
When channel 2 is set to phase counting mode, bits CKEG1 and CKEG0 in TCR2 are i
Phase counting takes precedence.

Bits 2 to 0—Timer Prescaler 2 to 0 (TPSC2 to TPSC0): These bits select the counter
source.

Bit 2 TPSC2	Bit 1 TPSC1	Bit 0 TPSC0	Function	
0	0	0	Internal clock: ϕ	(Initi
		1	Internal clock: $\phi/2$	
	1	0	Internal clock: $\phi/4$	
		1	Internal clock: $\phi/8$	
1	0	0	External clock A: TCLKA input	
		1	External clock B: TCLKB input	
	1	0	External clock C: TCLKC input	
		1	External clock D: TCLKD input	

TIOR is an 8-bit register. The ITU has five TIORs, one in each channel.

Channel	Abbreviation	Function
0	TIOR0	TIOR controls the general registers. Some functions diffe mode. TIOR3 and TIOR4 settings are ignored when com PWM mode or reset-synchronized PWM mode is selecte 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 inp function for GRA and GRB, and specifies the functions of the TIOCA and TIOCB pin output compare function is selected, TIOR also selects the type of output. If input capt 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.

	1	0		1 output at GRB compare match[1]
		1		Output toggles at GRB compare mat (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 instead.

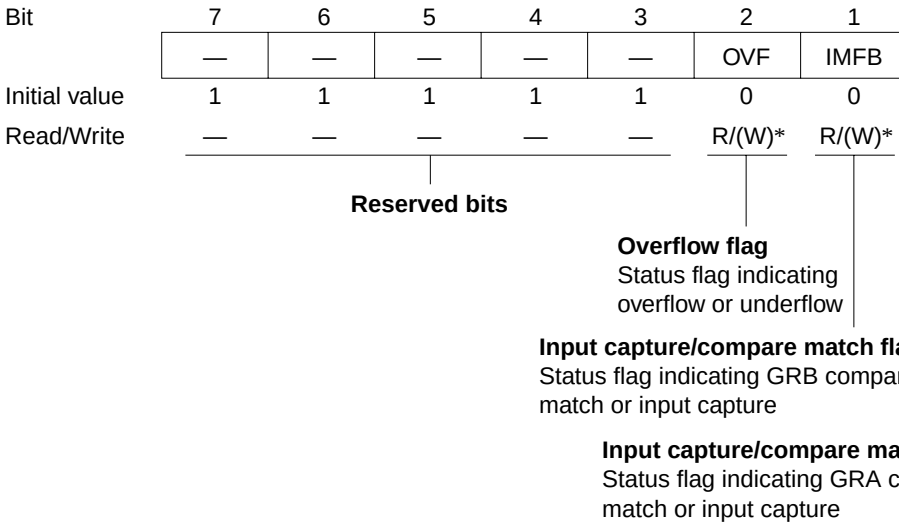
Bit 3—Reserved: This bit cannot be modified and is always read as 1.

Bits 2 to 0—I/O Control A2 to A0 (IOA2 to IOA0): These bits select the GRA functi

Bit 2 IOA2	Bit 1 IOA1	Bit 0 IOA0	Function	
0	0	0	GRA is an output compare register	No output at compare match (In
		1		0 output at GRA compare match*[1]
	1	0		1 output at GRA compare match*[1]
		1		Output toggles at GRA compare mat (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 instead.

3	TSR3
4	TSR4



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

Each TSR is an 8-bit readable/writable register containing flags that indicate TCNT ov
underflow and GRA or GRB compare match or input capture. These flags are interrup
and generate CPU interrupts if enabled by corresponding bits in the timer interrupt ena
(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.

Notes: * TCNT underflow occurs when TCNT operates as an up/down-counter. Unde
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
in TFCR)

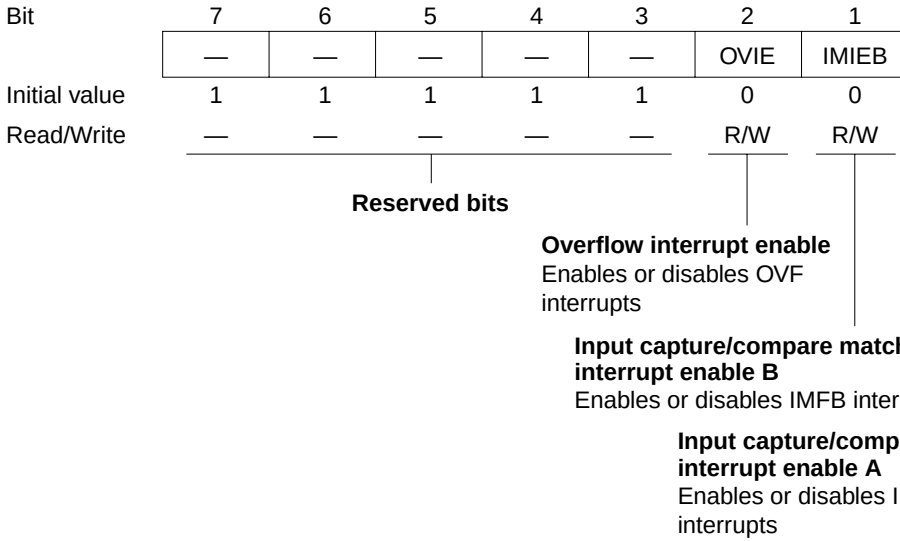
Bit 1—Input Capture/Compare Match Flag B (IMFB): This status flag indicates GR
compare match or input capture events.

Bit 1 IMFB	Description
0	[Clearing condition] Read IMFB when IMFB = 1, then write 0 in IMFB (Ini
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 as an input capture register.

Bit 0—Input Capture/Compare Match Flag A (IMFA): This status flag indicates GR
compare match or input capture events.

Bit 0 IMFA	Description
0	[Clearing condition] Read IMFA when IMFA = 1, then write 0 in IMFA. (Ini
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 as an input capture register.

3	TIER3
4	TIER4



Each TIER is an 8-bit readable/writable register that enables and disables overflow int 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 1—Input Capture/Compare Match Interrupt Enable B (IMIEB): Enables or di
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	(Ini
1	IMIB interrupt requested by IMFB is enabled	

Bit 0—Input Capture/Compare Match Interrupt Enable A (IMIEA): Enables or di
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	(Ini
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
and B (BRAs and BRBs) are 16-bit registers, and are linked to the CPU by an internal 1
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
8.10, and 8.11 show examples of byte access to TCNTH and TCNTL.

Figure 8.6 Access to Timer Counter (CPU Writes to TCNT, Word)

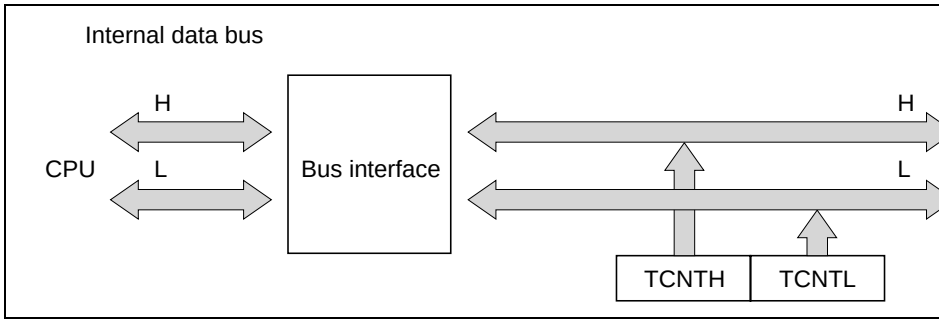


Figure 8.7 Access to Timer Counter (CPU Reads TCNT, Word)

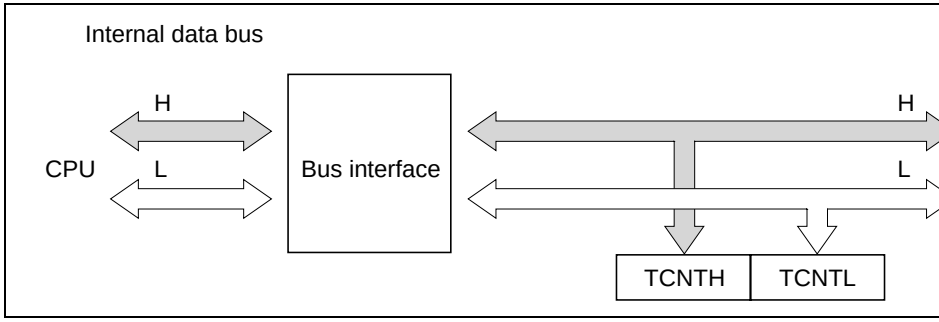


Figure 8.8 Access to Timer Counter (CPU Writes to TCNT, Upper By

Figure 8.9 Access to Timer Counter (CPU Writes to TCNT, Lower Byte)

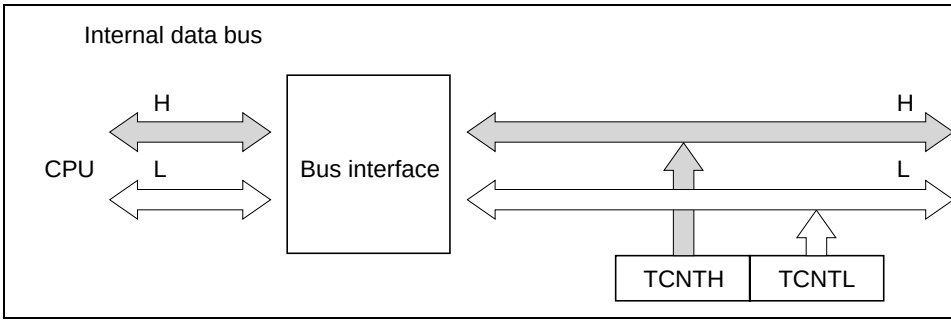


Figure 8.10 Access to Timer Counter (CPU Reads TCNT, Upper Byte)

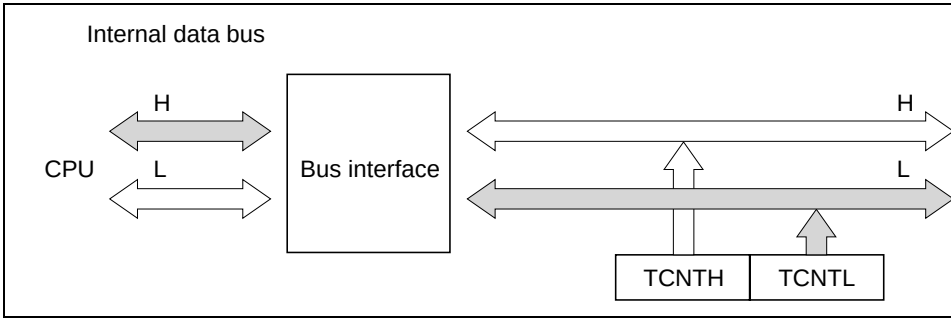


Figure 8.11 Access to Timer Counter (CPU Reads TCNT, Lower Byte)

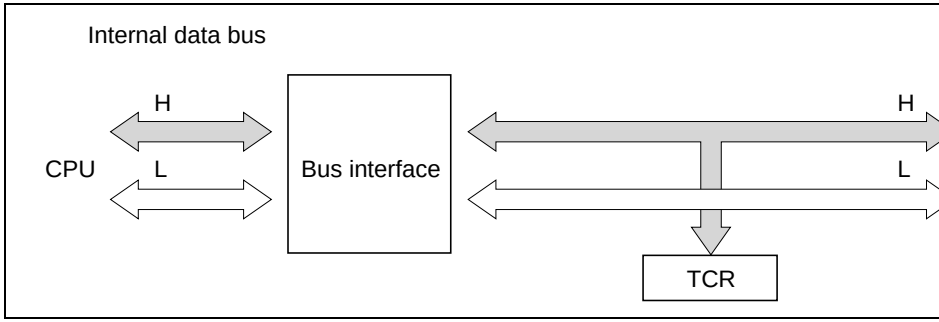


Figure 8.12 TCR Access (CPU Writes to TCR)

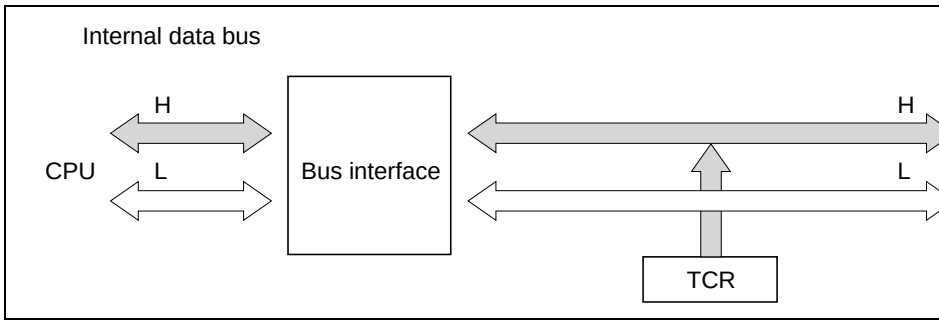


Figure 8.13 TCR Access (CPU Reads TCR)

General registers A and B can be used for input capture or output compare.

Synchronous Operation: The timer counters in designated channels are preset synchro
Data written to the timer counter in any one of these channels is simultaneously written
timer counters in the other channels as well. The timer counters can also be cleared syn
if so designated by the CCLR1 and CCLR0 bits in the TCRs.

PWM Mode: A PWM waveform is output from the TIOCA pin. The output goes to 1 a
match A and to 0 at compare match B. The duty cycle can be varied from 0% to 100%
on the settings of GRA and GRB. When a channel is set to PWM mode, its GRA and G
automatically become output compare registers.

Reset-Synchronized PWM Mode: Channels 3 and 4 are paired for three-phase PWM
complementary waveforms. (The three phases are related by having a common transitio
When reset-synchronized PWM mode is selected GRA3, GRB3, GRA4, and GRB4 aut
function as output compare registers, $TIOCA_3$, $TIOCB_3$, $TIOCA_4$, $TOCXA_4$, $TIOCB_4$, a
$TOCXB_4$ function as PWM output pins, and $TCNT_3$ operates as an up-counter. TCNT4
independently, and is not compared with GRA4 or GRB4.

Complementary PWM Mode: Channels 3 and 4 are paired for three-phase PWM outp
non-overlapping complementary waveforms. When complementary PWM mode is sele
GRA3, GRB3, GRA4, and GRB4 automatically function as output compare registers, a
$TIOCA_3$, $TIOCB_3$, $TIOCA_4$, $TOCXA_4$, $TIOCB_4$, and $TOCXB_4$ function as PWM output
TCNT3 and TCNT4 operate as up/down-counters.

Phase Counting Mode: The phase relationship between two clock signals input at TCl
TCLKB is detected and TCNT2 counts up or down accordingly. When phase counting
selected TCLKA and TCLKB become clock input pins and TCNT2 operates as an up/d
counter.

The buffer register value is transferred to the general register when TCNT3 and TC
change counting direction.

- Reset-synchronized PWM mode
  The buffer register value is transferred to the general register at GRA3 compare m

Sample setup procedure for counter: Figure 8.14 shows a sample procedure for setti
counter.

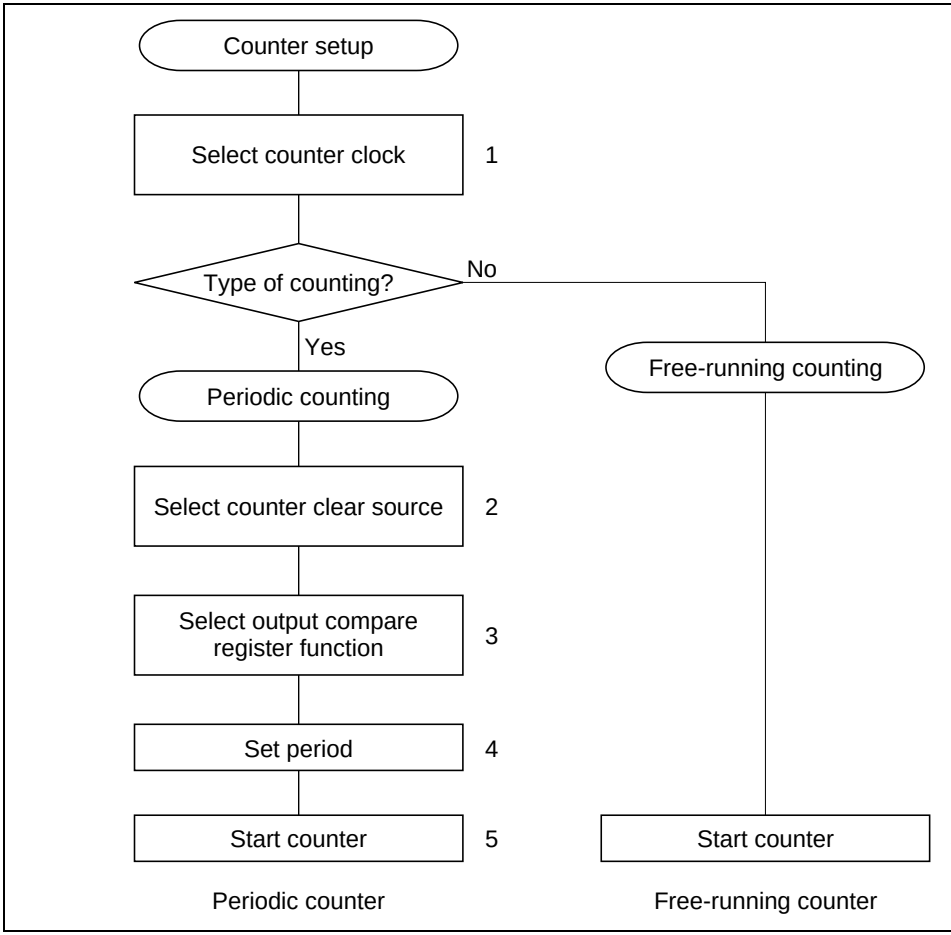


Figure 8.14 Counter Setup Procedure (Example)

5. Set the STR bit to 1 in TSTR to start the timer counter.

Free-running and periodic counter operation: A reset leaves the counters (TCNTs) channels 0 to 4 all set as free-running counters. A free-running counter starts counting corresponding bit in TSTR is set to 1. When the count overflows from H'FFFF to H'00 overflow flag (OVF) is set to 1 in the timer status register (TSR). If the corresponding set to 1 in the timer interrupt enable register, a CPU interrupt is requested. After the ov counter continues counting up from H'0000. Figure 8.15 illustrates free-running count

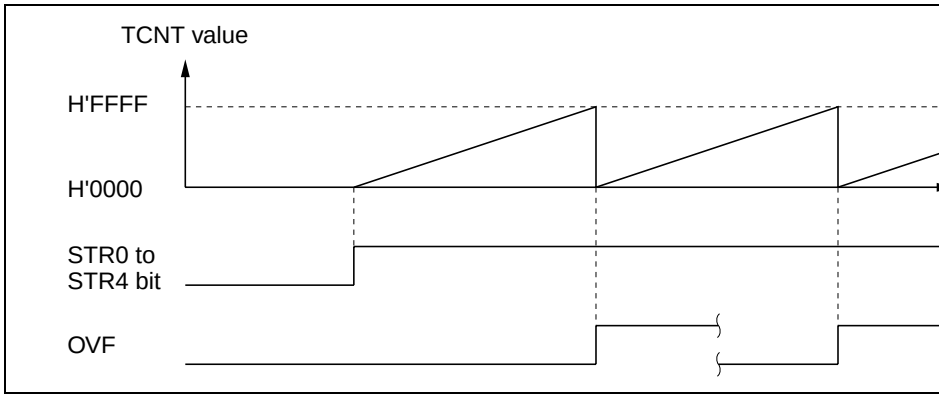


Figure 8.15 Free-Running Counter Operation

When a channel is set to have its counter cleared by compare match, in that channel T operates as a periodic counter. Select the output compare function of GRA or GRB, se or CCLR0 in the timer control register (TCR) to have the counter cleared by compare set the count period in GRA or GRB. After these settings, the counter starts counting u periodic counter when the corresponding bit is set to 1 in TSTR. When the count matc GRB, the IMFA or IMFB flag is set to 1 in TSR and the counter is cleared to H'0000. corresponding IMIEA or IMIEB bit is set to 1 in TIER, a CPU interrupt is requested a After the compare match, TCNT continues counting up from H'0000. Figure 8.16 illus periodic counting.

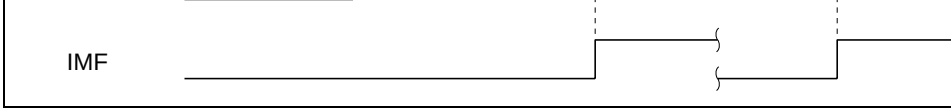


Figure 8.16 Periodic Counter Operation

Count timing:

- Internal clock source

  Bits TPSC2 to TPSC0 in TCR select the system clock (ϕ) or one of three internal cl
  obtained by prescaling the system clock (ϕ/2, ϕ/4, ϕ/8).
  Figure 8.17 shows the timing.

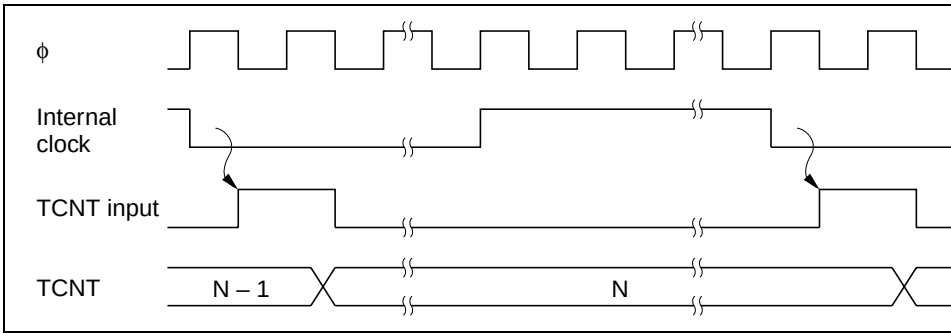


Figure 8.17 Count Timing for Internal Clock Sources

- External clock source

  Bits TPSC2 to TPSC0 in TCR select an external clock input pin (TCLKA to TCLK
  valid edge or edges are selected by bits CKEG1 and CKEG0. The rising edge, fallin
  both edges can be selected.

  The pulse width of the external clock signal must be at least 1.5 system clocks when
  edge is selected, and at least 2.5 system clocks when both edges are selected. Shorte
  will not be counted correctly.

  Figure 8.18 shows the timing when both edges are detected.

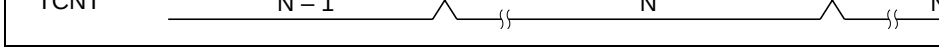


Figure 8.18 Count Timing for External Clock Sources (when Both Edges are

Waveform Output by Compare Match

In ITU channels 0, 1, 3, and 4, compare match A or B can cause the output at the TIO
TIOCB pin to go to 0, go to 1, or toggle. In channel 2 the output can only go to 0 or g

Sample setup procedure for waveform output by compare match: Figure 8.19 sho
procedure for setting up waveform output by compare match.

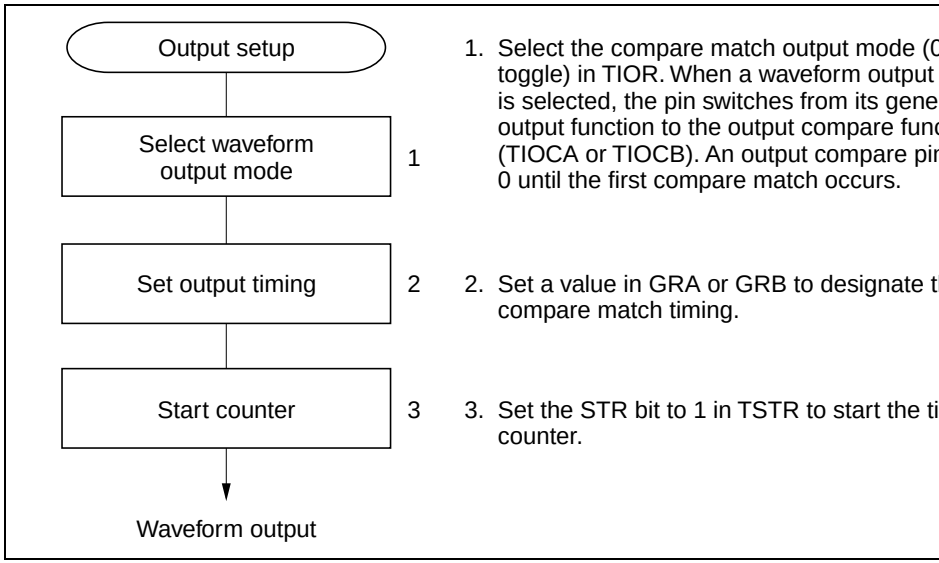


Figure 8.19 Setup Procedure for Waveform Output by Compare Match (Ex

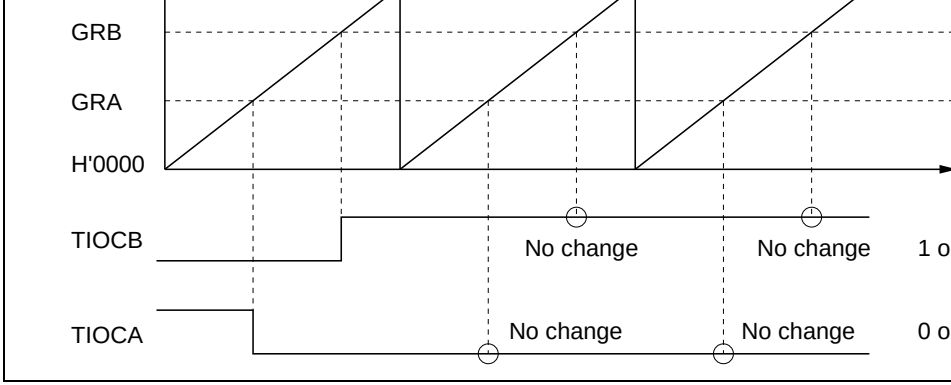


Figure 8.20 0 and 1 Output (Examples)

Figure 8.21 shows examples of toggle output. TCNT operates as a periodic counter, cle
compare match B. Toggle output is selected for both compare match A and B.

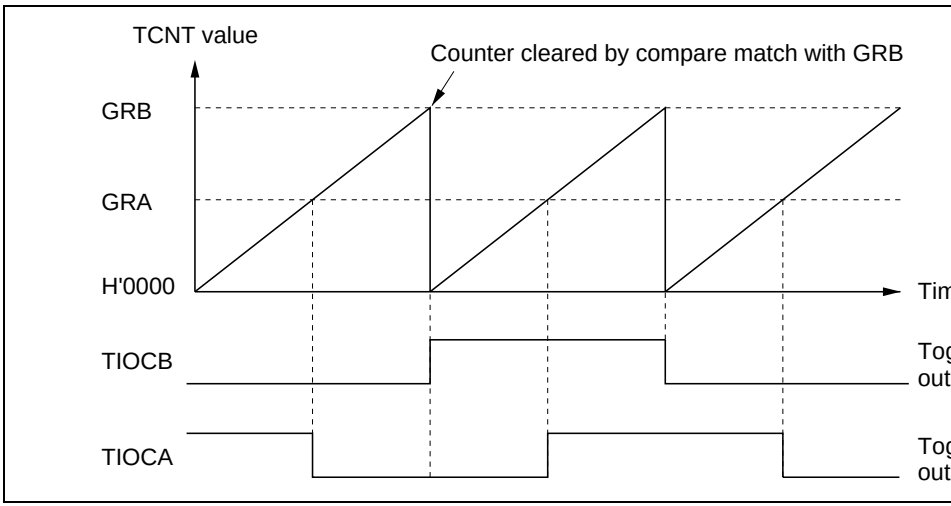


Figure 8.21 Toggle Output (Example)

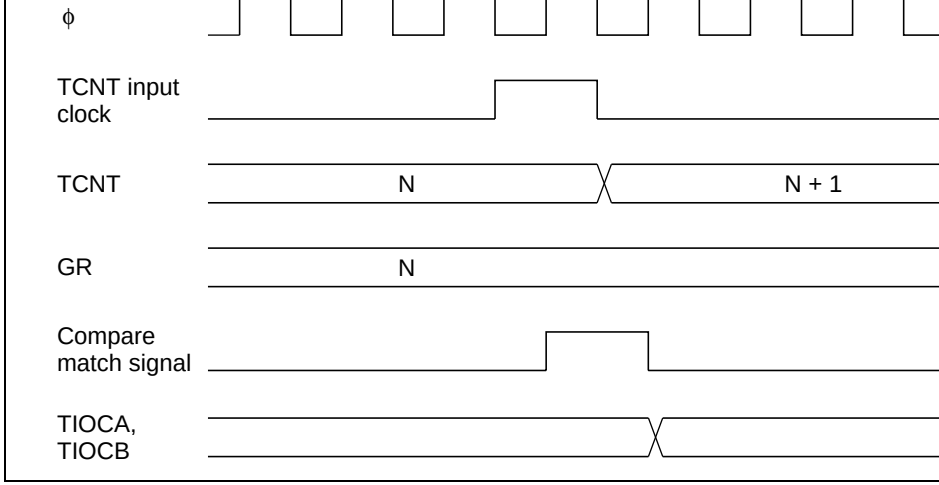


Figure 8.22 Output Compare Timing

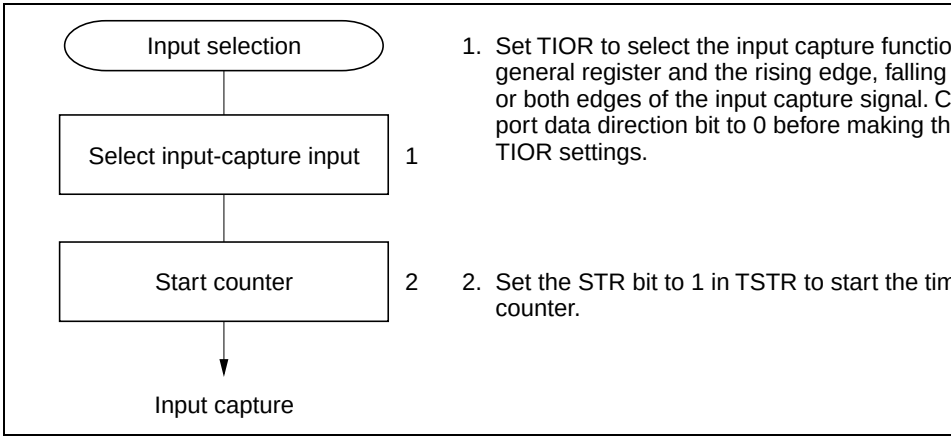


Figure 8.23 Setup Procedure for Input Capture (Example)

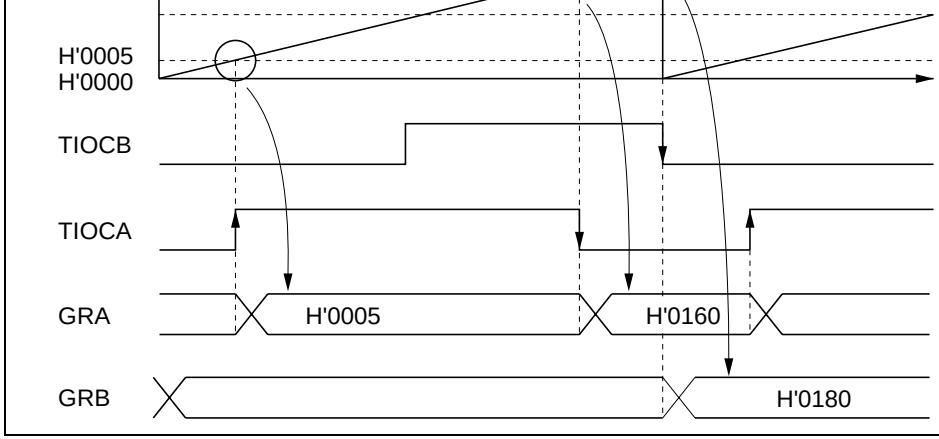


Figure 8.24 Input Capture (Example)

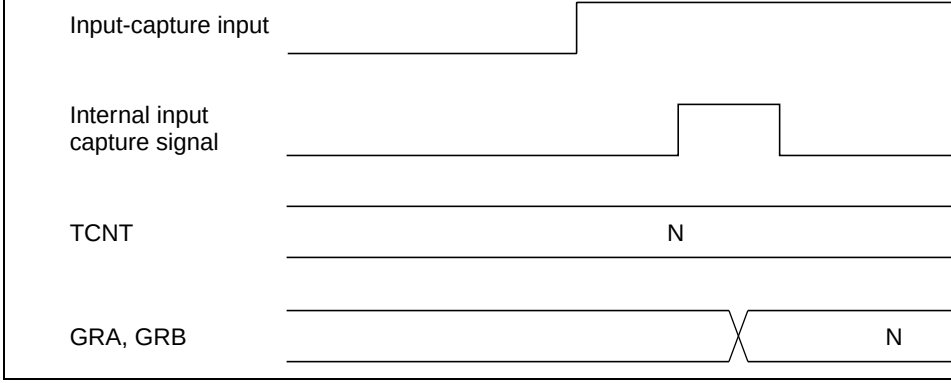
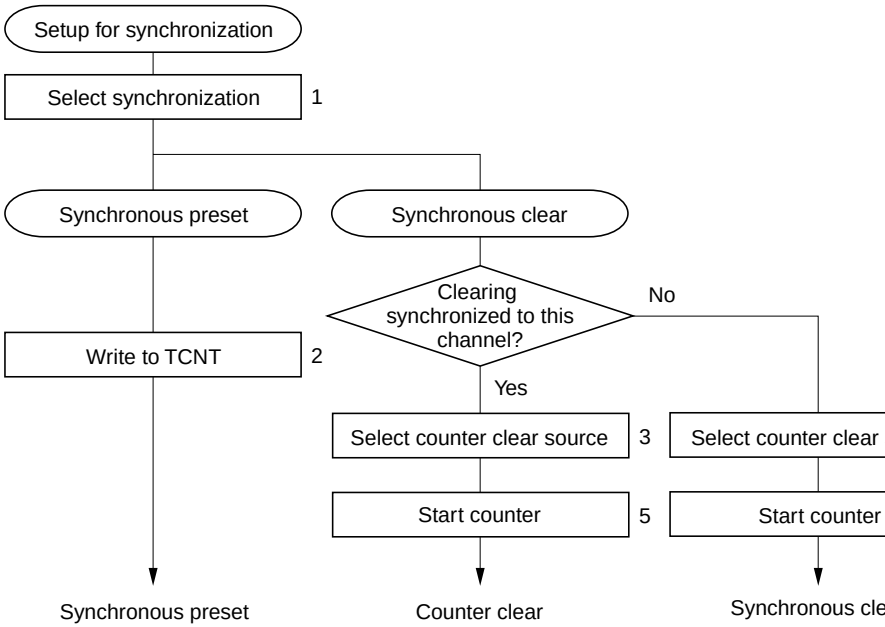


Figure 8.25 Input Capture Signal Timing

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 inpu
4. Set the CCLR1 and CCLR0 bits in TCR to have the counter cleared synchronously.
5. Set the STR bits in TSTR to 1 to start the synchronized counters.

Figure 8.26 Setup Procedure for Synchronization (Example)

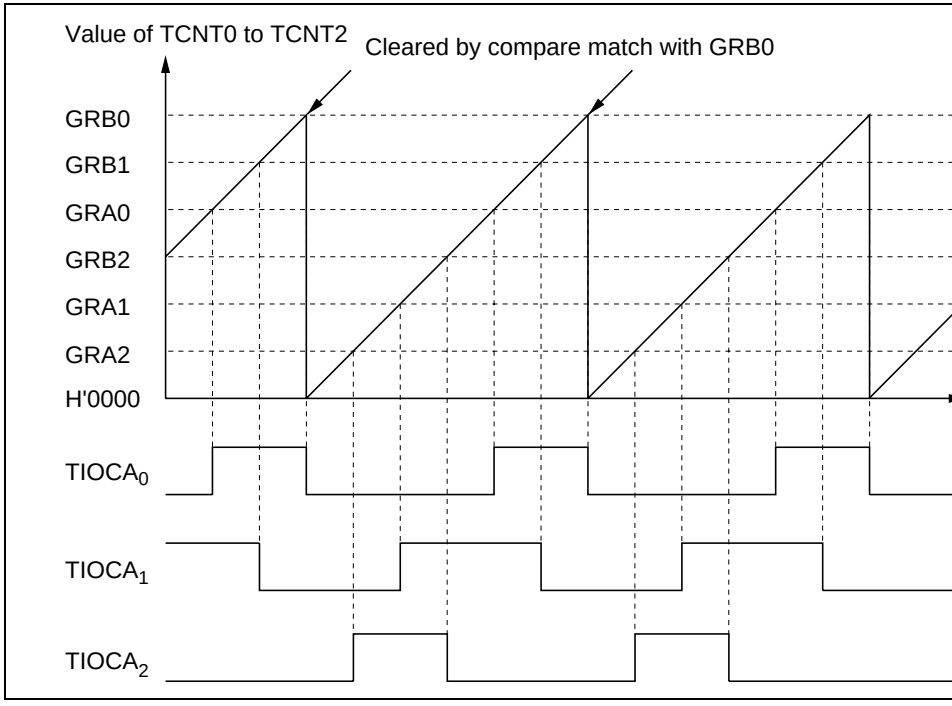
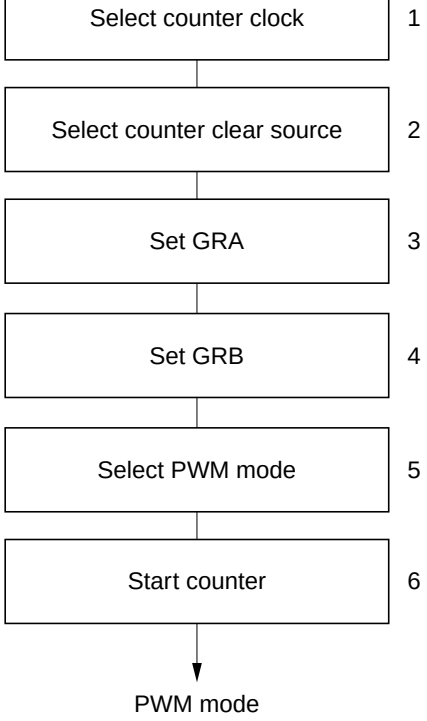


Figure 8.27 Synchronization (Example)

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



1. select the desired edge(s) of the external clock signal.
2. Set bits CCLR1 and CCLR0 in TC to select the counter clear source.
3. Set the time at which the PWM waveform should go to 1 in GRA.
4. Set the time at which the PWM waveform should go to 0 in GRB.
5. Set the PWM bit in TMDR to selec PWM mode. When PWM mode is selected, regardless of the TIOR contents, GRA and GRB become output compare registers specifyi the times at which the PWM goes to 1 and 0. The TIOCA pin automatically becomes the PWM output pin. The TIOCB pin conforn to the settings of bits IOB1 and IO in TIOR. If TIOCB output is not desired, clear both IOB1 and IOB
6. Set the STR bit to 1 in TSTR to st the timer counter.

Figure 8.28 Setup Procedure for PWM Mode (Example)

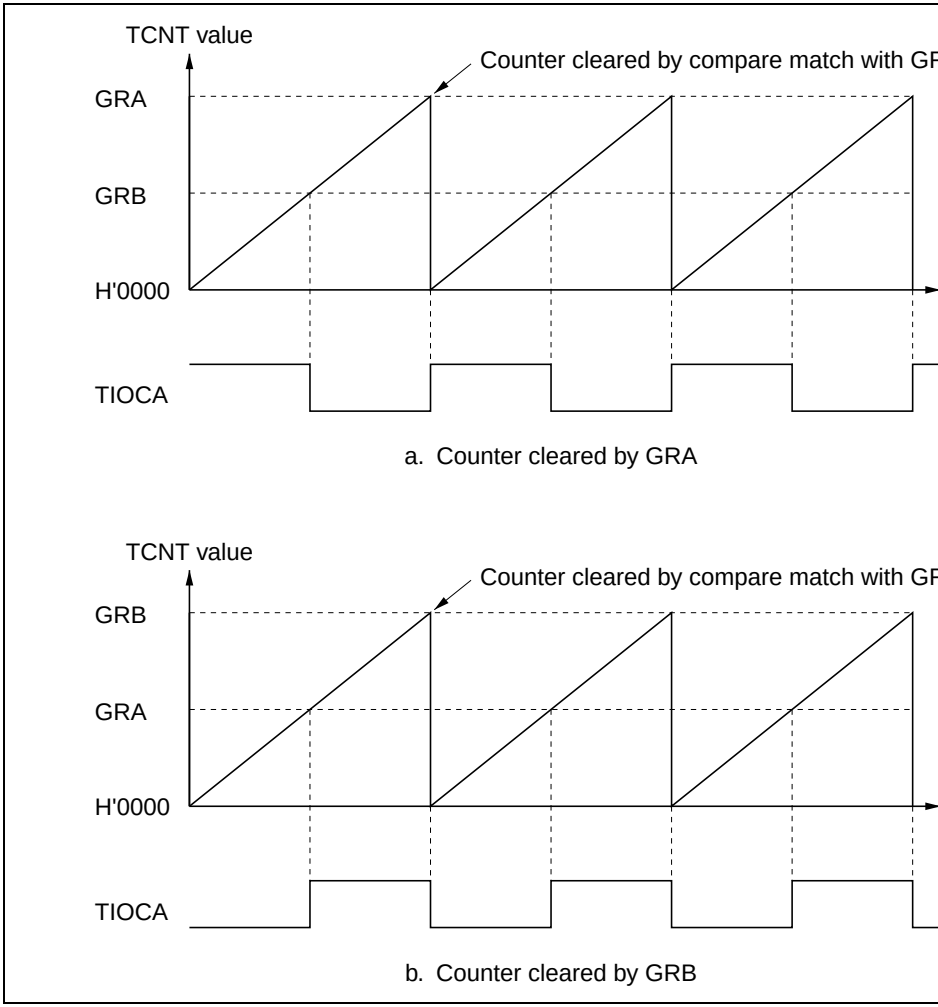


Figure 8.29 PWM Mode (Example 1)

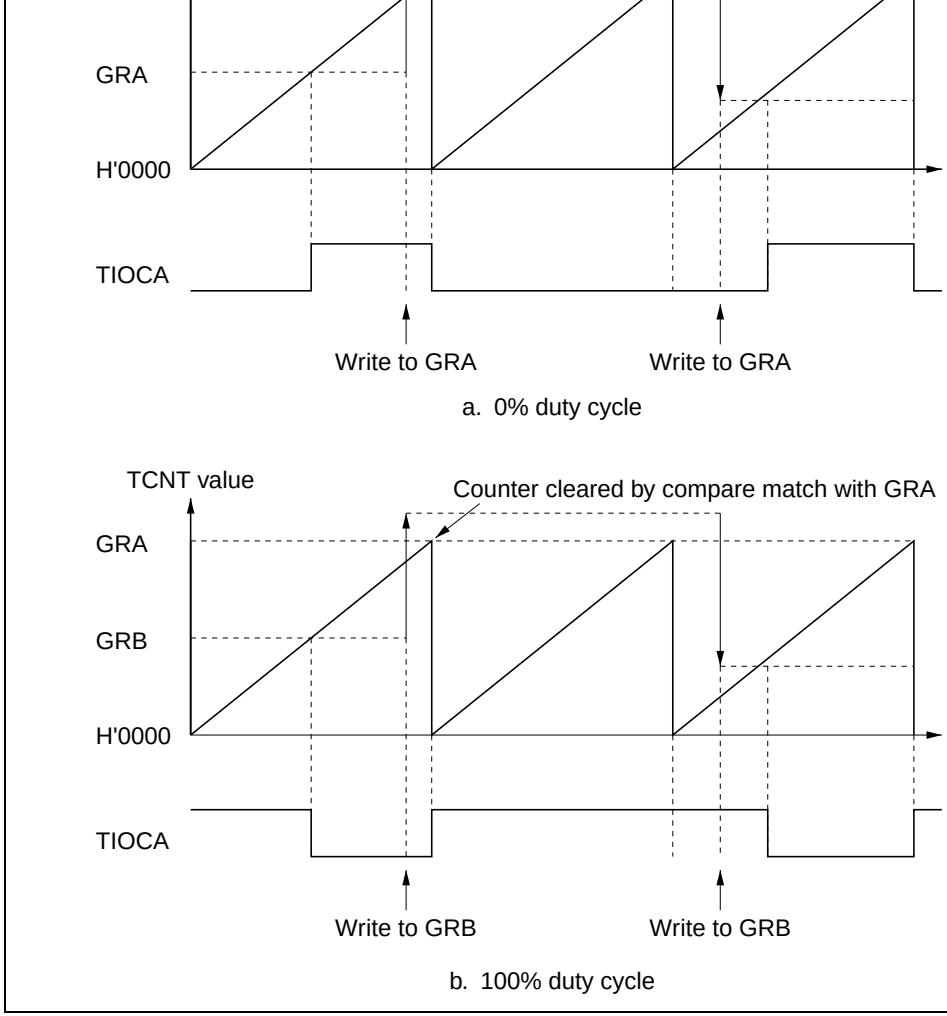


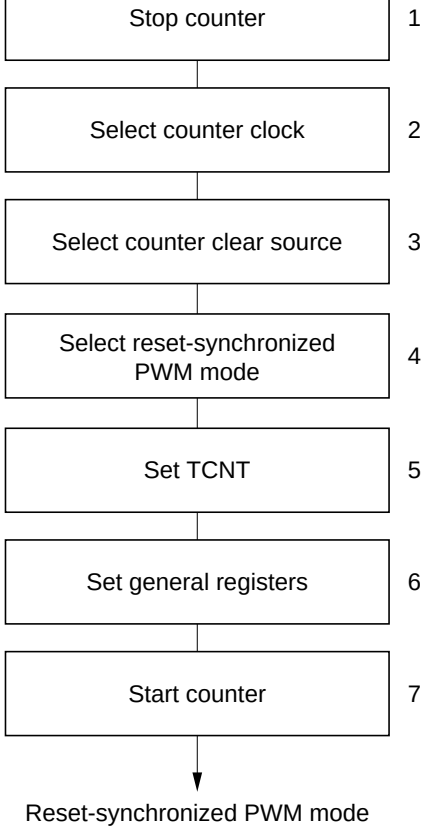
Figure 8.30 PWM Mode (Example 2)

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 outpu
4	$TIOCA_4$	PWM output 2
	$TOCXA_4$	PWM output 2' (complementary waveform to PWM outpu
	$TIOCB_4$	PWM output 3
	$TOCXB_4$	PWM output 3' (complementary waveform to PWM outpu

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 TI
GRA4	Specifies a transition point of PWM waveforms output from $TIOCA_4$ and TC
GRB4	Specifies a transition point of PWM waveforms output from $TIOCB_4$ and TC



TCNT3 is halted.

2. Set bits TPSC2 to TPSC0 in TCR select the counter clock source fo channel 3. If an external clock sou is selected, select the external clo edge(s) with bits CKEG1 and CKE in TCR.
3. Set bits CCLR1 and CCLR0 in TC to select GRA3 compare match as the counter clear source.
4. Set bits CMD1 and CMD0 in TFC select reset-synchronized PWM m $TIOCA_3$, $TIOCB_3$, $TIOCA_4$, TIOCE $TOCXA_4$, and $TOCXB_4$ automatic become PWM output pins.
5. Preset TCNT3 to H'0000. TCNT4 need not be preset.
6. GRA3 is the waveform period reg Set the waveform period value in GRA3. Set transition times of the PWM output waveforms in GRB3, GRA4, and GRB4. Set times withi the compare match range of TCN $X \leq GRA3$ (X: setting value)
7. Set the STR3 bit in TSTR to 1 to s TCNT3.

Figure 8.31 Setup Procedure for Reset-Synchronized PWM Mode (Exam

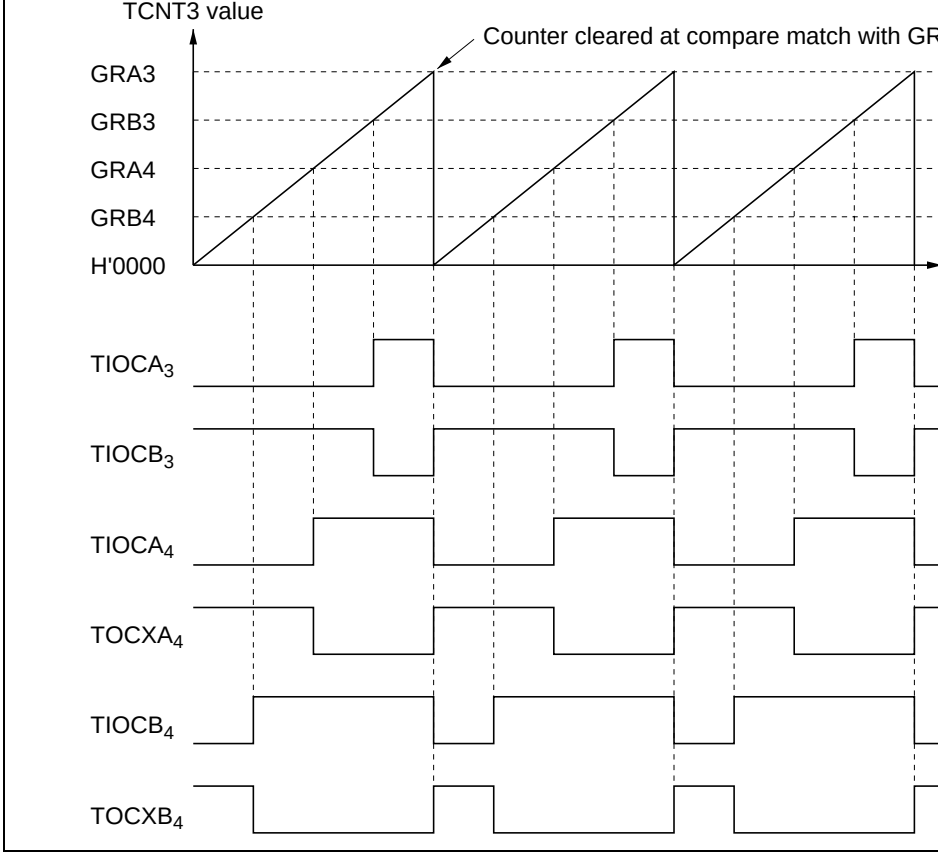


Figure 8.32 Operation in Reset-Synchronized PWM Mode (Example (when OLS3 = OLS4 = 1)

For the settings and operation when reset-synchronized PWM mode and buffer mode selected, see section 8.4.8, Buffering.

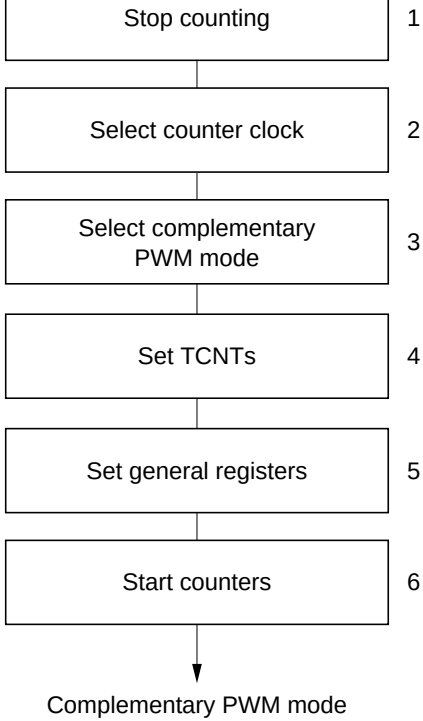
Table 8.7 lists the PWM output pins. Table 8.8 summarizes the register settings.

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 output 1)
4	$TIOCA_4$	PWM output 2
	$TOCXA_4$	PWM output 2' (non-overlapping complementary waveform output 2)
	$TIOCB_4$	PWM output 3
	$TOCXB_4$	PWM output 3' (non-overlapping complementary waveform 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 TIC
GRA4	Specifies a transition point of PWM waveforms output from $TIOCA_4$ and TO
GRB4	Specifies a transition point of PWM waveforms output from $TIOCB_4$ and TO

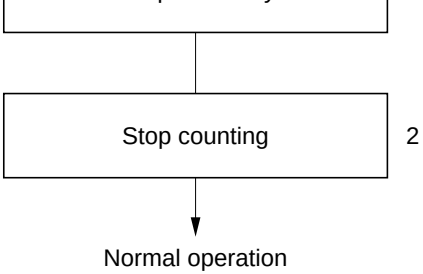


halted.

2. Set bits TPSC2 to TPSC0 in T select the same counter clock for channels 3 and 4. If an exte clock source is selected, selec external clock edge(s) with bits CKEG1 and CKEG0 in TCR. D select any counter clear sourc with bits CCLR1 and CCLR0 i
3. Set bits CMD1 and CMD0 in T to select complementary PWM $TIOCA_3$, $TIOCB_3$, $TIOCA_4$, TI $TOCXA_4$, and $TOCXB_4$ autom become PWM output pins.
4. Clear TCNT4 to H'0000. Set th non-overlap margin in TCNT3. set TCNT3 and TCNT4 to the value.
5. GRA3 is the waveform period register. Set the upper limit va TCNT3 minus 1 in GRA3. Set transition times of the PWM ou waveforms in GRB3, GRA4, a GRB4. Set times within the co match range of TCNT3 and TC $T \leq X$ (X: initial setting of GRB GRA4, or GRB4. T: initial setti TCNT3)
6. Set bits STR3 and STR4 in TS 1 to start TCNT3 and TCNT4.

Note: After exiting complementary PWM mode, to resume operating in complem PWM mode, follow the entire setup procedure from step 1 again.

Figure 8.33 Setup Procedure for Complementary PWM Mode (Examp



mode.

2. After setting channels 3 and 4 to
operating mode, wait at least one
count before clearing bits STR3 a
STR4 of TSTR to 0 to stop the co
operation of TCNT3 and TCNT4.

Figure 8.34 Clearing Procedure for Complementary PWM Mode (Exam

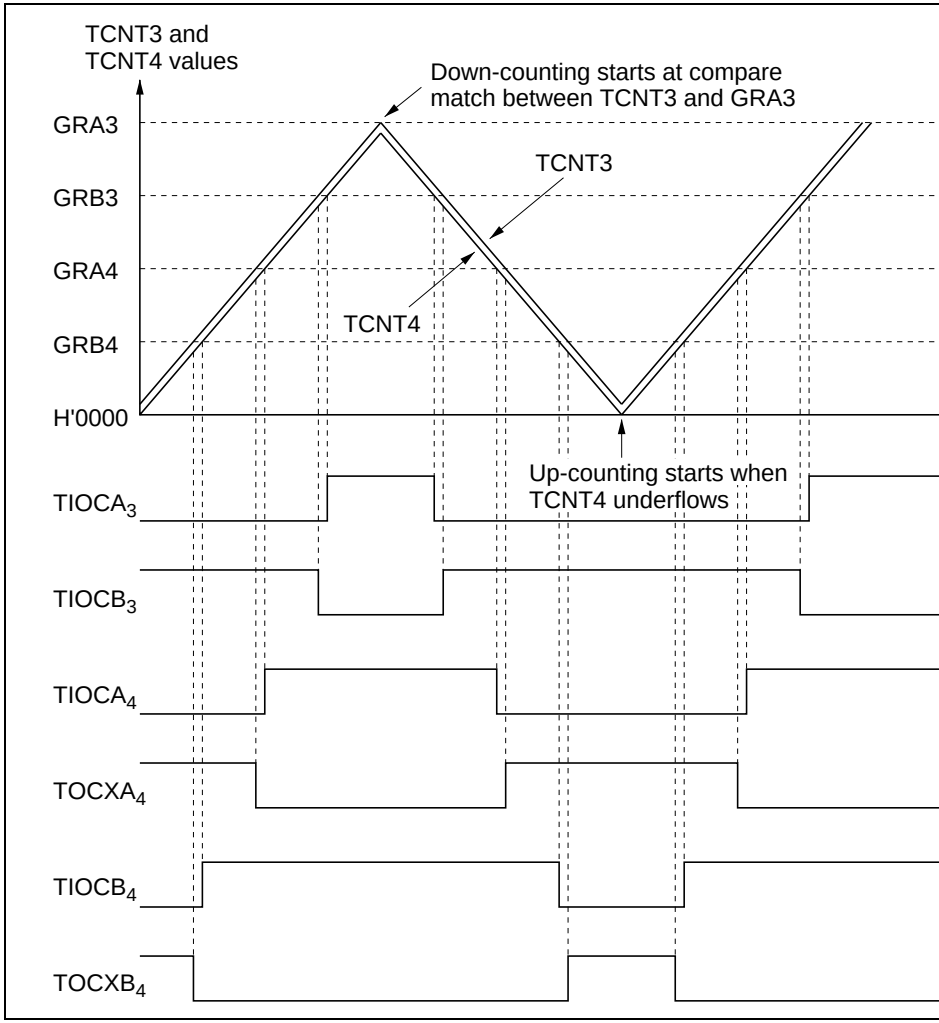
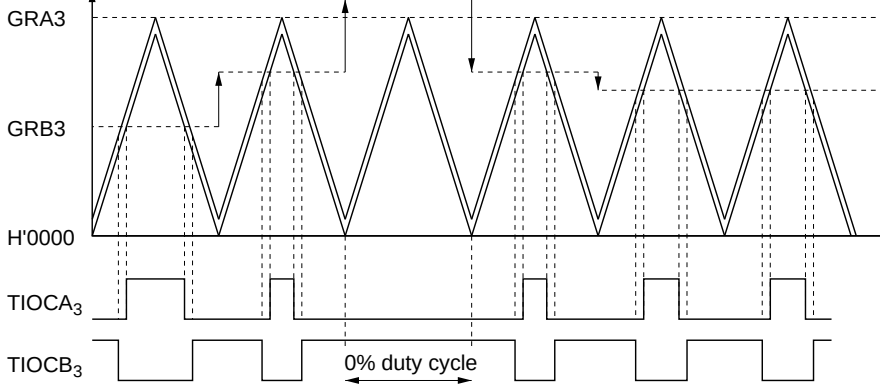
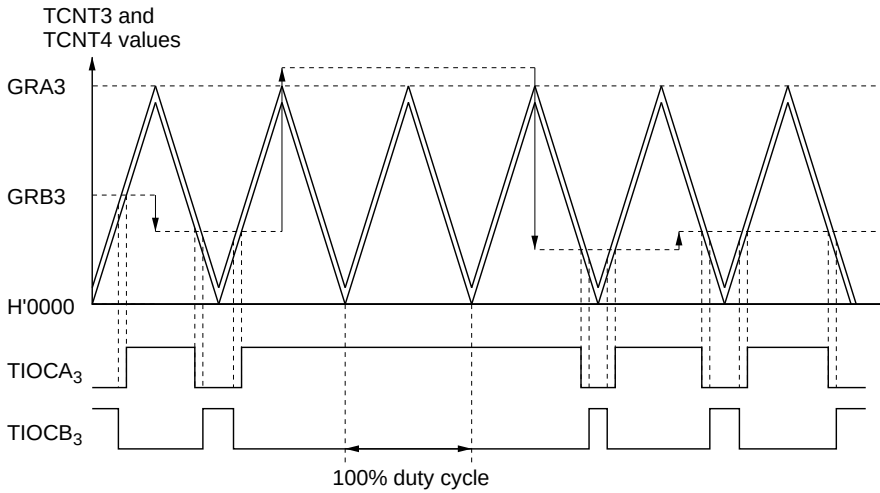


Figure 8.35 Operation in Complementary PWM Mode (Example 1) (when OLS3 = OLS4 = 1)



a. 0% duty cycle



b. 100% duty cycle

Figure 8.36 Operation in Complementary PWM Mode (Example 2) (when OLS3 = OLS4 = 1)

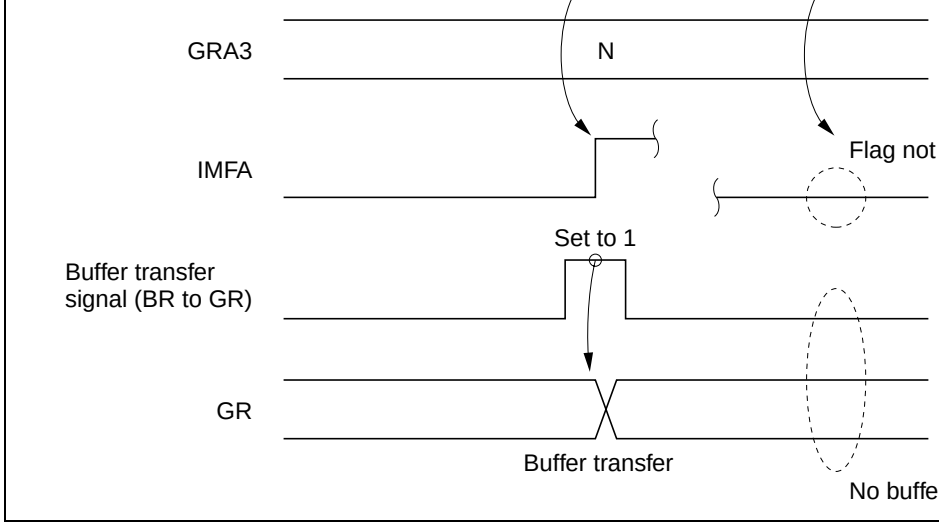


Figure 8.37 Overshoot Timing

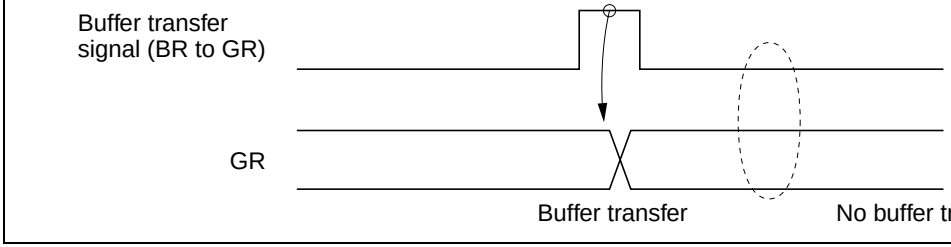


Figure 8.38 Undershoot Timing

In channel 3, IMFA is set to 1 only during up-counting. In channel 4, OVF is set to 1 o an underflow occurs. When buffering is selected, buffer register contents are transferre general register at compare match A3 during up-counting, and when TCNT4 underflow

General Register Settings in Complementary PWM Mode

When setting up general registers for complementary PWM mode or changing their set during operation, note the following points.

- Initial settings

  Do not set values from H'0000 to T – 1 (where T is the initial value of TCNT3). Af counters start and the first compare match A3 event has occurred, however, settings range also become possible.
- Changing settings

  Use the buffer registers. Correct waveform output may not be obtained if a general written to directly.
- Cautions on changes of general register settings

  Figure 8.39 shows six correct examples and one incorrect example.

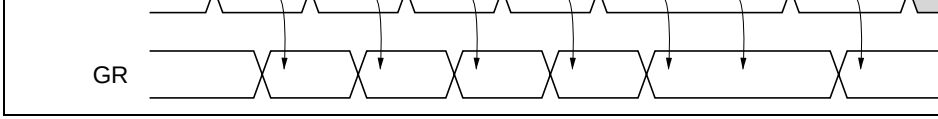


Figure 8.39 Changing a General Register Setting by Buffer Transfer (Exa

— Buffer transfer at transition from up-counting to down-counting

If the general register value is in the range from GRA3 – T + 1 to GRA3, do no buffer register value outside this range. Conversely, if the general register valu this range, do not transfer a value within this range. See figure 8.40.

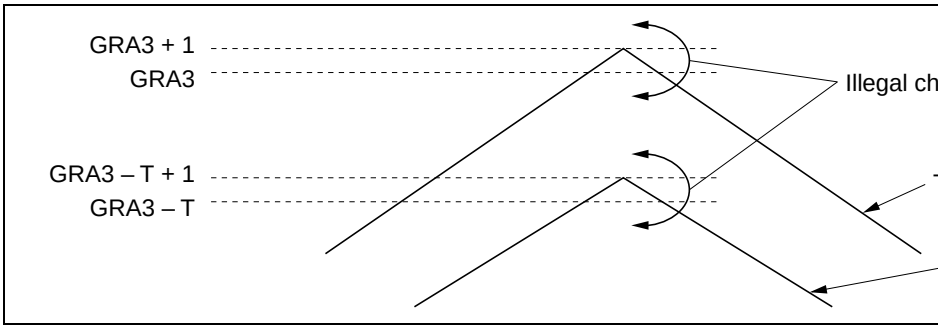


Figure 8.40 Changing a General Register Setting by Buffer Transfer (Cau

— Buffer transfer at transition from down-counting to up-counting

If the general register value is in the range from H'0000 to T – 1, do not transfe register value outside this range. Conversely, when a general register value is c range, do not transfer a value within this range. See figure 8.41.

H'FFFF

Figure 8.41 Changing a General Register Setting by Buffer Transfer (Cauti

— General register settings outside the counting range (H'0000 to GRA3)

Waveforms with a duty cycle of 0% or 100% can be output by setting a general a value outside the counting range. When a buffer register is set to a value outsi counting range, then later restored to a value within the counting range, the cou direction (up or down) must be the same both times. See figure 8.42.

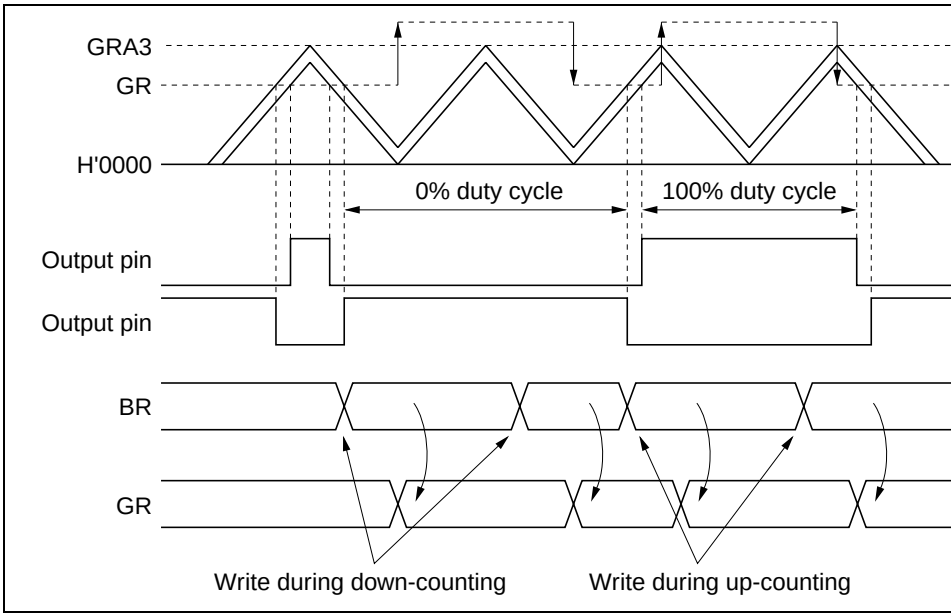


Figure 8.42 Changing a General Register Setting by Buffer Transfer (Exam

Settings can be made in this way by detecting GRA3 compare match or TCNT4 und before writing to the buffer register.

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.

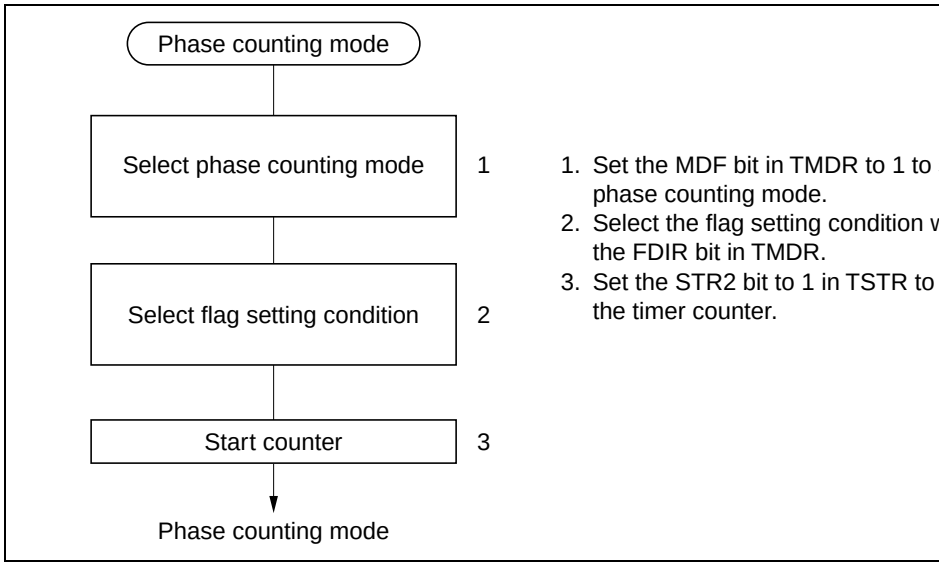


Figure 8.43 Setup Procedure for Phase Counting Mode (Example)

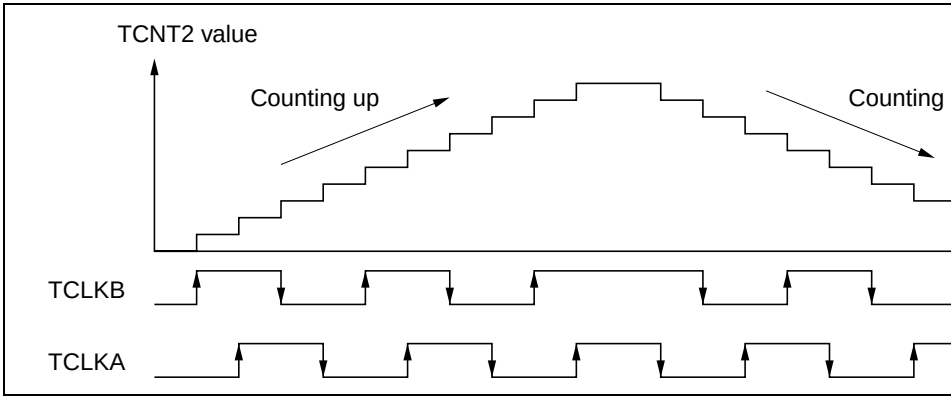


Figure 8.44 Operation in Phase Counting Mode (Example)

Table 8.9 Up/Down Counting Conditions

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

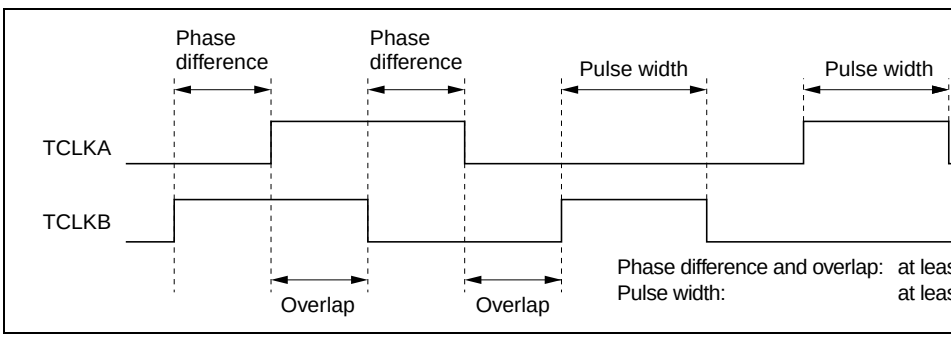


Figure 8.45 Phase Difference, Overlap, and Pulse Width in Phase Counting

The buffer register value is transferred to the general register at compare match. See figure 8.46.

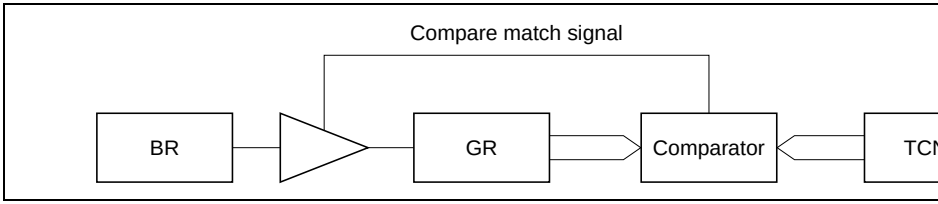


Figure 8.46 Compare Match Buffering

- General register used for input capture

  The TCNT value is transferred to the general register at input capture. The previou register value is transferred to the buffer register.
  See figure 8.47.

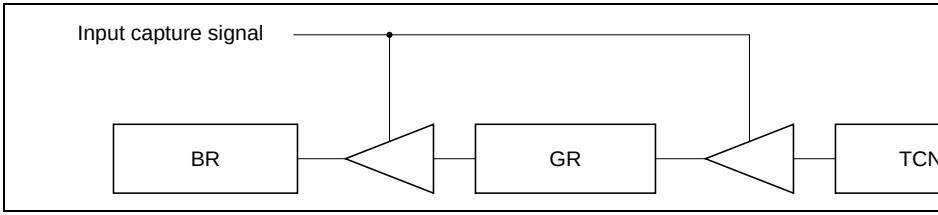


Figure 8.47 Input Capture Buffering

- Complementary PWM mode

  The buffer register value is transferred to the general register when TCNT3 and TC 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

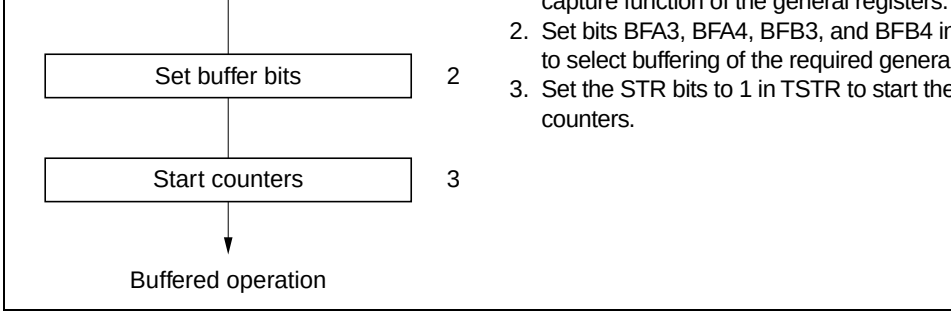


Figure 8.48 Buffering Setup Procedure (Example)

Examples of Buffering

Figure 8.49 shows an example in which GRA is set to function as an output compare re
buffered by BRA, TCNT is set to operate as a periodic counter cleared by GRB compar
and TIOCA and TIOCB are set to toggle at compare match A and B. Because of the bu
when TIOCA toggles at compare match A, the BRA value is simultaneously transferre
This operation is repeated each time compare match A occurs. Figure 8.50 shows the tr
timing.

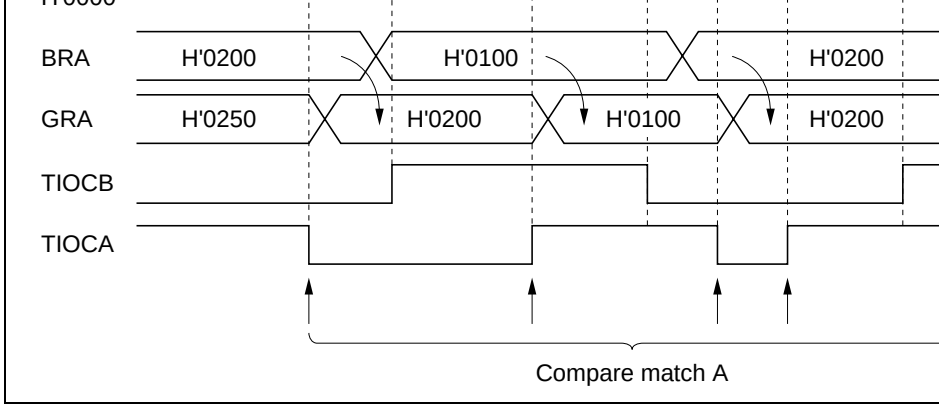


Figure 8.49 Register Buffering (Example 1: Buffering of Output Compare F

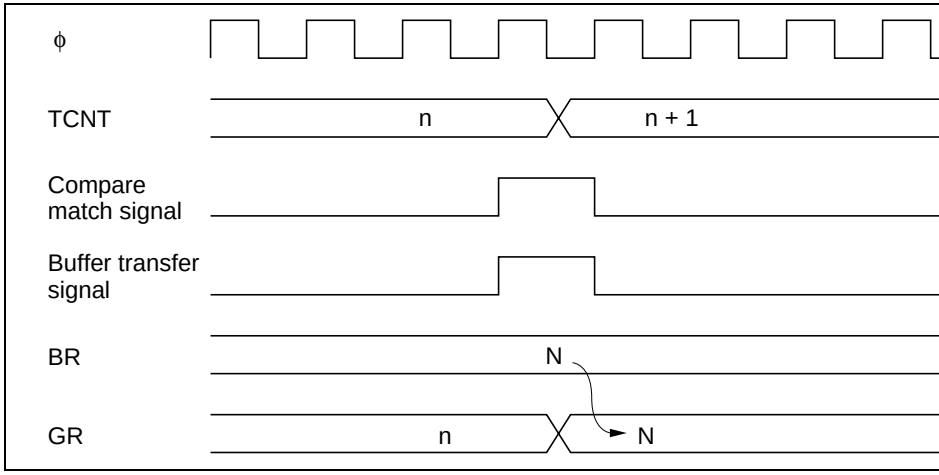


Figure 8.50 Compare Match and Buffer Transfer Timing (Example)

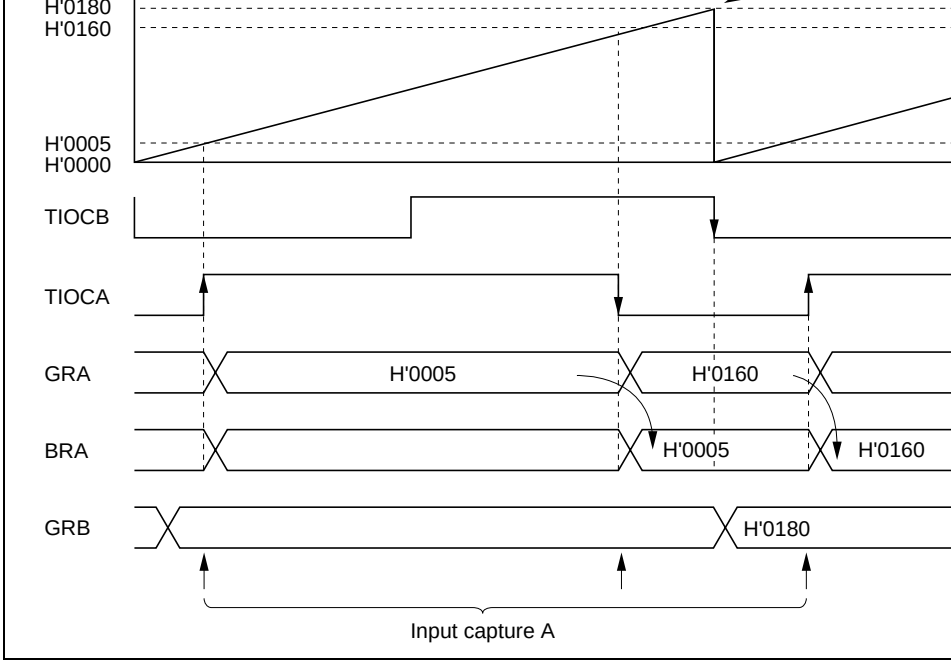


Figure 8.51 Register Buffering (Example 2: Buffering of Input Capture Reg

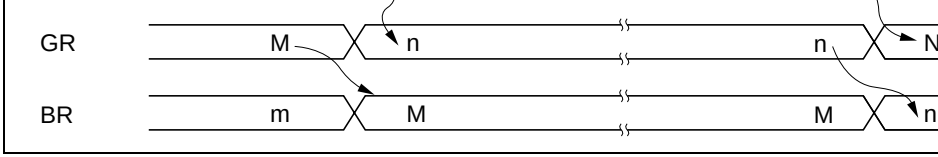


Figure 8.52 Input Capture and Buffer Transfer Timing (Example)

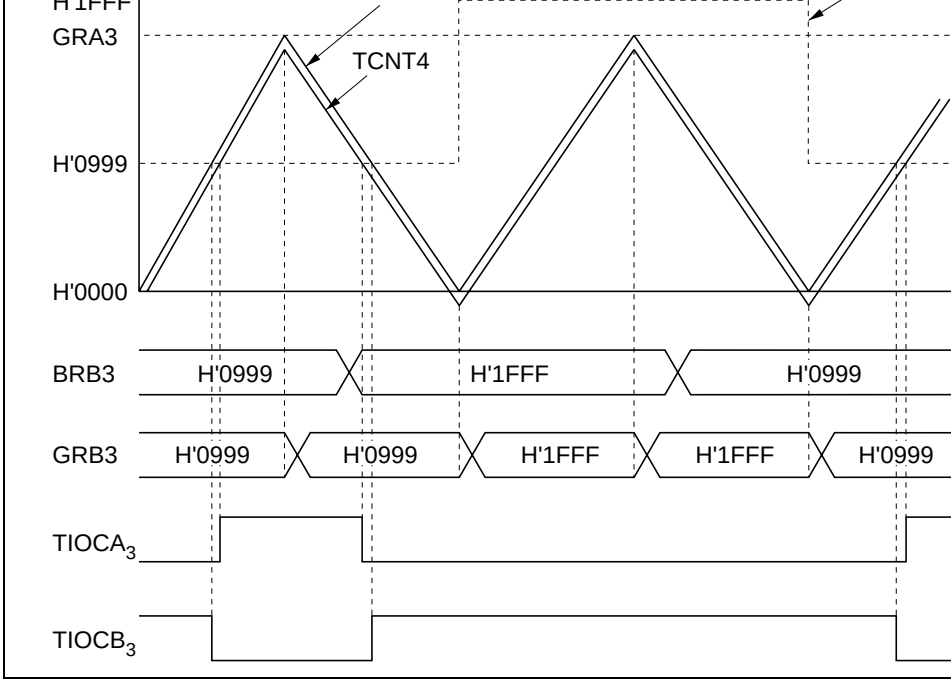


Figure 8.53 Register Buffering (Example 4: Buffering in Complementary PWM

arbitrary value can be output by appropriate settings of the data register (DR) and data
register (DDR) of the corresponding input/output port. Figure 8.54 illustrates the timin
enabling and disabling of ITU output by TOER.

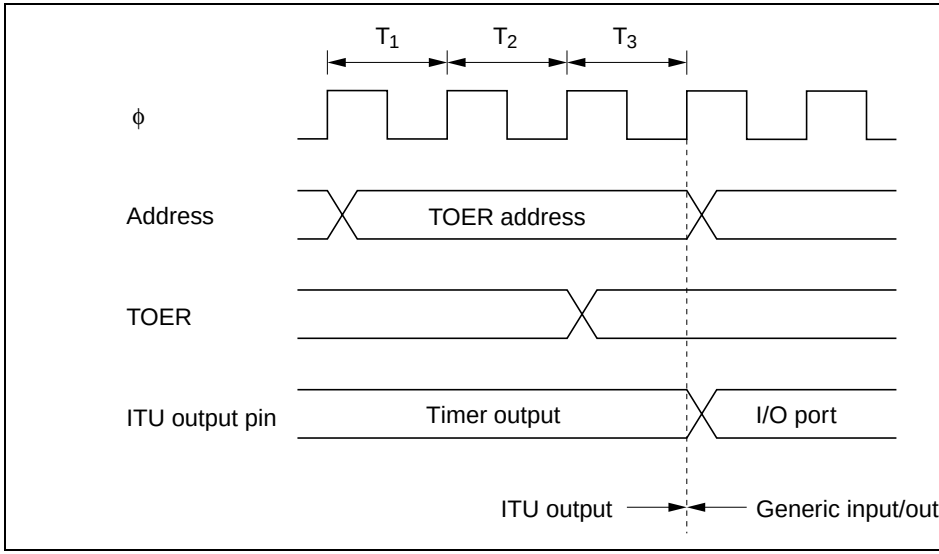
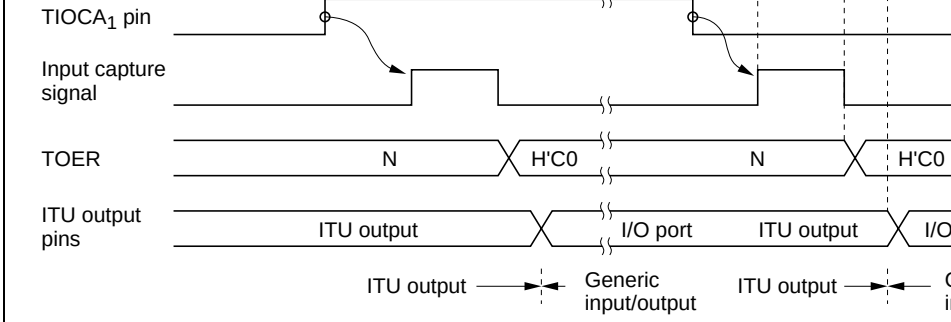


Figure 8.54 Timing of Disabling of ITU Output by Writing to TOER (Exa



Legend:
N: Arbitrary setting (H'C1 to H'FF)

Figure 8.55 Timing of Disabling of ITU Output by External Trigger (Exan

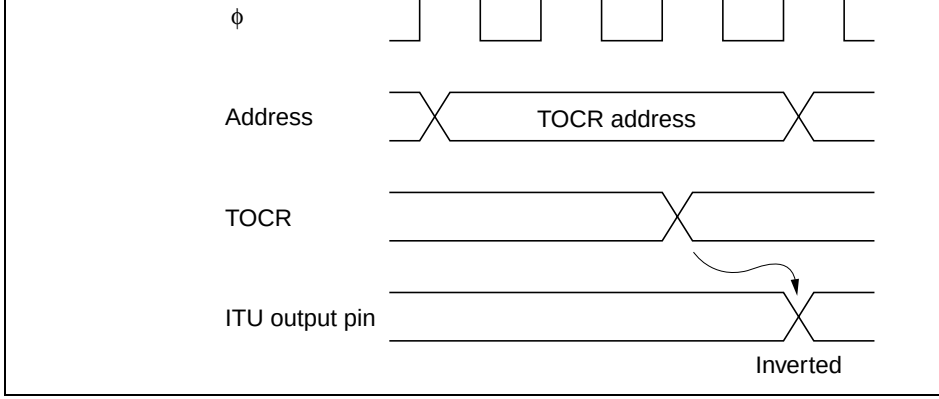


Figure 8.56 Timing of Inverting of ITU Output Level by Writing to TOCR (I

IMFA and IMFB are set to 1 by a compare match signal generated when TCNT matche
register (GR). The compare match signal is generated in the last state in which the valu
(when TCNT is updated from the matching count to the next count). Therefore, when T
matches a general register, the compare match signal is not generated until the next tim
input. Figure 8.57 shows the timing of the setting of IMFA and IMFB.

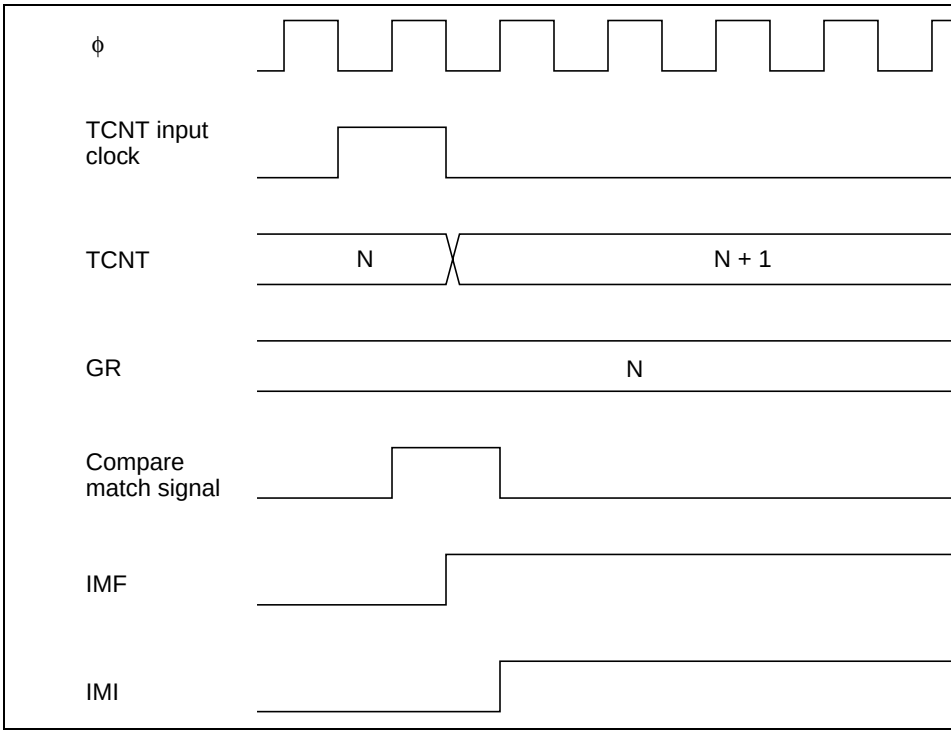


Figure 8.57 Timing of Setting of IMFA and IMFB by Compare Match

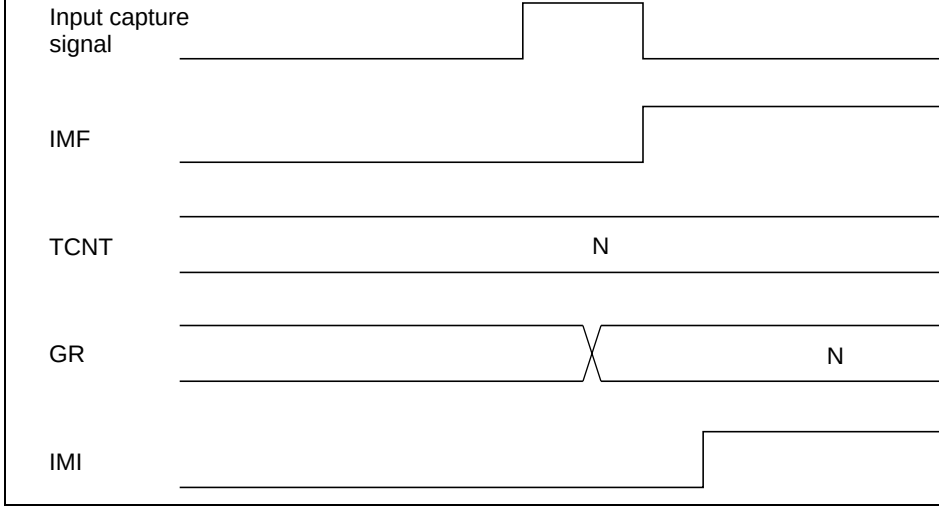


Figure 8.58 Timing of Setting of IMFA and IMFB by Input Capture

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
H'FFFF. Figure 8.59 shows the timing.

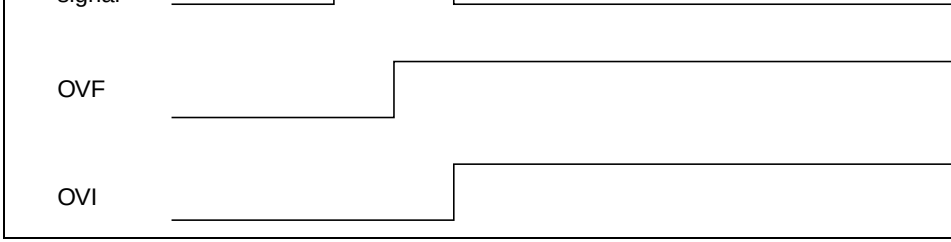


Figure 8.59 Timing of Setting of OVF

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 stat
cleared. Figure 8.60 shows the timing.

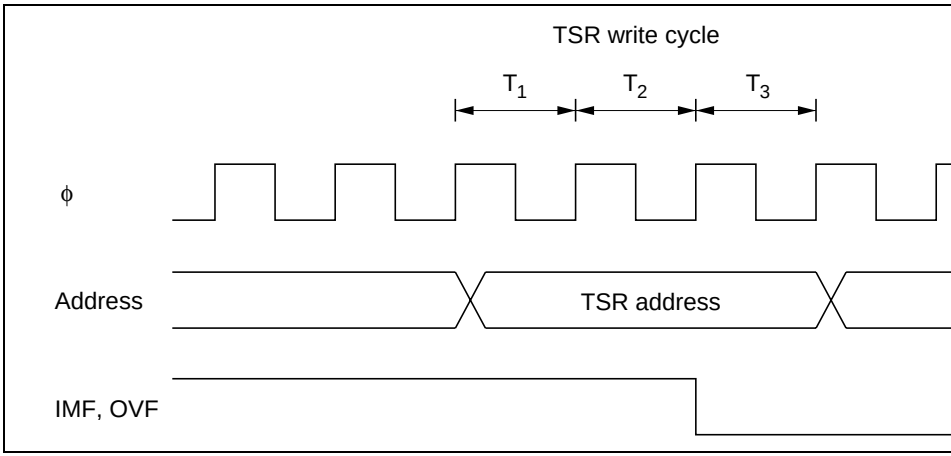


Figure 8.60 Timing of Clearing of Status Flags

Table 8.10 lists the interrupt sources.

Table 8.10 ITU Interrupt Sources

Channel	Interrupt Source	Description	Priorit
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 ca changed by settings in IPRA and IPRB.

priority and the write is not performed. See figure 8.61.

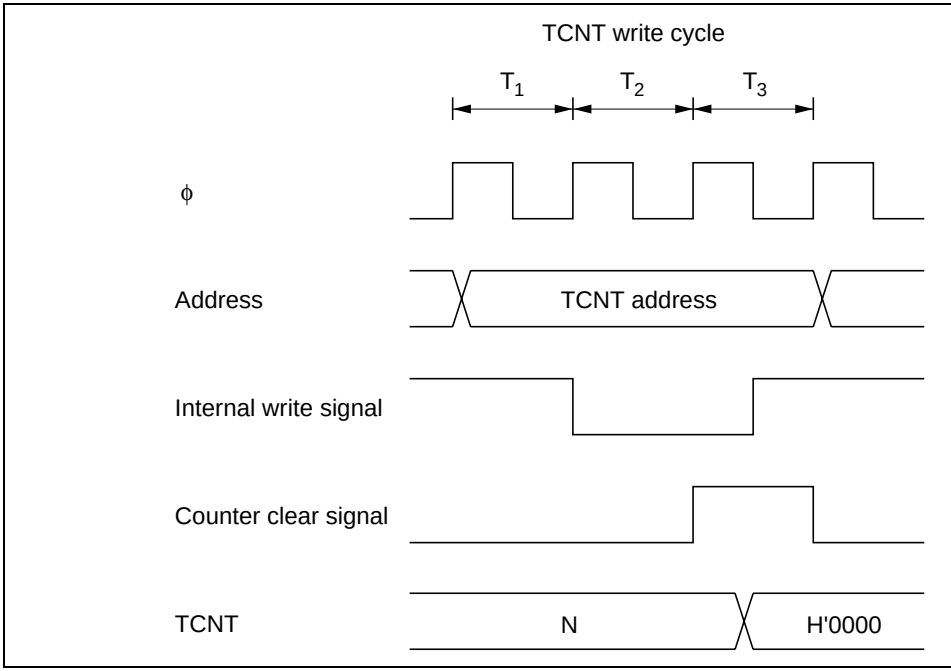


Figure 8.61 Contention between TCNT Write and Clear

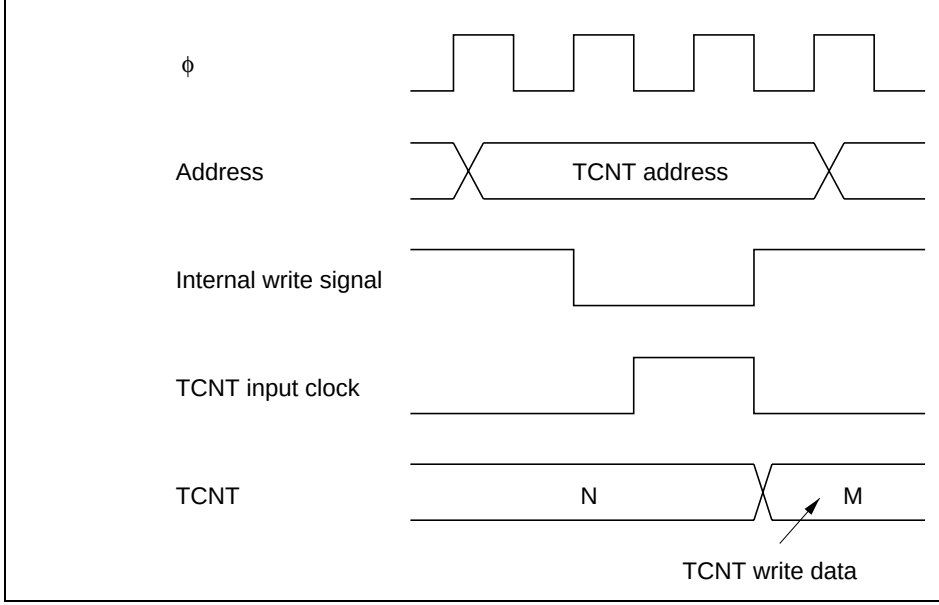


Figure 8.62 Contention between TCNT Word Write and Increment

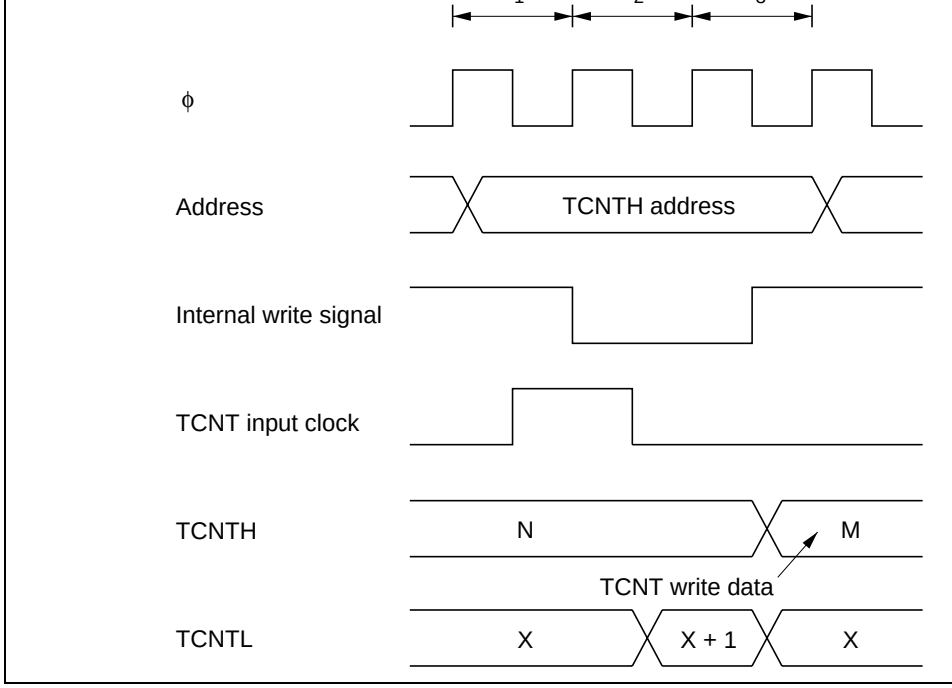


Figure 8.63 Contention between TCNT Byte Write and Increment

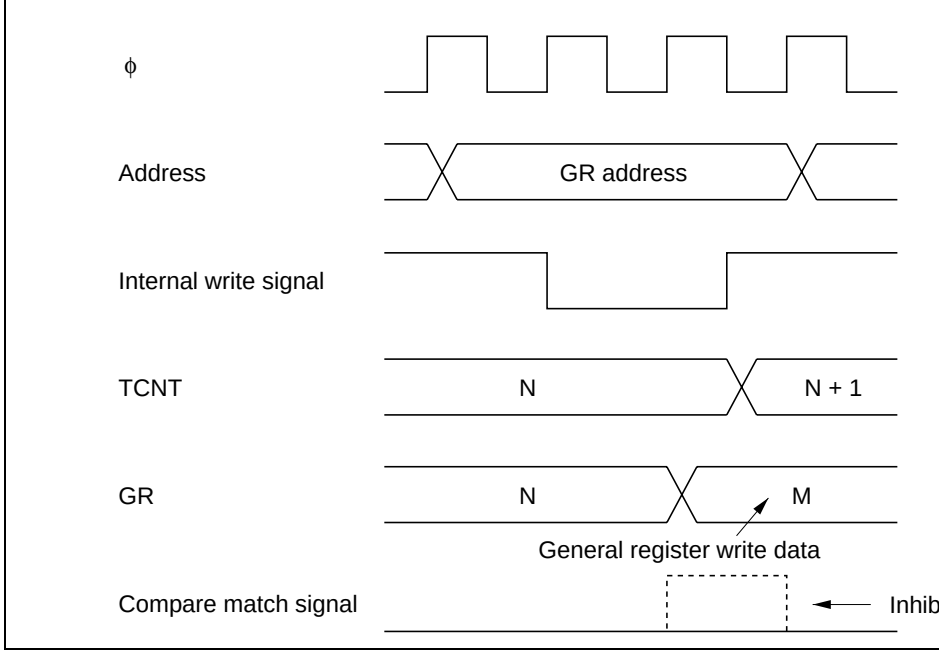


Figure 8.64 Contention between General Register Write and Compare M

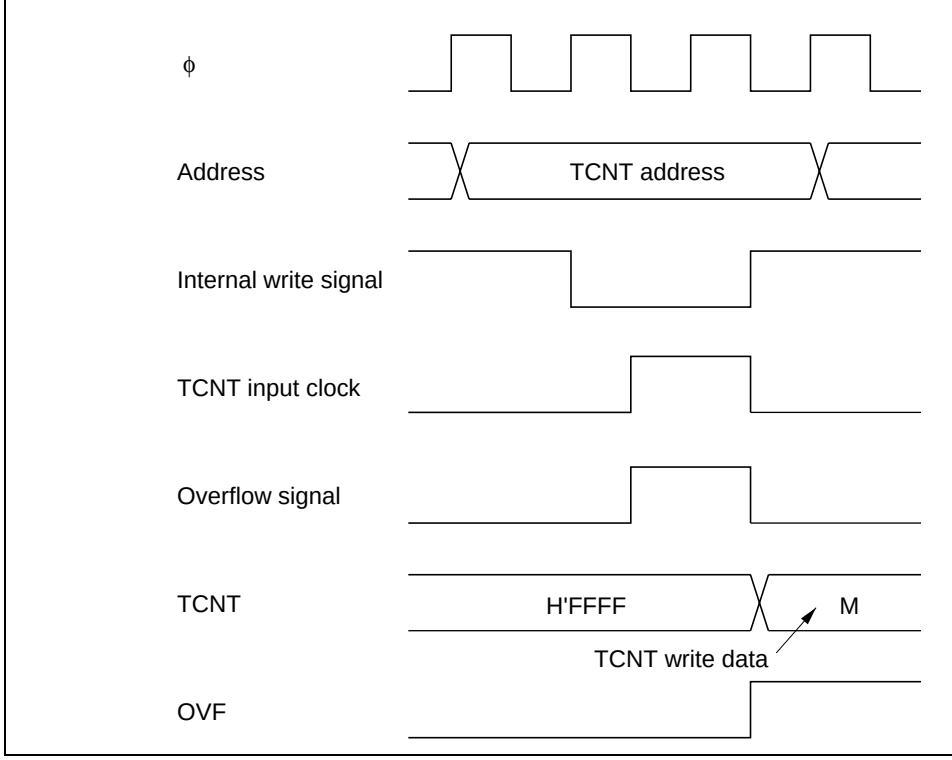


Figure 8.65 Contention between TCNT Write and Overflow

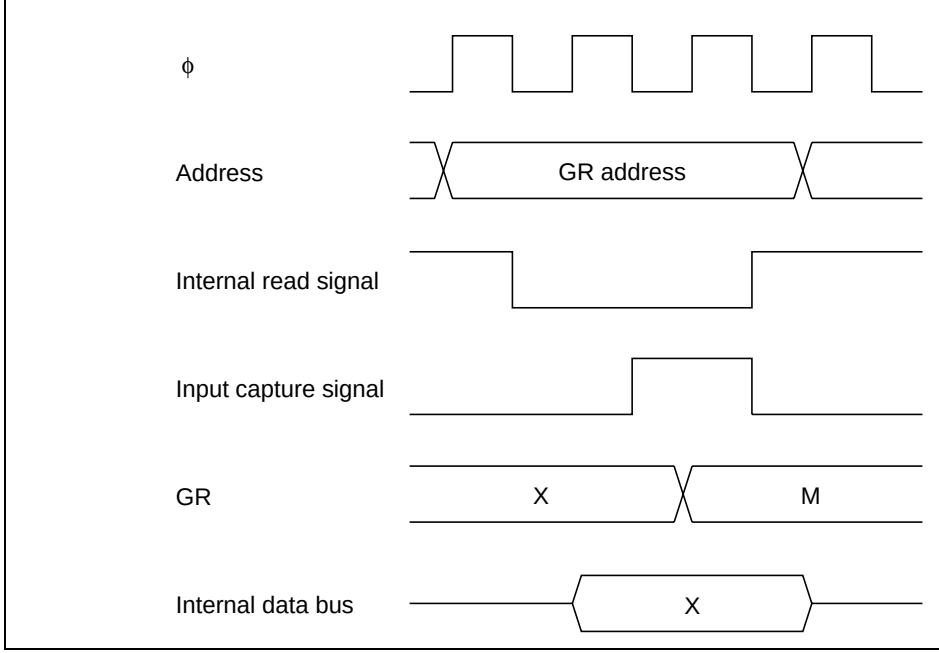


Figure 8.66 Contention between General Register Read and Input Capt

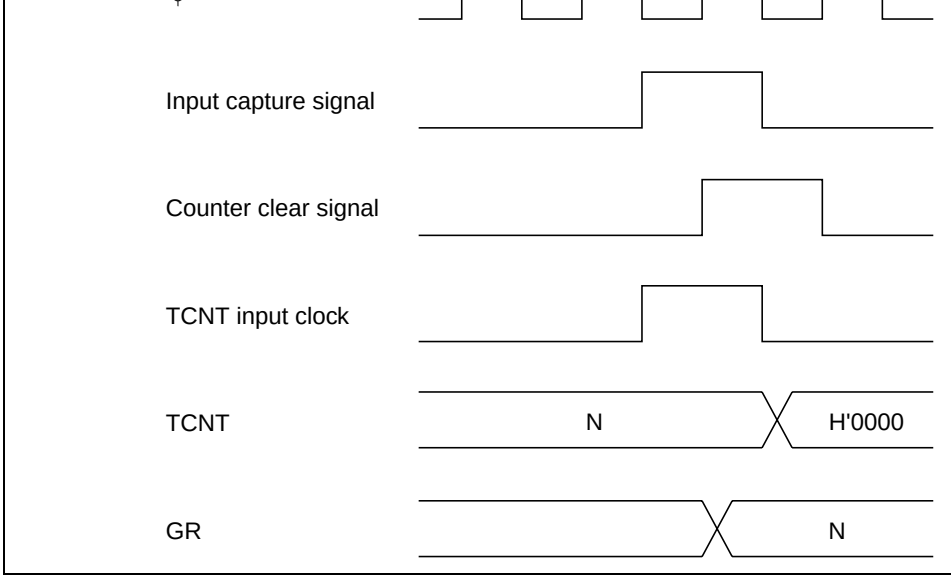


Figure 8.67 Contention between Counter Clearing by Input Capture an Counter Increment

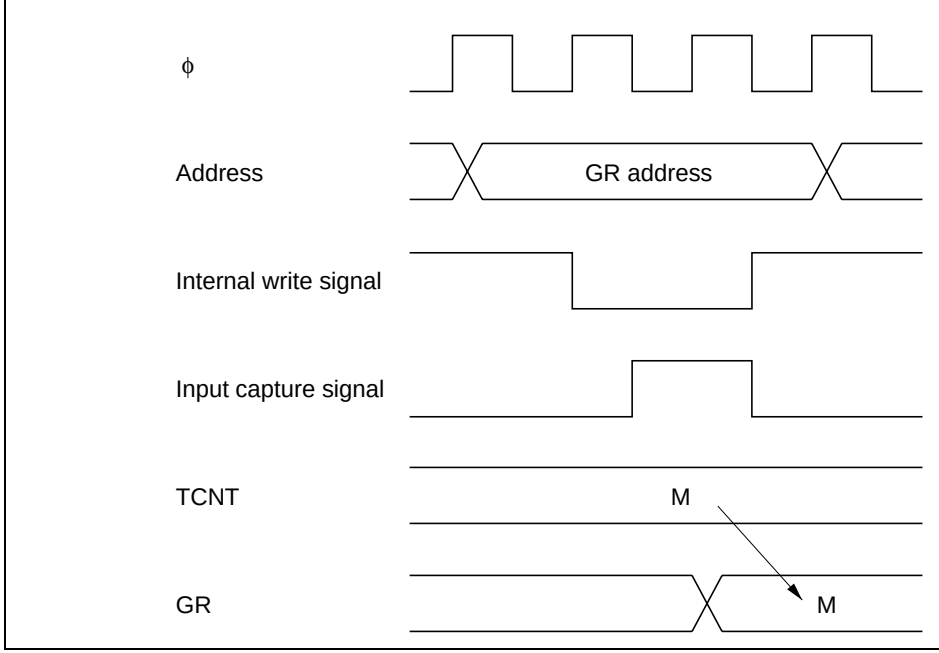


Figure 8.68 Contention between General Register Write and Input Cap

Note on Waveform Period Setting

When a counter is cleared by compare match, the counter is cleared in the last state at TCNT value matches the general register value, at the time when this value would nor updated to the next count. The actual counter frequency is therefore given by the follo formula:

$$f = \frac{\phi}{(N + 1)}$$

(f: counter frequency. ϕ: system clock frequency. N: value set in general regi

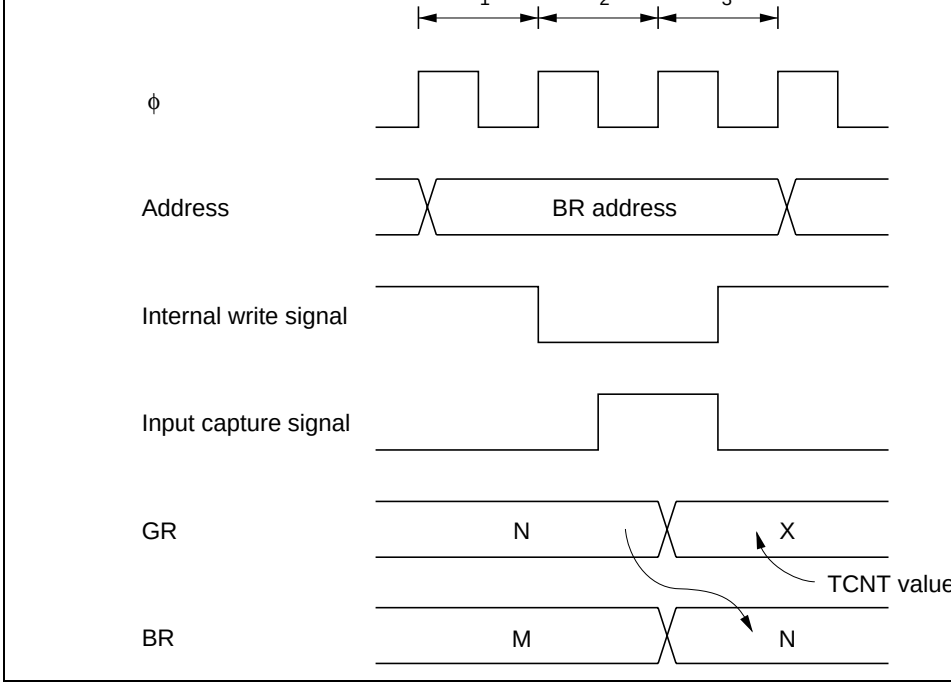
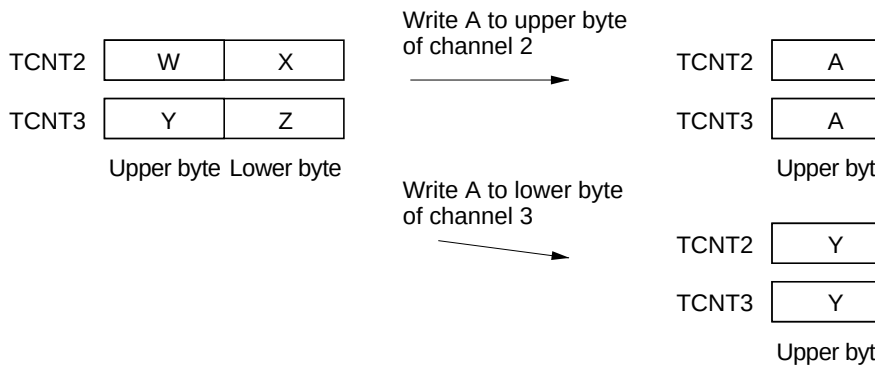
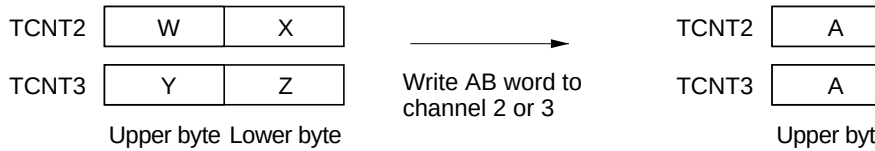


Figure 8.69 Contention between Buffer Register Write and Input Captu



- Word write to channel 2 or word write to channel 3



Note on Setup of Reset-Synchronized PWM Mode and Complementary PWM M

When setting bits CMD1 and CMD0 in TFCR, take the following precautions:

- Write to bits CMD1 and CMD0 only when TCNT3 and TCNT4 are stopped.
- Do not switch directly between reset-synchronized PWM mode and complementar mode. First switch to normal mode (by clearing bit CMD1 to 0), then select reset-s PWM mode or complementary PWM mode.

Operating Mode		Register Settings													
		TSNC	TMDR			TFCR			TOCR		TOER	TIOR0			
		Synchro-nization	MDF	FDIR	PWM	Comple-mentary PWM	Reset-Synchro-nized PWM	Buffer-ing	XTGD	Output Level Select	Master Enable	IOA	IOB		
Synchronous preset		SYNC0 = 1	—	—	○	—	—	—	—	—	—	○	○		
PWM mode		○	—	—	PWM0 = 1	—	—	—	—	—	—	—	○*		
Output compare A		○	—	—	PWM0 = 0	—	—	—	—	—	—	IOA2 = 0 Other bits unrestricted	○		
Output compare B		○	—	—	○	—	—	—	—	—	—	○	IOB2 = 0 Other bits 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	○	—	—	○	—	—	—	—	—	—	○	○		
	By compare match/input capture B	○	—	—	○	—	—	—	—	—	—	○	○		
	Syn-chronous clear	SYNC0 = 1	—	—	○	—	—	—	—	—	—	○	○		

Legend: ○: Setting available (valid). —: Setting does not affect this mode.

Note: The input capture function cannot be used in PWM mode. If compare match A and compare match B occur simultaneously, the compare match

Operating Mode		Register Settings												
		TSNC	TMDR			TFCR			TOCR		TOER	TIOR1		
		Synchro-nization	MDF	FDIR	PWM	Comple-mentary PWM	Reset-Synchro-nized PWM	Buffer-ing	XTGD	Output Level Select	Master Enable	IOA	IOB	
Synchronous preset		SYNC1 = 1	—	—	◯	—	—	—	—	—	—	◯		
PWM mode		◯	—	—	PWM1 = 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	◯	—	—	◯	—	—	—	—	—	—	◯		
	By compare match/input capture B	◯	—	—	◯	—	—	—	—	—	—	◯		
	Syn-chronous clear	SYNC1 = 1	—	—	◯	—	—	—	—	—	—	◯		

Legend: ◯: Setting available (valid). —: Setting does not affect this mode.

Notes: 1. The input capture function cannot be used in PWM mode. If compare match A and compare match B occur simultaneously, the comp
2. Valid only when channels 3 and 4 are operating in complementary PWM mode or reset-synchronized PWM mode.

RENESAS

Operating Mode		Register Settings											
		TSNC	TMDR			TFCR			TOCR		TOER	TIOR2	
		Synchro-nization	MDF	FDIR	PWM	Comple-mentary PWM	Reset-Synchro-nized PWM	Buffer-ing	XTGD	Output Level Select	Master Enable	IOA	I
Synchronous preset		SYNC2 = 1	◯	—	◯	—	—	—	—	—	—	◯	
PWM mode		◯	◯	—	PWM2 = 1	—	—	—	—	—	—	—	
Output compare A		◯	◯	—	PWM2 = 0	—	—	—	—	—	—	IOA2 = 0 Other bits unrestricted	
Output compare B		◯	◯	—	◯	—	—	—	—	—	—	◯	IOB2 Othe unre
Input capture A		◯	◯	—	PWM2 = 0	—	—	—	—	—	—	IOA2 = 1 Other bits unrestricted	
Input capture B		◯	◯	—	PWM2 = 0	—	—	—	—	—	—	◯	IOB2 Othe unre
Counter clearing	By compare match/input capture A	◯	◯	—	◯	—	—	—	—	—	—	◯	
	By compare match/input capture B	◯	◯	—	◯	—	—	—	—	—	—	◯	
	Syn-chronous clear	SYNC2 = 1	◯	—	◯	—	—	—	—	—	—	◯	
Phase counting mode		◯	MDF = 1	◯	◯	—	—	—	—	—	—	◯	

Legend: ◯: Setting available (valid). —: Setting does not affect this mode.
Note: The input capture function cannot be used in PWM mode. If compare match A and compare match B occur simultaneously , the compar

Operating Mode		Register Settings											
		TSNC	TMDR			TFCR			TOCR		TOER	TIOR3	
		Synchro-nization	MDF	FDIR	PWM	Comple-mentary PWM	Reset-Synchro-nized PWM	Buffering	XTGD	Output Level Select	Master Enable	IOA	
Synchronous preset		SYNC3 = 1	—	—	○	○*3	○	○	—	—	○*1	○	
PWM mode		○	—	—	PWM3 = 1	CMD1 = 0	CMD1 = 0	○	—	—	○	—	
Output compare A		○	—	—	PWM3 = 0	CMD1 = 0	CMD1 = 0	○	—	—	○	IOA2 = 0 Other bits unrestricted	
Output compare B		○	—	—	○	CMD1 = 0	CMD1 = 0	○	—	—	○	○	IO C u
Input capture A		○	—	—	PWM3 = 0	CMD1 = 0	CMD1 = 0	○	—	—	EA3 ignored Other bits unrestricted	IOA2 = 1 Other bits unrestricted	
Input capture B		○	—	—	PWM3 = 0	CMD1 = 0	CMD1 = 0	○	—	—	EB3 ignored Other bits unrestricted	○	IO C u
Counter clearing	By compare match/input capture A	○	—	—	○	Illegal setting: CMD1 = 1 CMD0 = 0	○*4	○	—	—	○*1	○	
	By compare match/input capture B	○	—	—	○	CMD1 = 0	CMD1 = 0	○	—	—	○*1	○	
	Syn-chronous clear	SYNC3 = 1	—	—	○	Illegal setting: CMD1 = 1 CMD0 = 0	○	○	—	—	○*1	○	
Complementary PWM mode		○*3	—	—	—	CMD1 = 1 CMD0 = 0	CMD1 = 1 CMD0 = 0	○	○*6	○	○	—	
Reset-synchronized PWM mode		○	—	—	—	CMD1 = 1 CMD0 = 1	CMD1 = 1 CMD0 = 1	○	○*6	○	○	—	
Buf fering (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 comp
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.

Operating Mode		Register Settings											
		TSNC	TMDR			TFCR			TOCR		TOER	TIOR4	
		Synchro-nization	MDF	FDIR	PWM	Comple-mentary PWM	Reset-Synchro-nized PWM	Buffering	XTGD	Output Level Select	Master Enable	IOA	
Synchronous preset		SYNC4 = 1	—	—	○	○*3	○	○	—	—	○*1	○	
PWM mode		○	—	—	PWM4 = 1	CMD1 = 0	CMD1 = 0	○	—	—	○	—	
Output compare A		○	—	—	PWM4 = 0	CMD1 = 0	CMD1 = 0	○	—	—	○	IOA2 = 0 Other bits unrestricted	
Output compare B		○	—	—	○	CMD1 = 0	CMD1 = 0	○	—	—	○	○	IO O un
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	○	IO O un
Counter clearing	By compare match/input capture A	○	—	—	○	Illegal setting: CMD1 = 1 CMD0 = 0	○*4	○	—	—	○*1	○	
	By compare match/input capture B	○	—	—	○	Illegal setting: CMD1 = 1 CMD0 = 0	○*4	○	—	—	○*1	○	
	Syn-chronous clear	SYNC4 = 1	—	—	○	Illegal setting: CMD1 = 1 CMD0 = 0	○*4	○	—	—	○*1	○	
Complementary PWM mode		○*3	—	—	—	CMD1 = 1 CMD0 = 0	CMD1 = 1 CMD0 = 0	○	○	○	○	—	
Reset-synchronized PWM mode		○	—	—	—	CMD1 = 1 CMD0 = 1	CMD1 = 1 CMD0 = 1	○	○	○	○	—	
Buffering (BRA)		○	—	—	○	○	○	BFA4 = 1 Other bits unrestricted	—	—	○*1	○	
Buffering (BRB)		○	—	—	○	○	○	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 comp
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 av
5. In complementary PWM mode, select the same clock source for channels 3 and 4.

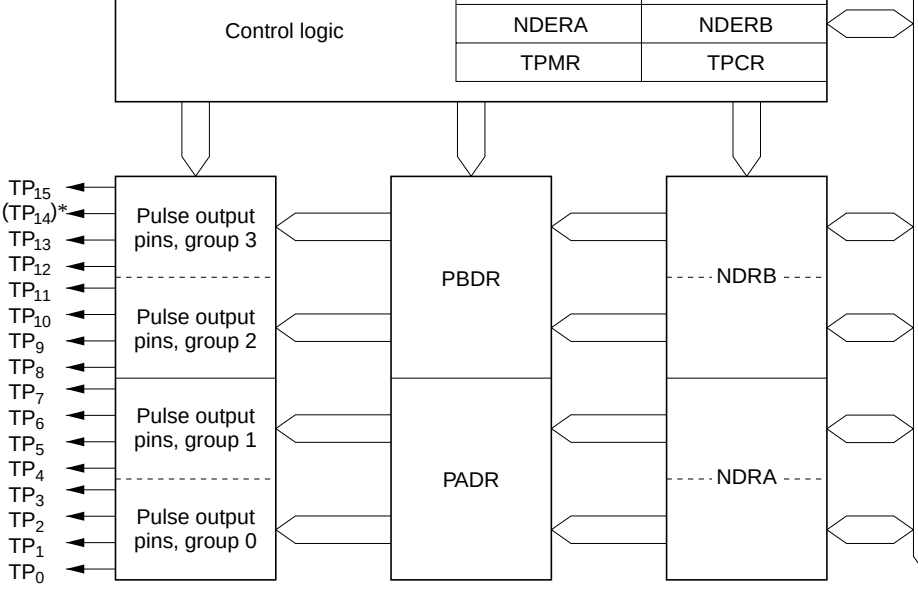
independently.

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 ba
- Four output groups and one 3-bit output.
  Output trigger signals can be selected in 4-bit groups to provide up to three differe
  outputs and one 3-bit output.
- Selectable output trigger signals
  Output trigger signals can be selected for each group from the compare-match sign
  ITU channels.
- Non-overlap mode
  A non-overlap margin can be provided between pulse outputs.

Note: * Note that since this LSI does not have a TP_{14} pin, it is a 15-bit programmab
pattern controller (TPC).



Legend:

TPMR:	TPC output mode register
TPCR:	TPC output control register
NDERB:	Next data enable register B
NDERA:	Next data enable register A
PBDDR:	Port B data direction register
PADDR:	Port A data direction register
NDRB:	Next data register B
NDRA:	Next data register A
PBDR:	Port B data register
PADR:	Port A data register

Note: * Since this LSI does not have this pin, this signal cannot be output to the outside.

Figure 9.1 TPC Block Diagram

TPC output 2	TP_2	Output	
TPC output 3	TP_3	Output	
TPC output 4	TP_4	Output	Group 1 pulse outp
TPC output 5	TP_5	Output	
TPC output 6	TP_6	Output	
TPC output 7	TP_7	Output	
TPC output 8	TP_8	Output	Group 2 pulse outp
TPC output 9	TP_9	Output	
TPC output 10	TP_{10}	Output	
TPC output 11	TP_{11}	Output	
TPC output 12	TP_{12}	Output	Group 3 pulse outp
TPC output 13	TP_{13}	Output	
(TPC output 14)*	(TP_{14})*	(Output)*	
TPC output 15	TP_{15}	Output	

Note: * Since this LSI does not have this pin, this signal cannot be output to the ou

H'FFD4	Port B data direction register	PBDDR	W	H'00
H'FFD6	Port B data register	PBDR	R/(W)*[2]	H'00
H'FFA0	TPC output mode register	TPMR	R/W	H'F0
H'FFA1	TPC output control register	TPCR	R/W	H'FF
H'FFA2	Next data enable register B	NDERB	R/W	H'00
H'FFA3	Next data enable register A	NDERA	R/W	H'00
H'FFA5/ H'FFA7*[3]	Next data register A	NDRA	R/W	H'00
H'FFA4/ H'FFA6*[3]	Next data register B	NDRB	R/W	H'00

Notes: 1. Lower 16 bits of the address.

2. Bits used for TPC output cannot be written.

3. The NDRA address is H'FFA5 when the same output trigger is selected for T groups 0 and 1 by settings in TPCR. When the output triggers are different, t address is H'FFA7 for group 0 and H'FFA5 for group 1. Similarly, the addres is H'FFA4 when the same output trigger is selected for TPC output groups 2 settings in TPCR. When the output triggers are different, the NDRB address for group 2 and H'FFA4 for group 3.

Initial value	0	0	0	0	0	0	0
Read/Write	W	W	W	W	W	W	W

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

Port A is multiplexed with pins TP_7 to TP_0. Bits corresponding to pins used for TPC o be set to 1. For further information about PADDR, see section 7.10, Port A.

9.2.2 Port A Data Register (PADR)

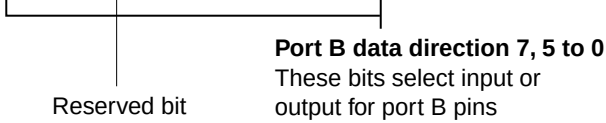
PADR is an 8-bit readable/writable register that stores TPC output data for groups 0 a these TPC output groups are used.

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

Port A data 7 to 0
These bits store output data for TPC output groups 0 and 1

Note: * Bits selected for TPC output by NDERA settings become read-only bits.

For further information about PADR, see section 7.10, Port A.

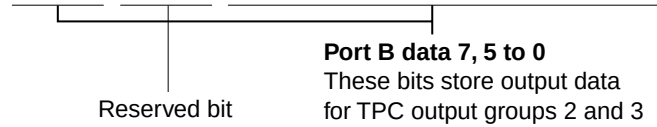


Port B is multiplexed with pins TP_{15}, TP_{13} to TP_{8}. Bits corresponding to pins used for TI must be set to 1. For further information about PBDDR, see section 7.11, Port B.

9.2.4 Port B Data Register (PBDR)

PBDR is an 8-bit readable/writable register that stores TPC output data for groups 2 an these TPC output groups are used.

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



Note: * Bits selected for TPC output by NDERB settings become read-only bits.

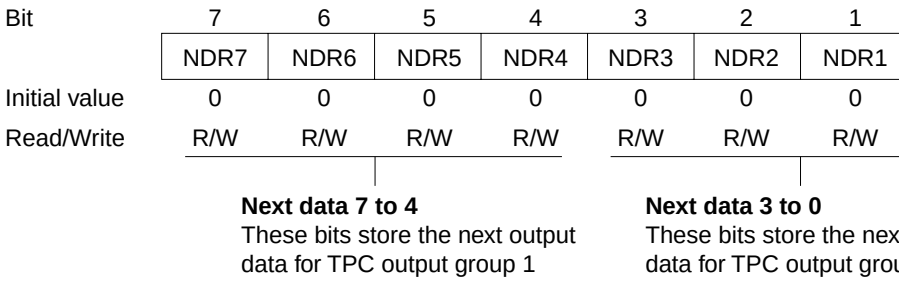
For further information about PBDR, see section 7.11, Port B.

software standby mode.

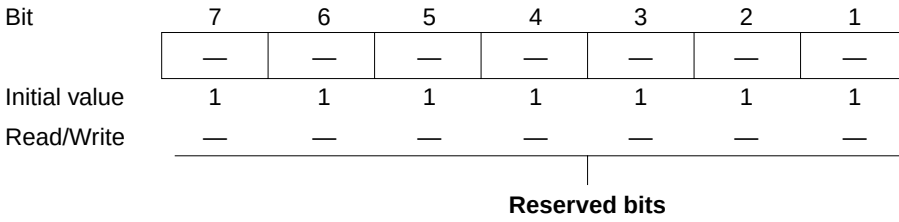
Same Trigger for TPC Output Groups 0 and 1

If TPC output groups 0 and 1 are triggered by the same compare match event, the ND
is H'FFA5. The upper 4 bits belong to group 1 and the lower 4 bits to group 0. Addres
consists entirely of reserved bits that cannot be modified and always read 1.

Address H'FFA5



Address H'FFA7



Bit	7	6	5	4	3	2	1
	NDR7	NDR6	NDR5	NDR4	—	—	—
Initial value	0	0	0	0	1	1	1
Read/Write	R/W	R/W	R/W	R/W	—	—	—

Next data 7 to 4
These bits store the next output data for TPC output group 1

Reserved bits

Address H'FFA7

Bit	7	6	5	4	3	2	1
	—	—	—	—	NDR3	NDR2	NDR1
Initial value	1	1	1	1	0	0	0
Read/Write	—	—	—	—	R/W	R/W	R/W

Reserved bits

Next data 3 to 0
These bits store the next
data for TPC output grou

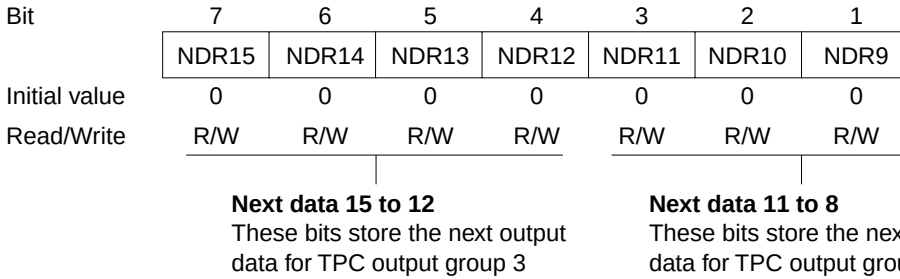
software standby mode.

Note: * Since this LSI does not have a TP_{14} pin, the TP_{14} signal cannot be output to

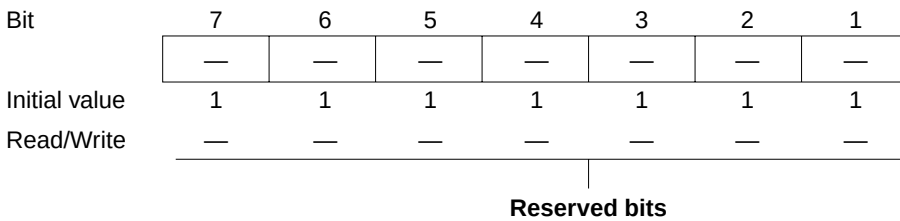
Same Trigger for TPC Output Groups 2 and 3

If TPC output groups 2 and 3 are triggered by the same compare match event, the ND
is H'FFA4. The upper 4 bits belong to group 3 and the lower 4 bits to group 2. Addres
consists entirely of reserved bits that cannot be modified and always read 1.

Address H'FFA4



Address H'FFA6



Address H'FFA4

Bit	7	6	5	4	3	2	1
	NDR15	NDR14	NDR13	NDR12	—	—	—
Initial value	0	0	0	0	1	1	1
Read/Write	R/W	R/W	R/W	R/W	—	—	—

Next data 15 to 12
These bits store the next output data for TPC output group 3

Reserved bits

Address H'FFA6

Bit	7	6	5	4	3	2	1
	—	—	—	—	NDR11	NDR10	NDR9
Initial value	1	1	1	1	0	0	0
Read/Write	—	—	—	—	R/W	R/W	R/W

Reserved bits

Next data 11 to 8
These bits store the n
data for TPC output g

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

Next data enable 7 to 0
These bits enable or disable
TPC output groups 1 and 0

If a bit is enabled for TPC output by NDERA, then when the ITU compare match even the TPC output control register (TPCR) occurs, the NDRA value is automatically tran the corresponding PADR bit, updating the output value. If TPC output is disabled, the not transferred from NDRA to PADR and the output value does not change.

NDERA is initialized to H'00 by a reset and in hardware standby mode. It is not initial software standby mode.

Bits 7 to 0—Next Data Enable 7 to 0 (NDER7 to NDER0): These bits enable or dis output groups 1 and 0 (TP_7 to TP_0) on a bit-by-bit basis.

Bits 7 to 0 NDER7 to NDER0	Description	
0	TPC outputs TP_7 to TP_0 are disabled (NDR_7 to NDR0 are not transferred to PA_7 to PA_0)	(In
1	TPC outputs TP_7 to TP_0 are enabled (NDR_7 to NDR_0 are transferred to PA_7 to PA_0)	

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

Next data enable 15 to 8
These bits enable or disable
TPC output groups 3 and 2

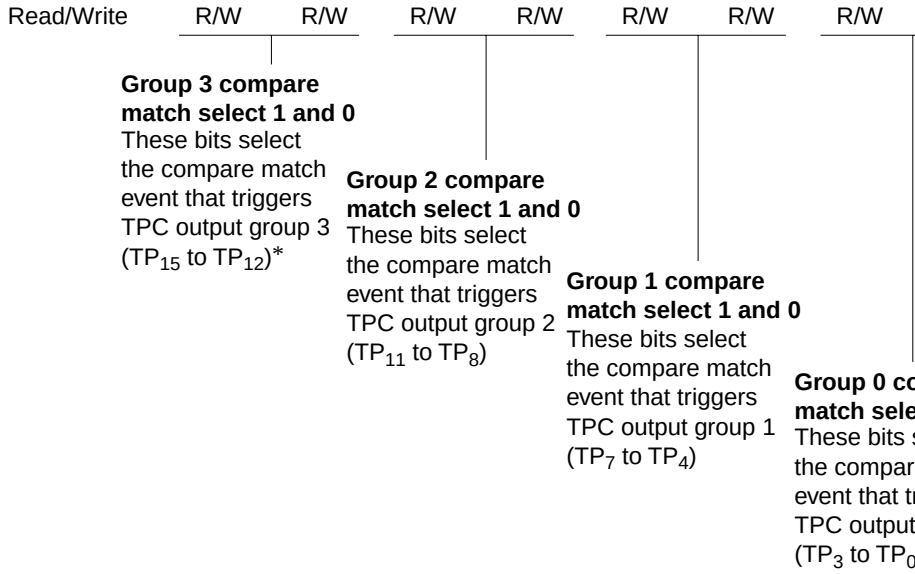
If a bit is enabled for TPC output by NDERB, then when the ITU compare match event the TPC output control register (TPCR) occurs, the NDRB value is automatically transf corresponding PBDR bit, updating the output value. If TPC output is disabled, the bit v transferred from NDRB to PBDR and the output value does not change.

NDERB is initialized to H'00 by a reset and in hardware standby mode. It is not initiali software standby mode.

Bits 7 to 0—Next Data Enable 15 to 8 (NDER15 to NDER8): These bits enable or di output groups 3 and 2 (TP_{15} to TP_8)* on a bit-by-bit basis.

Bits 7 to 0 NDER15 to NDER8	Description
0	TPC outputs TP_{15} to TP_8 are disabled (NDR15 to NDR8 are not transferred to PB_7 to PB_0) (Ini
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 t



Note: * Since this LSI does not have a TP_{14} pin, the TP_{14} signal cannot be output to

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

1	0	TPC output group 3 (TP_{15} to TP_{12})* is triggered by comp in ITU channel 2
	1	TPC output group 3 (TP_{15} to TP_{12})* is triggered by comp in ITU channel 3 (Ini

Note: * Since this LSI does not have a TP_{14} pin, the TP_{14} signal cannot be output off-

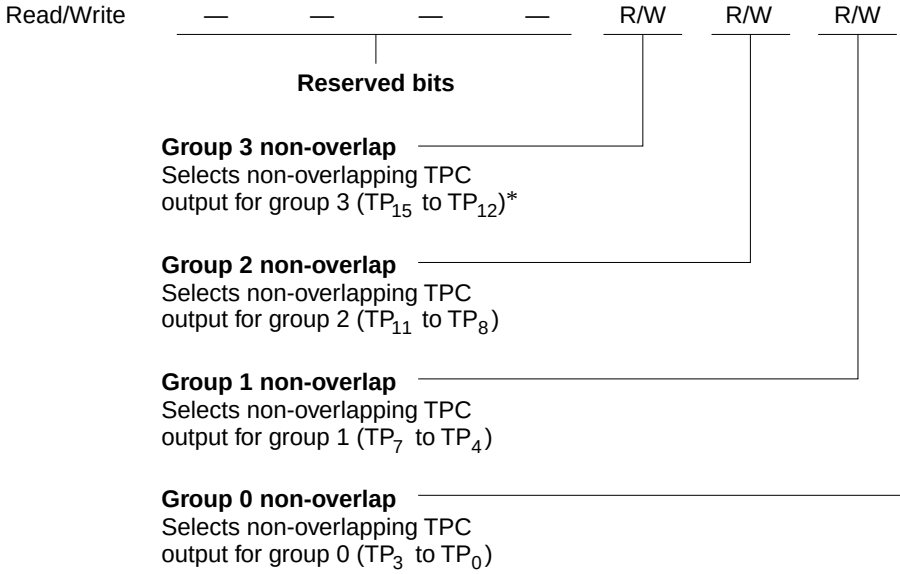
Bits 5 and 4—Group 2 Compare Match Select 1 and 0 (G2CMS1, G2CMS0): Thes select the compare match event that triggers TPC output group 2 (TP_{11} to TP_8).

Bit 5 G2CMS1	Bit4 G2CMS0	Description
0	0	TPC output group 2 (TP_{11} to TP_8) is triggered by compa ITU channel 0
	1	TPC output group 2 (TP_{11} to TP_8) is triggered by compa ITU channel 1
1	0	TPC output group 2 (TP_{11} to TP_8) is triggered by compa ITU channel 2
	1	TPC output group 2 (TP_{11} to TP_8) is triggered by compa ITU channel 3 (Ini

1	0	TPC output group 1 (TP_7 to TP_4) is triggered by compa ITU channel 2
	1	TPC output group 1 (TP_7 to TP_4) is triggered by compa ITU channel 3 (I

Bits 1 and 0—Group 0 Compare Match Select 1 and 0 (G0CMS1, G0CMS0): The 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 compa ITU channel 0
	1	TPC output group 0 (TP_3 to TP_0) is triggered by compa ITU channel 1
1	0	TPC output group 0 (TP_3 to TP_0) is triggered by compa ITU channel 2
	1	TPC output group 0 (TP_3 to TP_0) is triggered by compa ITU channel 3 (I



Note: * Since this LSI does not have a TP_{14} pin, the TP_{14} signal cannot be output to t

The output trigger period of a non-overlapping TPC output waveform is set in general r (GRB) in the ITU channel selected for output triggering. The non-overlap margin is set register A (GRA). The output values change at compare match A and B. For details see 9.3.4, Non-Overlapping TPC Output.

TPMR is initialized to H'F0 by a reset and in hardware standby mode. It is not initialize software standby mode.

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

1	Non-overlapping TPC output in group 3 (independent 1 and 0 output at match A and B in the selected ITU channel)

Bit 2—Group 2 Non-Overlap (G2NOV): Selects normal or non-overlapping TPC ou group 2 (TP_{11} to TP_8).

Bit 2 G2NOV	Description
0	Normal TPC output in group 2 (output values change at compare match selected ITU channel) (I
1	Non-overlapping TPC output in group 2 (independent 1 and 0 output at match A and B in the selected ITU channel)

Bit 1—Group 1 Non-Overlap (G1NOV): Selects normal or non-overlapping TPC ou group 1 (TP_7 to TP_4).

Bit 1 G1NOV	Description
0	Normal TPC output in group 1 (output values change at compare match selected ITU channel) (I
1	Non-overlapping TPC output in group 1 (independent 1 and 0 output at 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, T
is enabled. The TPC output initially consists of the corresponding PADR or PBDR con
When a compare-match event selected in TPCR occurs, the corresponding NDRA or N
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 operatin
conditions.

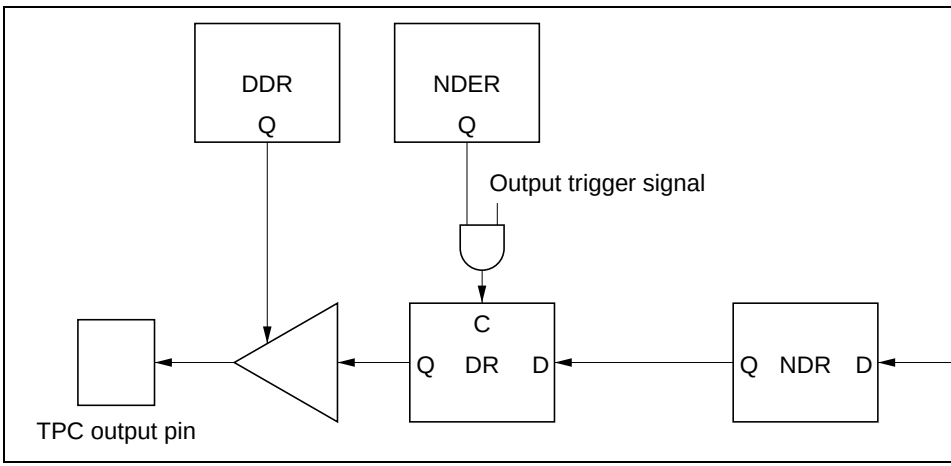


Figure 9.2 TPC Output Operation

Sequential output of up to 16-bit patterns is possible by writing new output data to ND
NDRB before the next compare match. For information on non-overlapping operation
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
when the selected compare match event occurs. Figure 9.3 shows the timing of these
for the case of normal output in groups 0 and 1, triggered by compare match A.

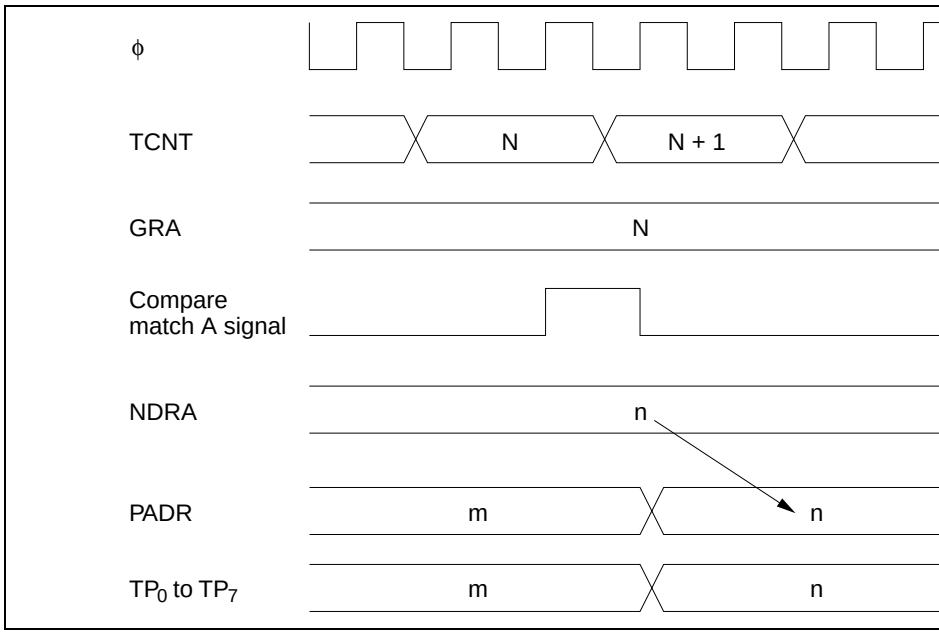
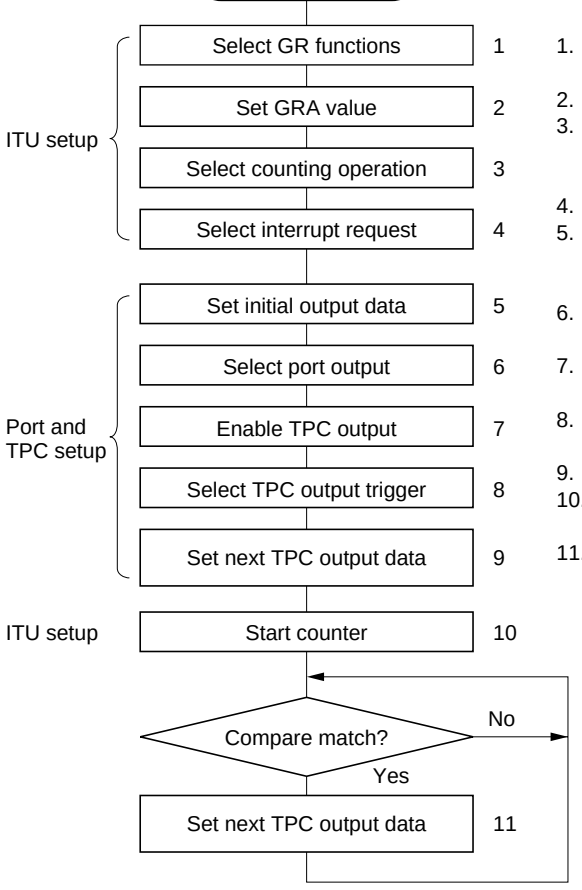
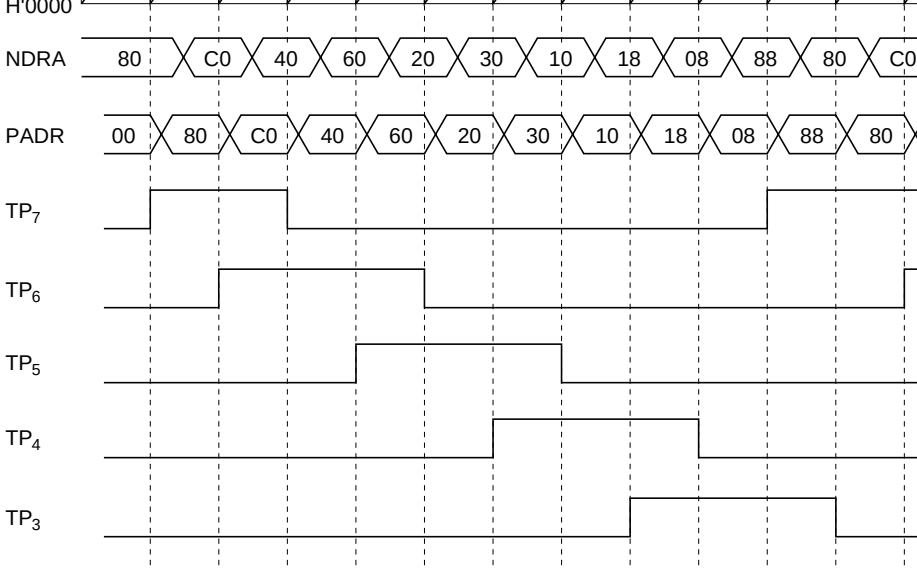


Figure 9.3 Timing of Transfer of Next Data Register Contents and Output (



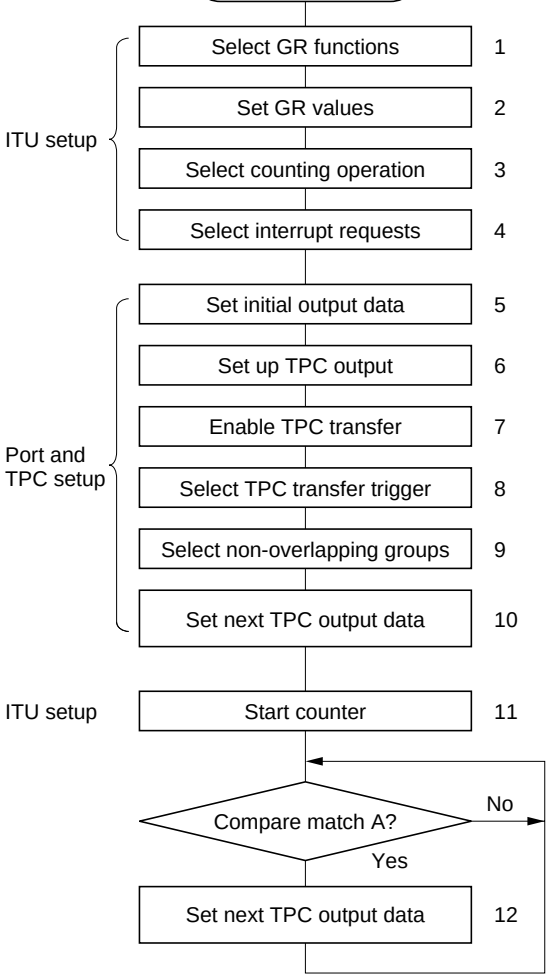
1. Set TIOR to make GRA an output
register (with output inhibited).
2. Set the TPC output trigger period.
3. Select the counter clock source wit
TPSC2 to TPSC0 in TCR. Select t
clear source with bits CCLR1 and
4. Enable the IMFA interrupt in TIER.
5. Set the initial output values in the D
of the input/output port pins to be u
TPC output.
6. Set the DDR bits of the input/outpu
pins to be used for TPC output to 1
7. Set the NDER bits of the pins to be
TPC output to 1.
8. Select the ITU compare match eve
used as the TPC output trigger in T
9. Set the next TPC output values in th
10. Set the STR bit to 1 in TSTR to star
timer counter.
11. At each IMFA interrupt, set the next
values in the NDR bits.

Figure 9.4 Setup Procedure for Normal TPC Output (Example)



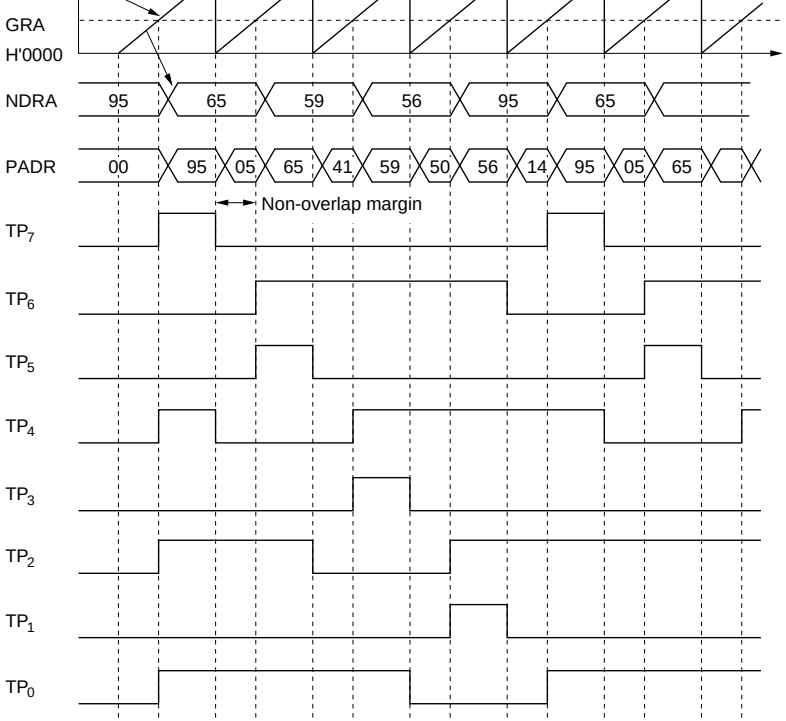
- The ITU channel to be used as the output trigger channel is set up so that GRA is an output c register and the counter will be cleared by compare match A. The trigger period is set in GRA The IMIEA bit is set to 1 in TIER to enable the compare match A interrupt.
- H'F8 is written in PADDR and NDERA, and bits G1CMS1, G1CMS0, G0CMS1, and G0CMS0 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 c are transferred to PADR and output. The compare match/input capture A (IMFA) interrupt ser 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 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)



1. Set TIOR to make GRA and GRB
compare registers (with output inh
2. Set the TPC output trigger period i
and the non-overlap margin in GR
3. Select the counter clock source wi
TPSC2 to TPSC0 in TCR. Select t
clear source with bits CCLR1 and
4. Enable the IMFA interrupt in TIER
5. Set the initial output values in the
of the input/output port pins to be u
TPC output.
6. Set the DDR bits of the input/outpu
to be used for TPC output to 1.
7. Set the NDER bits of the pins to b
TPC output to 1.
8. In TPCR, select the ITU compare
event to be used as the TPC outpu
9. In TPMR, select the groups that w
in non-overlap mode.
10. Set the next TPC output values in
bits.
11. Set the STR bit to 1 in TSTR to sta
counter.
12. At each IMFA interrupt, write the n
value in the NDR bits.

Figure 9.6 Setup Procedure for Non-Overlapping TPC Output (Exampl



- The ITU channel to be used as the output trigger channel is set up so that GRA and GRB are outpu 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 en IMFA interrupts.
- H'FF is written in PADDR and NDERA, and bits G1CMS1, G1CMS0, G0CMS1, and G0CMS0 are s 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 fro 1 to 0. When compare match A occurs, outputs change from 0 to 1 (the change from 0 to 1 is delay 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' at successive IMFA interrupts.

Figure 9.7 Non-Overlapping TPC Output Example (Four-Phase Complem Non-Overlapping Pulse Output)

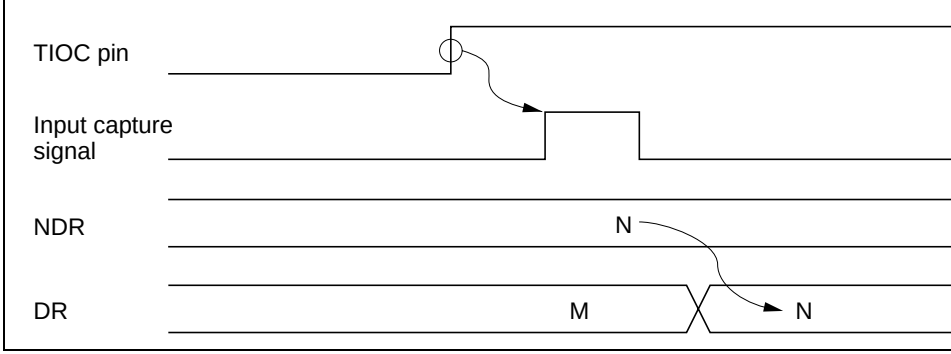


Figure 9.8 TPC Output Triggering by Input Capture (Example)

Pin functions should be changed only under conditions in which the output trigger eve
occur.

Note: * Since this LSI does not have a TP_{14} pin, the TP_{14} signal cannot be output to

9.4.2 Note on Non-Overlapping Output

During non-overlapping operation, the transfer of NDR bit values to DR bits takes pla
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
   if their value is 1.

Figure 9.9 illustrates the non-overlapping TPC output operation.

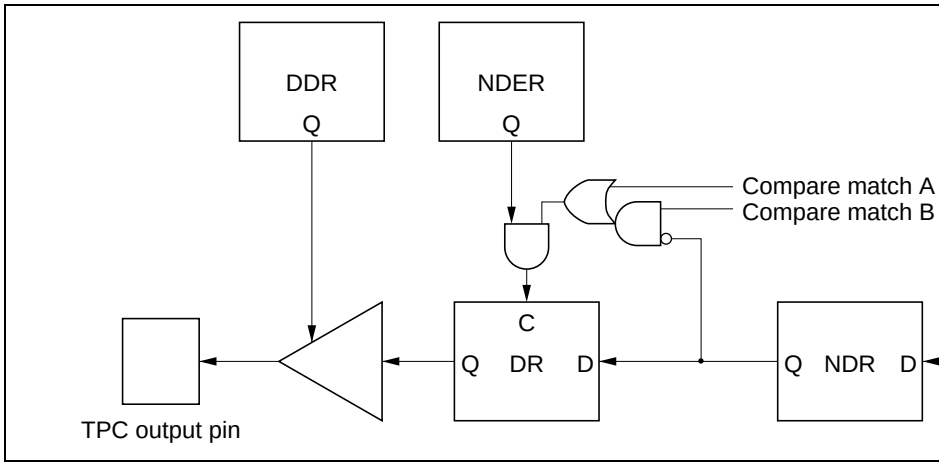


Figure 9.9 Non-Overlapping TPC Output

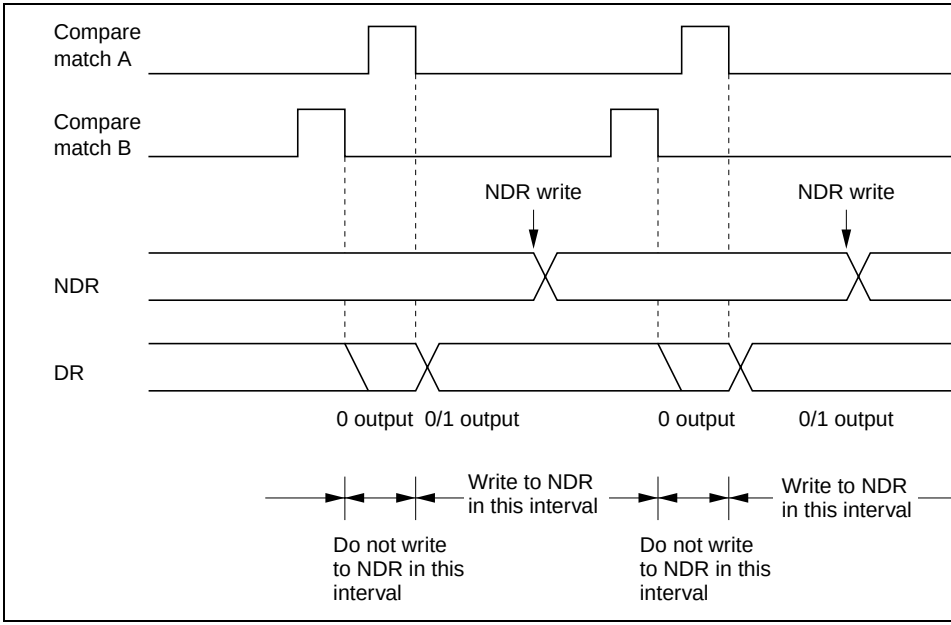


Figure 9.10 Non-Overlapping Operation and NDR Write Timing

system crash allows the timer counter (TCNT) to overflow before being rewritten. In
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 interr
  generated in interval timer operation.
- Watchdog timer reset signal resets the entire H8/3039 Group chip internally, and c
  output externally.*
  The reset signal generated by timer counter overflow during watchdog timer opera
  the entire H8/3039 Group internally. An external reset signal can be output from th
  to reset other system devices simultaneously.

Note: * The $\overline{\text{RESO}}$ pin of the mask ROM version is the dedicated FWE input pin o
ZTAT version. Therefore, the F-ZTAT version cannot output the reset sign
outside.

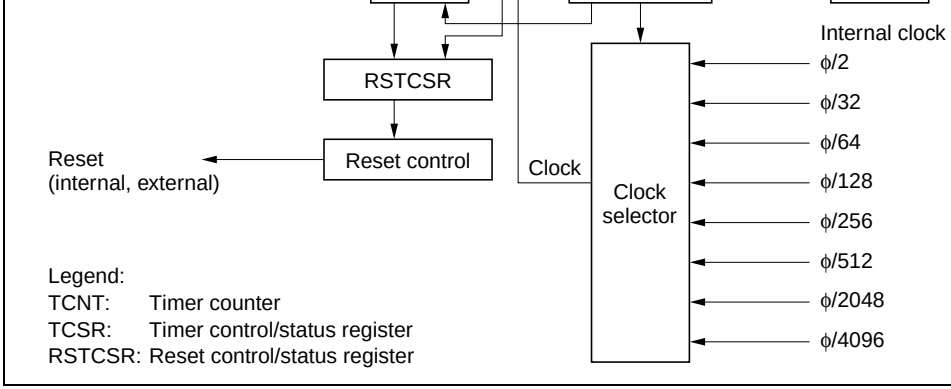


Figure 10.1 WDT Block Diagram

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 use
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 r

Note: * Open-drain output. Externally pull-up to Vcc whether or not the reset output

	H'FFA9	Timer counter	TCNT	R/W
H'FFAA	H'FFAB	Reset control/status register	RSTCSR	R/(W)*[3]

Notes: 1. Lower 16 bits of the address.
2. Write word data starting at this address.
3. Only 0 can be written in bit 7 to clear the flag.

10.2 Register Descriptions

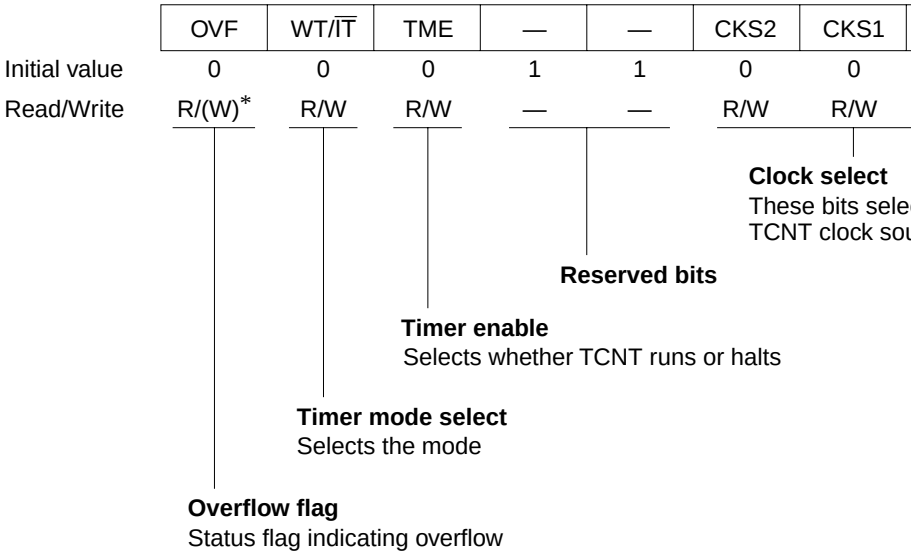
10.2.1 Timer Counter (TCNT)

TCNT is an 8-bit readable and writable* up-counter.

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

When the TME bit is set to 1 in TCSR, TCNT starts counting pulses generated from a clock source selected by bits CKS2 to CKS0 in TCSR. When the count overflows (cha H'FF to H'00), the OVF bit is set to 1 in TCSR. TCNT is initialized to H'00 by a reset the TME bit is cleared to 0.

Note: * TCNT is write-protected by a password. For details see section 10.2.4, Not Register Access.



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

Bits 7 to 5 are initialized to 0 by a reset and in standby mode. Bits 2 to 0 are initialized
reset. In software standby mode bits 2 to 0 are not initialized, but retain their previous v

	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 interval timer. If used as an interval timer, the WDT generates an interval timer interru when TCNT overflows. If used as a watchdog timer, the WDT generates a reset signal TCNT overflows.

Bit 6 WT/$\overline{IT}$	Description	
0	Interval timer: requests interval timer interrupts	(
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	(
1	TCNT is counting	

Bits 4 and 3—Reserved: These bits cannot be modified and are always read as 1.

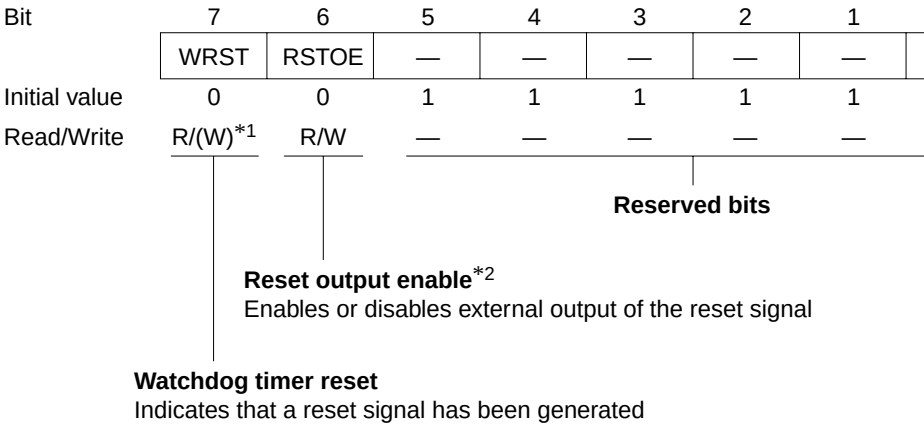
Bits 2 to 0—Clock Select 2 to 0 (CKS2/1/0): These bits select one of eight internal c obtained by prescaling the system clock (ϕ), for input to TCNT.

	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 h 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, N Register Access.



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-ZTA 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 initial reset signals generated by watchdog timer overflow.

	(1) Cleared to 0 by reset signal input at 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

Bit 6—Reset Output Enable (RSTOE): Enables or disables external output at the $\overline{\text{RE}}$ the reset signal generated if TCNT overflows during watchdog timer operation.

Bit 6 RSTOE	Description	
0	Reset signal is not output externally	(
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.

instructions. Figure 10.2 shows the format of data written to TCNT and TCSR. TCNT a
both have the same write address. The write data must be contained in the lower byte o
written word. The upper byte must contain H'5A (password for TCNT) or H'A5 (passw
TCSR). This transfers the write data from the lower byte to TCNT or TCSR.

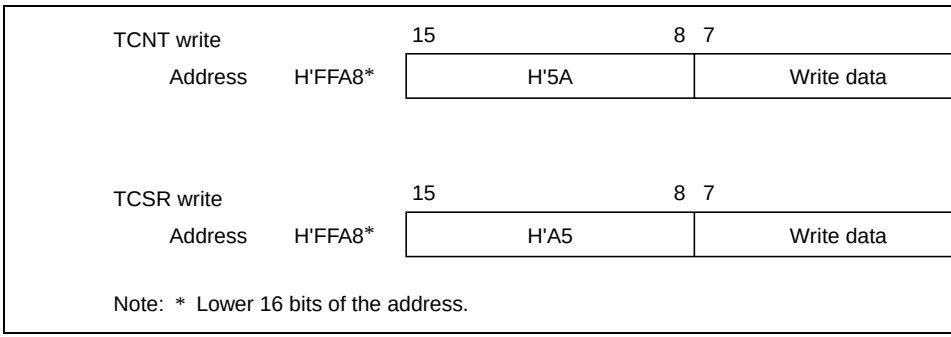


Figure 10.2 Format of Data Written to TCNT and TCSR

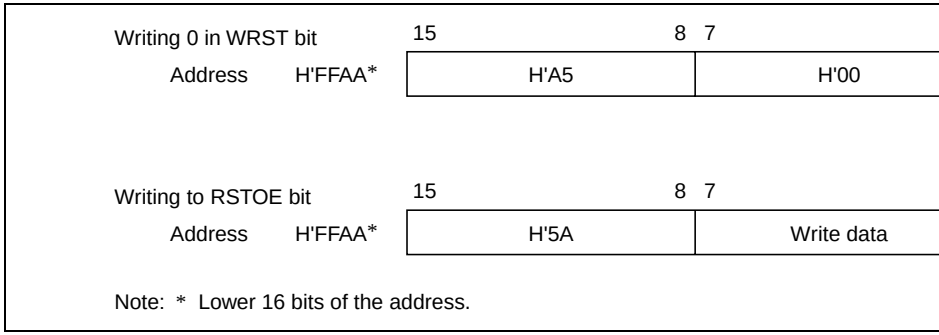


Figure 10.3 Format of Data Written to RSTCSR

Reading TCNT, TCSR, and RSTCSR

These registers are read like other registers. Byte access instructions can be used. The addresses are H'FFA8 for TCSR, H'FFA9 for TCNT, and H'FFAB for RSTCSR, as lis 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.

WT/IT and TME bits to 1 in TCSR. Software must prevent TCNT overflow by rewritin
TCNT value (normally by writing H'00) before overflow occurs. If TCNT fails to be re
overflows due to a system crash etc., the H8/3039 Group is internally reset for a duratio
states.

The watchdog reset signal can be externally output from the $\overline{\text{RESO}}$ pin* to reset externa
devices. The reset signal is output externally for 132 states. External output can be enab
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. Soft
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, t
signal cannot be output to the outside.

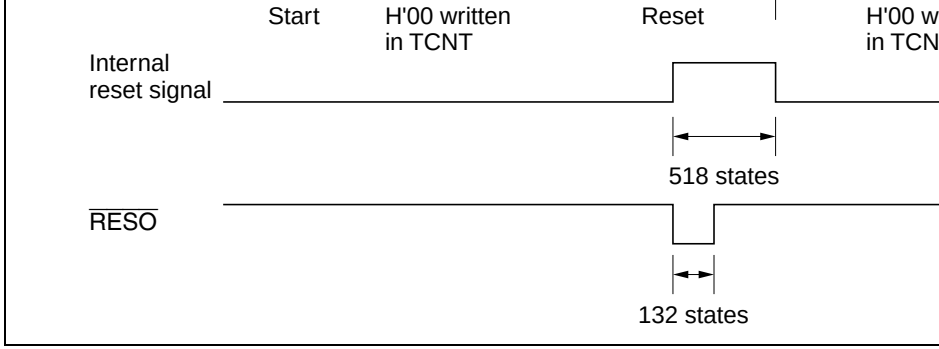


Figure 10.4 Watchdog Timer Operation (Mask ROM Version)

10.3.2 Interval Timer Operation

Figure 10.5 illustrates interval timer operation. To use the WDT as an interval timer, c
WT/$\overline{IT}$ to 0 and set bit TME to 1 in TCSR. An interval timer interrupt request is gener
TCNT overflow. This function can be used to generate interval timer interrupts at regu
intervals.

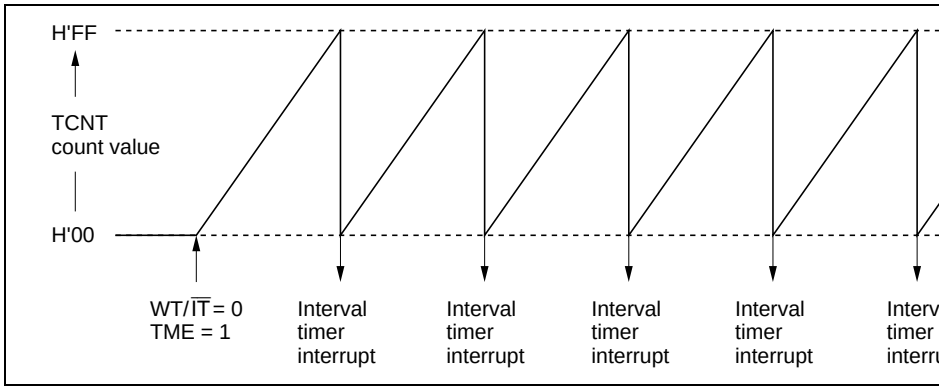


Figure 10.5 Interval Timer Operation

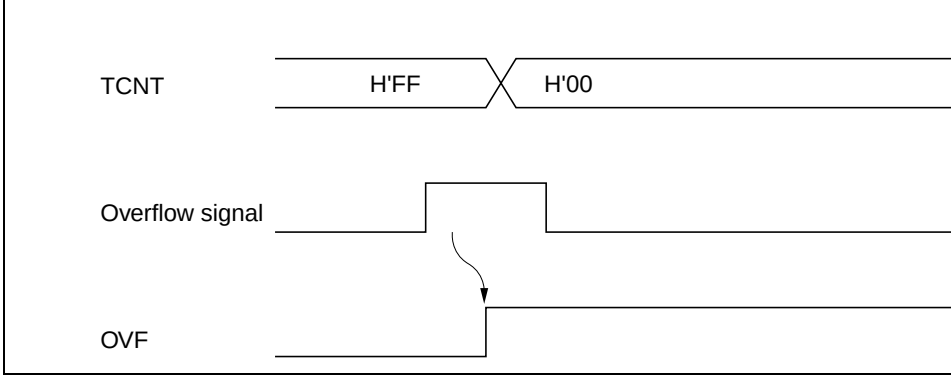


Figure 10.6 Timing of Setting of OVF

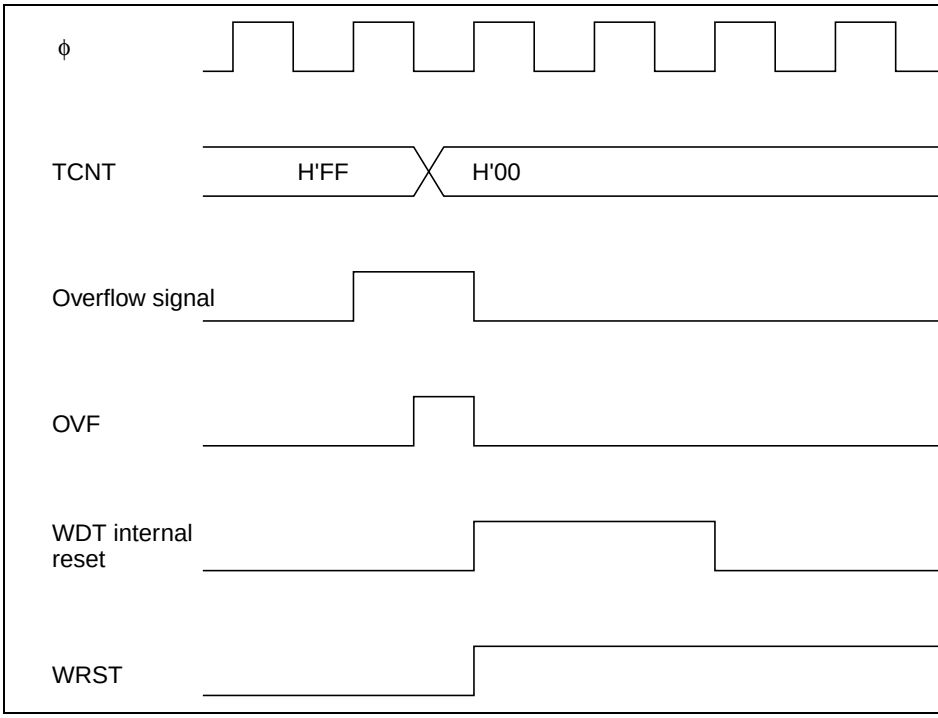


Figure 10.7 Timing of Setting of WRST Bit and Internal Reset

10.4 Interrupts

During interval timer operation, an overflow generates an interval timer interrupt (WO interval timer interrupt is requested whenever the OVF bit is set to 1 in TCSR.

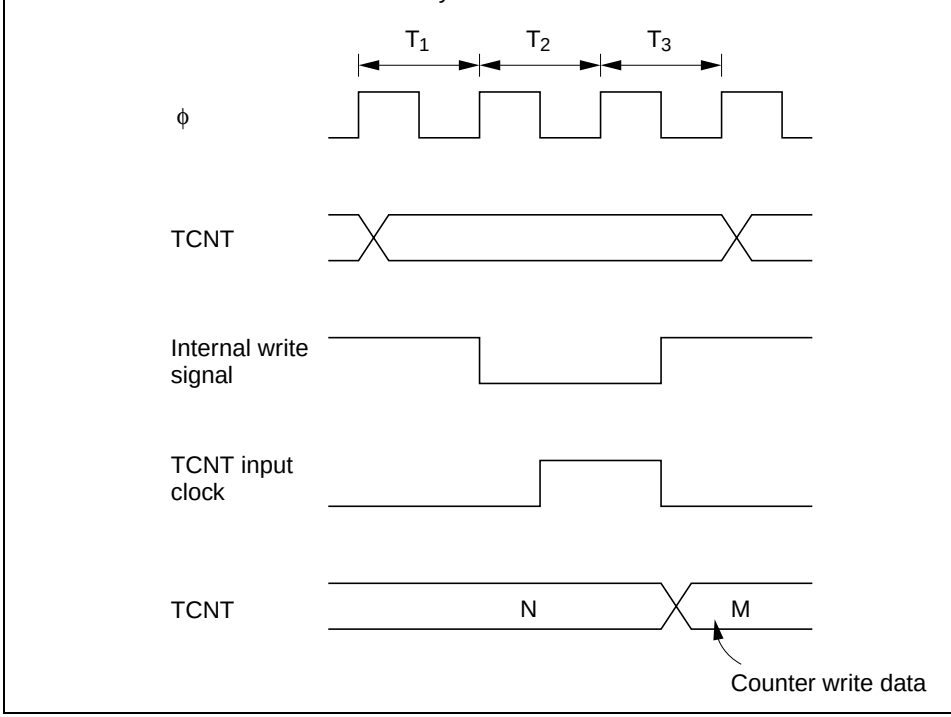


Figure 10.8 Contention between TCNT Write and Increment

Changing CKS2 to CKS0 Values

Halt TCNT by clearing the TME bit to 0 in TCSR before changing the values of bits CI
CKS0.

among two or more processors.

When the SCI is not used, it can be halted to conserve power. Each SCI channel can b independently. For details see section 17.6, Module Standby Function.

Channel 0 (SCI0) also has a smart card interface function conforming to the ISO/IEC7 (Identification Card) standard. This function supports serial communication with a sm details, see section 12, Smart Card Interface.

11.1.1 Features

SCI features are listed below.

- Selection of asynchronous or synchronous mode for serial communication

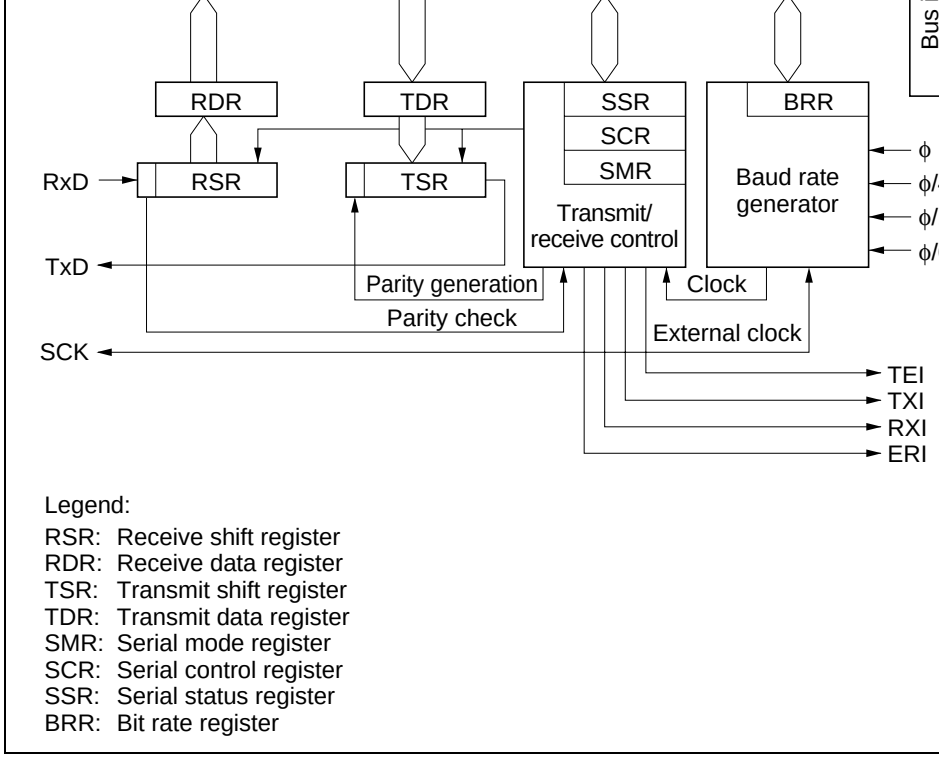
a. Asynchronous mode

Serial data communication is synchronized one character at a time. The SCI can co with a universal asynchronous receiver/transmitter (UART), asynchronous commu interface adapter (ACIA), or other chip that employs standard asynchronous serial communication. It can also communicate with two or more other processors using 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 err

The transmitting and receiving sections are independent, so the SCI can transmit an simultaneously. The transmitting and receiving sections are both double-buffered, s data can be transmitted and received continuously.

- Built-in baud rate generator with selectable bit rates
- Selectable transmit/receive clock sources: internal clock from baud rate generator, c clock from the SCK pin.
- Four types of interrupts
  Transmit-data-empty, transmit-end, receive-data-full, and receive-error interrupts ar independently.



Legend:

RSR: Receive shift register
RDR: Receive data register
TSR: Transmit shift register
TDR: Transmit data register
SMR: Serial mode register
SCR: Serial control register
SSR: Serial status register
BRR: Bit rate register

Figure 11.1 SCI Block Diagram

	Transmit data pin	TxD_0	Output	SCI_0 transmit d
1	Serial clock pin	SCK_1	Input/output	SCI_1 clock inpu
	Receive data pin	RxD_1	Input	SCI_1 receive da
	Transmit data pin	TxD_1	Output	SCI_1 transmit d

0	H'FFB0	Serial mode register	SMR	R/W	H
	H'FFB1	Bit rate register	BRR	R/W	H
	H'FFB2	Serial control register	SCR	R/W	H
	H'FFB3	Transmit data register	TDR	R/W	H
	H'FFB4	Serial status register	SSR	R/(W)*[2]	H
	H'FFB5	Receive data register	RDR	R	H
1	H'FFB8	Serial mode register	SMR	R/W	H
	H'FFB9	Bit rate register	BRR	R/W	H
	H'FFBA	Serial control register	SCR	R/W	H
	H'FFBB	Transmit data register	TDR	R/W	H
	H'FFBC	Serial status register	SSR	R/(W)*[2]	H
	H'FFBD	Receive data register	RDR	R	H

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

Read/Write	—	—	—	—	—	—	—

The SCI loads serial data input at the RxD pin into RSR in the order received, LSB (bit
thereby converting the data to parallel data. When 1 byte has been received, it is autom
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
Initial value	0	0	0	0	0	0	0
Read/Write	R	R	R	R	R	R	R

When the SCI finishes receiving 1 byte of serial data, it transfers the received data from
RDR for storage. RSR is then ready to receive the next data. This double buffering allo
be received continuously.

RDR is a read-only register. Its contents cannot be modified by the CPU. RDR is initia
H'00 by a reset and in standby mode.

The SCI loads transmit data from TDR into TSR, then transmits the data serially from pin, LSB (bit 0) first. After transmitting one data byte, the SCI automatically loads the transmit data from TDR into TSR and starts transmitting it. If the TDRE flag is set to however, the SCI does not load the TDR contents into TSR. The CPU cannot read or directly.

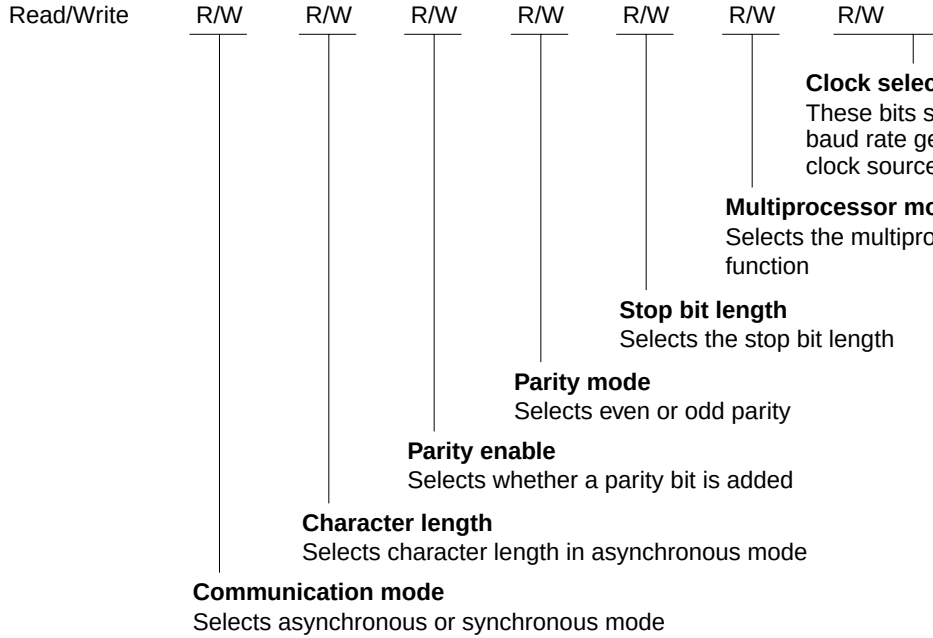
11.2.4 Transmit Data Register (TDR)

TDR is an 8-bit register that stores data for serial transmission.

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

When the SCI detects that TSR is empty, it moves transmit data written in TDR from TSR and starts serial transmission. Continuous serial transmission is possible by writi 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 mode.



The CPU can always read and write SMR. SMR is initialized to H'00 by a reset and in s
mode.

Bit 7—Communication Mode (C/$\overline{A}$): Selects whether the SCI operates in asynchrono
synchronous mode.

Bit 7 C/$\overline{A}$	Description	
0	Asynchronous mode	(Ini
1	Synchronous mode	

Bit 5—Parity Enable (PE): In asynchronous mode, this bit enables or disables the ad
parity bit to transmit data, and the checking of the parity bit in receive data. In synchro
the parity bit is neither added nor checked, regardless of the PE setting.

Bit 5 PE	Description	
0	Parity bit not added or checked	(In
1	Parity bit added and checked*	

Note: * When PE is set to 1, an even or odd parity bit is added to transmit data acc
even or odd parity mode selected by the O/$\overline{E}$ bit, and the parity bit in receiv
checked to see that it matches the even or odd mode selected by the O/$\overline{E}$ b

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
The O/$\overline{E}$ setting is ignored in synchronous mode, or when parity adding and checking
in asynchronous mode.

Bit 4 O/$\overline{E}$	Description	
0	Even parity*[1]	(In
1	Odd parity*[2]	

Notes: 1. When even parity is selected, the parity bit added to transmit data makes a
number of 1s in the transmitted character and parity bit combined. Receive
have an even number of 1s in the received character and parity bit combine
2. When odd parity is selected, the parity bit added to transmit data makes an
of 1s in the transmitted character and parity bit combined. Receive data mu
odd number of 1s in the received character and parity bit combined.

Bit 3—Stop Bit Length (STOP): Selects one or two stop bits in asynchronous mode.
is used only in asynchronous mode. In synchronous mode no stop bit is added, so the
setting is ignored.

stop bit is 1 it is treated as a stop bit. If the second stop bit is 0 it is treated as the start b
next incoming character.

Bit 2—Multiprocessor Mode (MP): Selects a multiprocessor format. When a multipr
format is selected, parity settings made by the PE and O/$\overline{E}$ bits are ignored. The MP bit
valid only in asynchronous mode. It is ignored in synchronous mode.

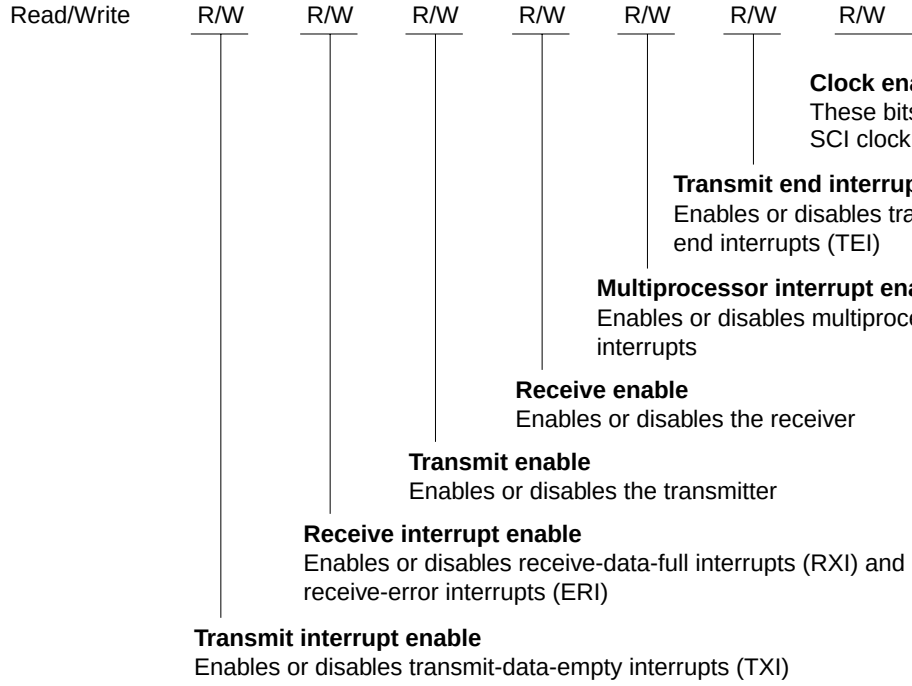
For further information on the multiprocessor communication function, see section 11.
Multiprocessor Communication.

Bit 2 MP	Description	
0	Multiprocessor function disabled	(Ini
1	Multiprocessor format selected	

Bits 1 and 0—Clock Select 1 and 0 (CKS1/0): These bits select the clock source of th
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, se
11.2.8, Bit Rate Register (BRR).

Bit 1 CKS1	Bit 0 CKS0	Description	
0	0	ϕ clock selected	(Ini
0	1	ϕ/4 clock selected	
1	0	ϕ/16 clock selected	
1	1	ϕ/64 clock selected	



The CPU can always read and write SCR. SCR is initialized to H'00 by a reset and in
mode.

Note: * TXI interrupt requests can be cleared by reading the value 1 from the TDRE 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 inte requested when the RDRF flag is set to 1 in SSR due to transfer of serial receive data fr 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* (Ini
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 th 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

Bit 5 TE	Description
0	Transmitting disabled*[1] (Ini
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 0 after writing of transmit data into TDR. Select the transmit format in SMR b setting the TE bit to 1.

2. In the enabled state, serial receiving starts when a start bit is detected in as
mode, or serial clock input is detected in synchronous mode. Select the rec
in SMR before setting the RE bit to 1.

Bit 3—Multiprocessor Interrupt Enable (MPIE): Enables or disables multiprocess
The MPIE setting is valid only in asynchronous mode, and only if the MP bit is set to
The MPIE setting is ignored in synchronous mode or when the MP bit is cleared to 0.

Bit 3 MPIE	Description
0	Multiprocessor interrupts are disabled (normal receive operation) (I [Clearing conditions] • The MPIE bit is cleared to 0 • MPB = 1 in received data
1	Multiprocessor interrupts are enabled* Receive-data-full interrupts (RXI), receive-error interrupts (ERI), and settin RDRF, FER, and ORER status flags in SSR are disabled until data with th multiprocessor bit set to 1 is received.

Note: * The SCI does not transfer receive data from RSR to RDR, does not detect errors, and does not set the RDRF, FER, and ORER flags in SSR. When it data in which MPB = 1, the SCI sets the MPB bit to 1 in SSR, automatically MPIE bit to 0, and enables RXI and ERI interrupts (if the RIE bit is set to 1 i setting of the FER and ORER flags.

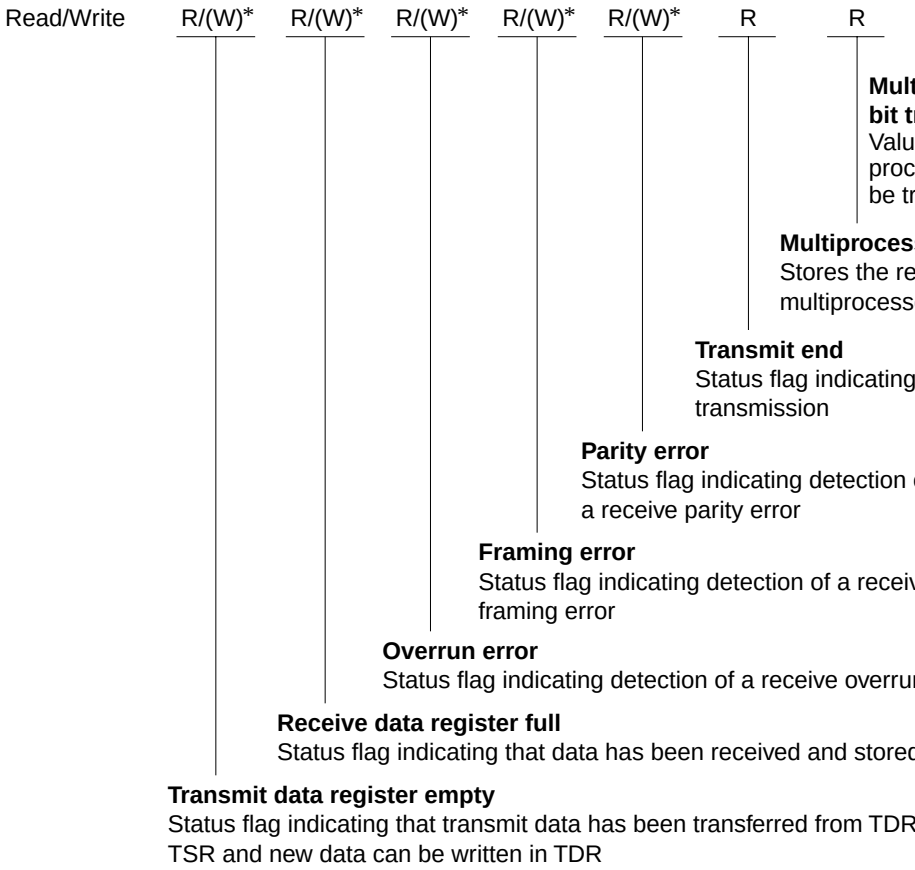
SSR, then clearing the TDRE flag to 0, thereby also clearing the TEND flag t
clearing the TEIE bit to 0.

Bits 1 and 0—Clock Enable 1 and 0 (CKE1/0): These bits select the SCI clock sourc
enable or disable clock output from the SCK pin. Depending on the settings of CKE1 a
the SCK pin can be used for generic input/output, serial clock output, or serial clock inp

The CKE0 setting is valid only in asynchronous mode, and only when the SCI is intern
clocked (CKE1 = 0). The CKE0 setting is ignored in synchronous mode, or when an ex
clock source is selected (CKE1 = 1). After setting the CKE1 and CKE0 bits, select the
operating mode in SMR. For further details on selection of the SCI clock source, see ta

Bit 1 CKE1	Bit 0 CKE0	Description	
0	0	Asynchronous mode	Internal clock, SCK pin available for gene input/output*[1]
		Synchronous mode	Internal clock, SCK pin used for serial clo
0	1	Asynchronous mode	Internal clock, SCK pin used for clock outp
		Synchronous mode	Internal clock, SCK pin used for serial clo
1	0	Asynchronous mode	External clock, SCK pin used for clock inp
		Synchronous mode	External clock, SCK pin used for serial clo
1	1	Asynchronous mode	External clock, SCK pin used for clock inp
		Synchronous mode	External clock, SCK pin used for serial clo

Notes: 1. Initial value
2. The output clock frequency is the same as the bit rate.
3. The input clock frequency is 16 times the bit rate.



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

The CPU can always read and write SSR, but cannot write 1 in the TDRE, RDRF, OR and FER flags. These flags can be cleared to 0 only if they have first been read while s TEND and MPB flags are read-only bits that cannot be written.

	Software reads TDRE while it is set to 1, then writes 0
1	TDR does not contain valid transmit data (Ini [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

Bit 6—Receive Data Register Full (RDRF): Indicates that RDR contains new receive

Bit 6 RDRF	Description
0	RDR does not contain new receive data (Ini [Clearing conditions] • The chip is reset or enters standby mode • Software reads RDRF while it is set to 1, then writes 0 • The DMAC reads data from RDR
1	RDR contains new receive data [Setting condition] When serial data is received normally and transferred from RSR to RDR

Note: The RDR contents and RDRF flag are not affected by detection of receive errors clearing of the RE bit to 0 in SCR. They retain their previous values. If the RDRF set to 1 when reception of the next data ends, an overrun error occurs and recei lost.

	• Software reads ORER while it is set to 1, then writes 0
1	A receive overrun error occurred*[2] [Setting condition] Reception of the next serial data ends when RDRF = 1

Notes: 1. Clearing the RE bit to 0 in SCR does not affect the ORER flag, which retain previous value.
2. RDR continues to hold the receive data before the overrun error, so subsec data is lost. Serial receiving cannot continue while the ORER flag is set to 1 synchronous mode, serial transmitting is also disabled.

Bit 4—Framing Error (FER): Indicates that data reception ended abnormally due to error in asynchronous mode.

Bit 4 FER	Description
0	Receiving is in progress or has ended normally (I [Clearing conditions] • The chip is reset or enters standby mode • Software reads FER while it is set to 1, then writes 0
1	A receive framing error occurred*[2] [Setting condition] The stop bit at the end of receive data is checked and found to be 0

Notes: 1. Clearing the RE bit to 0 in SCR does not affect the FER flag, which retains value.
2. When the stop bit length is 2 bits, only the first bit is checked. The second s checked. When a framing error occurs the SCI transfers the receive data in does not set the RDRF flag. Serial receiving cannot continue while the FER to 1. In synchronous mode, serial transmitting is also disabled.

	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 th odd parity setting of O/$\overline{E}$ in SMR

Notes: 1. Clearing the RE bit to 0 in SCR does not affect the PER flag, which retains it value.
2. When a parity error occurs the SCI transfers the receive data into RDR but d the RDRF flag. Serial receiving cannot continue while the PER flag is set to synchronous mode, serial transmitting is also disabled.

Bit 2—Transmit End (TEND): Indicates that when the last bit of a serial character wa transmitted TDR did not contain new transmit data, so transmission has ended. The TE a read-only bit and cannot be written.

Bit 2 TEND	Description
0	Transmission is in progress [Clearing condition] Software reads TDRE while it is set to 1, then writes 0 in the TDRE flag
1	End of transmission (Ini [Setting conditions] • The chip is reset or enters standby mode. The TE bit is cleared to 0 in S • TDRE is 1 when the last bit of a serial character is transmitted

Note: * If the RE bit is cleared to 0 when a multiprocessor format is selected, MPB previous value.

Bit 0—Multiprocessor Bit Transfer (MPBT): Stores the value of the multiprocesso transmit data when a multiprocessor format is selected for transmitting in asynchronou The MPBT setting is ignored in synchronous mode, when a multiprocessor format is n or when the SCI is not transmitting.

Bit 0 MPBT	Description	
0	Multiprocessor bit value in transmit data is 0	(I
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 sele rate generator clock source, determines the serial communication bit rate.

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

The CPU can always read and write BRR. BRR is initialized to H'FF by a reset and in mode.

The baud rate generator is controlled separately for the individual channels, so differe may be set for each.

Table 11.3 shows examples of BRR settings in asynchronous mode. Table 11.4 shows BRR settings in synchronous mode.

300	0	207	0.16	0	217	0.21	0	255	0	1	77
600	0	103	0.16	0	108	0.21	0	127	0	0	155
1200	0	51	0.16	0	54	–0.70	0	63	0	0	77
2400	0	25	0.16	0	26	1.14	0	31	0	0	38
4800	0	12	0.16	0	13	–2.48	0	15	0	0	19
9600	0	6	–6.99	0	6	–2.48	0	7	0	0	9
19200	0	2	8.51	0	2	13.78	0	3	0	0	4
31250	0	1	0	0	1	4.86	0	1	22.88	0	2
38400	0	1	–18.62	0	1	–14.67	0	1	0	—	—

	ϕ (MHz)										
	3.6864			**4**			**4.9152**				
Bit Rate (bits/s)	**n**	**N**	**Error (%)**	**n**	**N**	**Error (%)**	**n**	**N**	**Error (%)**	**n**	**N**
110	2	64	0.70	2	70	0.03	2	86	0.31	2	88
150	1	191	0	1	207	0.16	1	255	0	2	64
300	1	95	0	1	103	0.16	1	127	0	1	129
600	0	191	0	0	207	0.16	0	255	0	1	64
1200	0	95	0	0	103	0.16	0	127	0	0	129
2400	0	47	0	0	51	0.16	0	63	0	0	64
4800	0	23	0	0	25	0.16	0	31	0	0	32
9600	0	11	0	0	12	0.16	0	15	0	0	15
19200	0	5	0	0	6	–6.99	0	7	0	0	7
31250	—	—	—	0	3	0	0	4	–1.70	0	4
38400	0	2	0	0	2	8.51	0	3	0	0	3

600	1	77	0.16	1	79	0	1	95	0	1	103
1200	0	155	0.16	0	159	0	0	191	0	0	207
2400	0	77	0.16	0	79	0	0	95	0	0	103
4800	0	38	0.16	0	39	0	0	47	0	0	51
9600	0	19	–2.34	0	19	0	0	23	0	0	25
19200	0	9	–2.34	0	9	0	0	11	0	0	12
31250	0	5	0	0	5	2.40	0	6	5.33	0	7
38400	0	4	–2.34	0	4	0	0	5	0	0	6

	ϕ (MHz)										
	9.8304			**10**			**12**			**1**	
Bit Rate (bits/s)	**n**	**N**	**Error (%)**	**n**	**N**	**Error (%)**	**n**	**N**	**Error (%)**	**n**	**N**
110	2	174	–0.26	2	177	–0.25	2	212	0.03	2	217
150	2	127	0	2	129	0.16	2	155	0.16	2	159
300	1	255	0	2	64	0.16	2	77	0.16	2	79
600	1	127	0	1	129	0.16	1	155	0.16	1	159
1200	0	255	0	1	64	0.16	1	77	0.16	1	79
2400	0	127	0	0	129	0.16	0	155	0.16	0	159
4800	0	63	0	0	64	0.16	0	77	0.16	0	79
9600	0	31	0	0	32	–1.36	0	38	0.16	0	39
19200	0	15	0	0	15	1.73	0	19	–2.34	0	19
31250	0	9	–1.70	0	9	0	0	11	0	0	11
38400	0	7	0	0	7	1.73	0	9	–2.34	0	9

600	1	181	0.16	1	191	0	1	207	0.16	1	233
1200	1	90	0.16	1	95	0	1	103	0.16	1	116
2400	0	181	0.16	0	191	0	0	207	0.16	0	233
4800	0	90	0.16	0	95	0	0	103	0.16	0	116
9600	0	45	–0.93	0	47	0	0	51	0.16	0	58
19200	0	22	–0.93	0	23	0	0	25	0.16	0	28
31250	0	11	0	0	14	–1.70	0	15	0	0	17
38400	0	10	3.57	0	11	0	0	12	0.16	0	14

1 k	1	124	1	249	2	124	—	—	2	249
2.5 k	0	199	1	99	1	199	1	249	2	99
5 k	0	99	0	199	1	99	1	124	1	199
10 k	0	49	0	99	0	199	0	249	1	99
25 k	0	19	0	39	0	79	0	99	0	159
50 k	0	9	0	19	0	39	0	49	0	79
100 k	0	4	0	9	0	19	0	24	0	39
250 k	0	1	0	3	0	7	0	9	0	15
500 k	0	0*	0	1	0	3	0	4	0	7
1 M			0	0*	0	1	—	—	0	3
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.

$$N = \frac{\phi}{8 \times 2^{2n-1} \times B} \times 10^6 - 1$$

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.

$$\text{Error (\%)} = \{\frac{\phi \times 10^6}{(N + 1) \times B \times 64 \times 2^{2n-1}} - 1\} \times 100$$

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

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

12	2.0000	2000000.0
14	2.3333	2333333.3
16	2.6667	2666666.7
18	3.0000	3000000.0

selected by the C/A bit in SMR and the CKE1 and CKE0 bits in SCR, as shown in table

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 selections determine the communication format and character length.
- In receiving, it is possible to detect framing errors, parity errors, overrun errors, and 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 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 frequen 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 and outputs a serial clock signal to external devices.
  - When an external clock is selected, the SCI operates on the input serial clock. T baud rate generator is not used.

0	0	0	1	0				Pres
0	0	0	1	1				
0	1	0	0	0		7-bit data		Abse
0	1	0	0	1				
0	1	0	1	0				Pres
0	1	0	1	1				
0	0	1	—	0	Asynchronous mode (multi-processor format)	8-bit data	Present	Abse
0	0	1	—	1				
0	1	1	—	0		7-bit data		
0	1	1	—	1				
1	—	—	—	—	Synchronous mode	8-bit data	Absent	

Table 11.9 SMR and SCR Settings and SCI Clock Source Selection

SMR	SCR Settings			SCI Communication Form	
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 th
0	0	1			Outputs a clock wit matching the bit rat
0	1	0		External	Inputs a clock with 16 times the bit rate
0	1	1			
1	0	0	Synchronous mode	Internal	Outputs the serial c
1	0	1			
1	1	0		External	Inputs the serial clo
1	1	1			

Figure 11.2 shows the general format of asynchronous serial communication. In asynch
serial communication the communication line is normally held in the mark (high) state.
monitors the line and starts serial communication when the line goes to the space (low)
indicating a start bit. One serial character consists of a start bit (low), data (LSB first), p
(high or low), and stop bit (high), in that order.

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

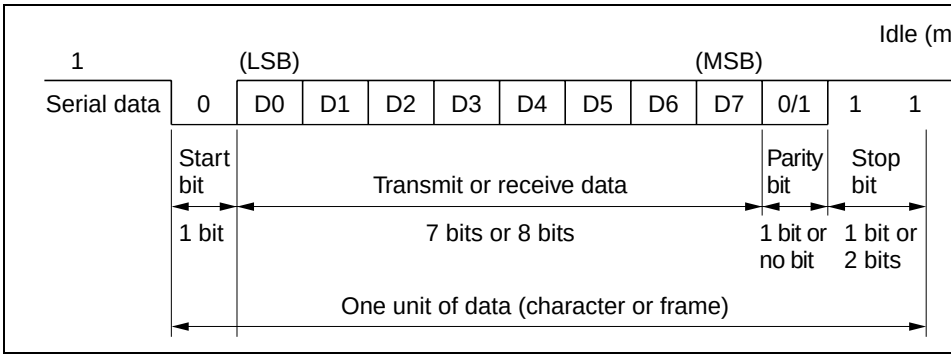


**Figure 11.2 Data Format in Asynchronous Communication (Example: 8-Bit D
Parity and 2 Stop Bits)**

0	0	0	0	S	8-bit data		STOP	
0	0	0	1	S	8-bit data		STOP	ST
0	1	0	0	S	8-bit data		P	ST
0	1	0	1	S	8-bit data		P	ST
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 ST
0	—	1	0	S	8 bit data		MPB	ST
0	—	1	1	S	8 bit data		MPB	ST
1	—	1	0	S	7-bit data		MPB	STOP
1	—	1	1	S	7-bit data		MPB	STOP ST

Legend:
S: Start bit
STOP: Stop bit
P: Parity bit
MPB: Multiprocessor bit

When the SCI operates on an internal clock, it can output a clock signal at the SCK pin frequency of this output clock is equal to the bit rate. The phase is aligned as in figure 1 the rising edge of the clock occurs at the center of each transmit data bit.

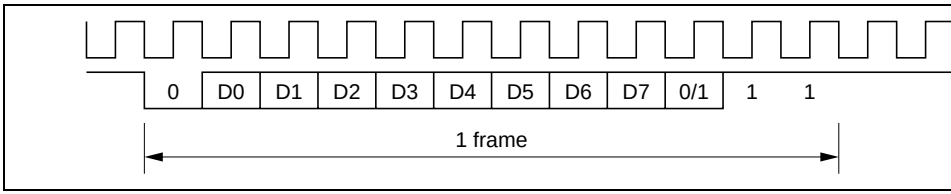


Figure 11.3 Phase Relationship between Output Clock and Serial Data (Asynchronous Mode)

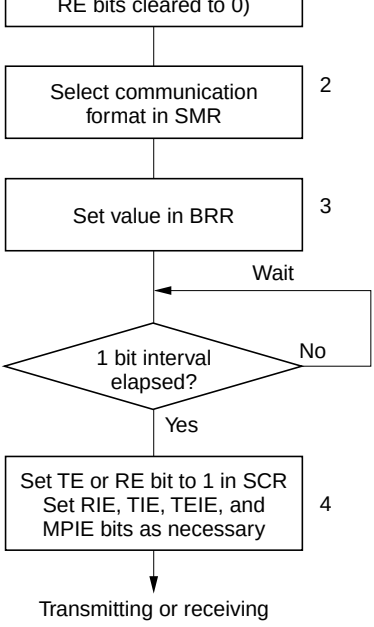
Transmitting and Receiving Data

SCI Initialization (Asynchronous Mode): Before transmitting or receiving, clear the bits to 0 in SCR, then initialize the SCI as follows.

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

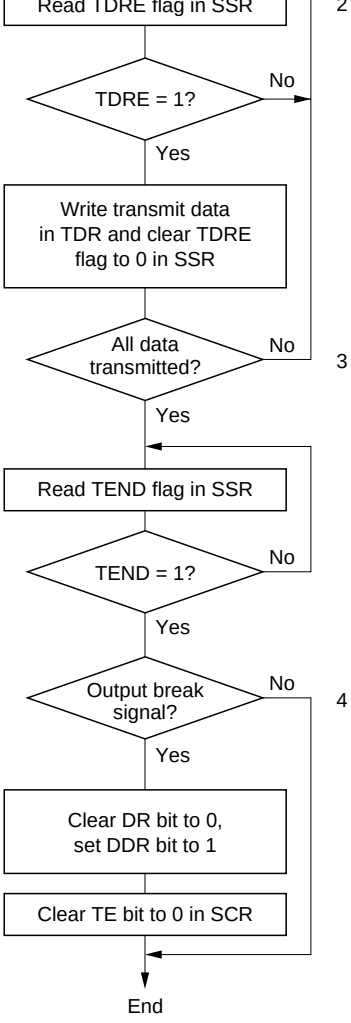
When an external clock is used, the clock should not be stopped during initialization or operation. SCI operation becomes unreliable if the clock is stopped.

Figure 11.4 shows a sample flowchart for initializing the SCI.



TIE, TEIE, and MPIE bits as necessary. Setting
or RE bit enables the SCI to use the TxD or Rx

Figure 11.4 Sample Flowchart for SCI Initialization



written, write data in TDR, then clear the TDRE flag to 0.

4. To output a break signal at the end of serial transm set the DDR bit to 1 and clear the DR bit to 0 (DDR and DR are I/O port registers), then clear th TE bit to 0 in SCR.

Figure 11.5 Sample Flowchart for Transmitting Serial Data

— 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 mu bit is output. Formats in which neither a parit a multiprocessor bit is output can also be sele

— Stop bit: One or two 1 bits (stop bits) are output.

— Mark state: Output of 1 continues until the start bit of the transmit data.

- The SCI checks the TDRE flag when it outputs the stop bit. If the TDRE flag is 0, loads new data from TDR into TSR, outputs the stop bit, then begins serial transm next frame. If the TDRE flag is 1, the SCI sets the TEND flag to 1 in SSR, outputs then continues output of 1 in the mark state. If the TEIE bit is set to 1 in SCR, a tra interrupt (TEI) is requested at this time.

Figure 11.6 shows an example of SCI transmit operation in asynchronous mode.

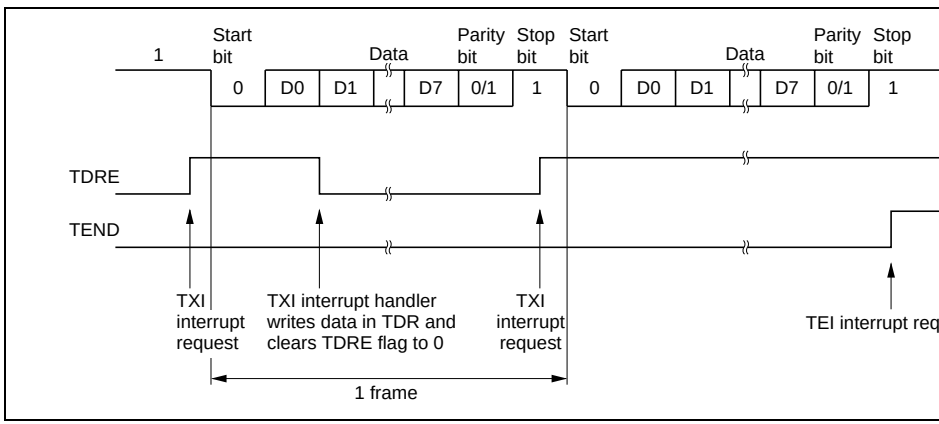
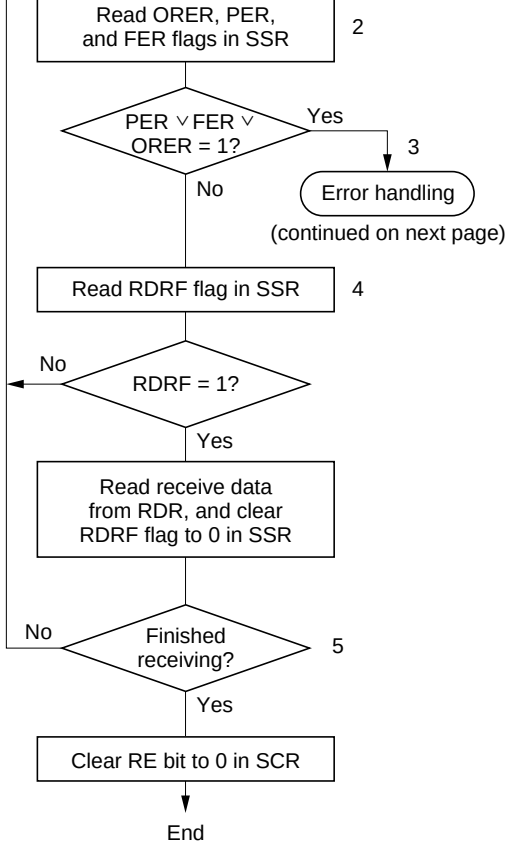


Figure 11.6 Example of SCI Transmit Operation in Asynchronous Mo (8-Bit Data with Parity and 1 Stop Bit)



handling, clear the ORER, PER, and
flags all to 0. Receiving cannot resu
of the ORER, PER, and FER flags re
set to 1. When a framing error occu
RxD pin can be read to detect the br

4. SCI status check and receive data re
SSR, check that RDRF is set to 1, th
receive data from RDR and clear the
flag to 0. Notification that the RDRF
changed from 0 to 1 can also be give
RXI interrupt.
5. To continue receiving serial data: ch
RDRF flag, read RDR, and clear the
flag to 0 before the stop bit of the cu
frame is received.

Figure 11.7 Sample Flowchart for Receiving Serial Data (1)

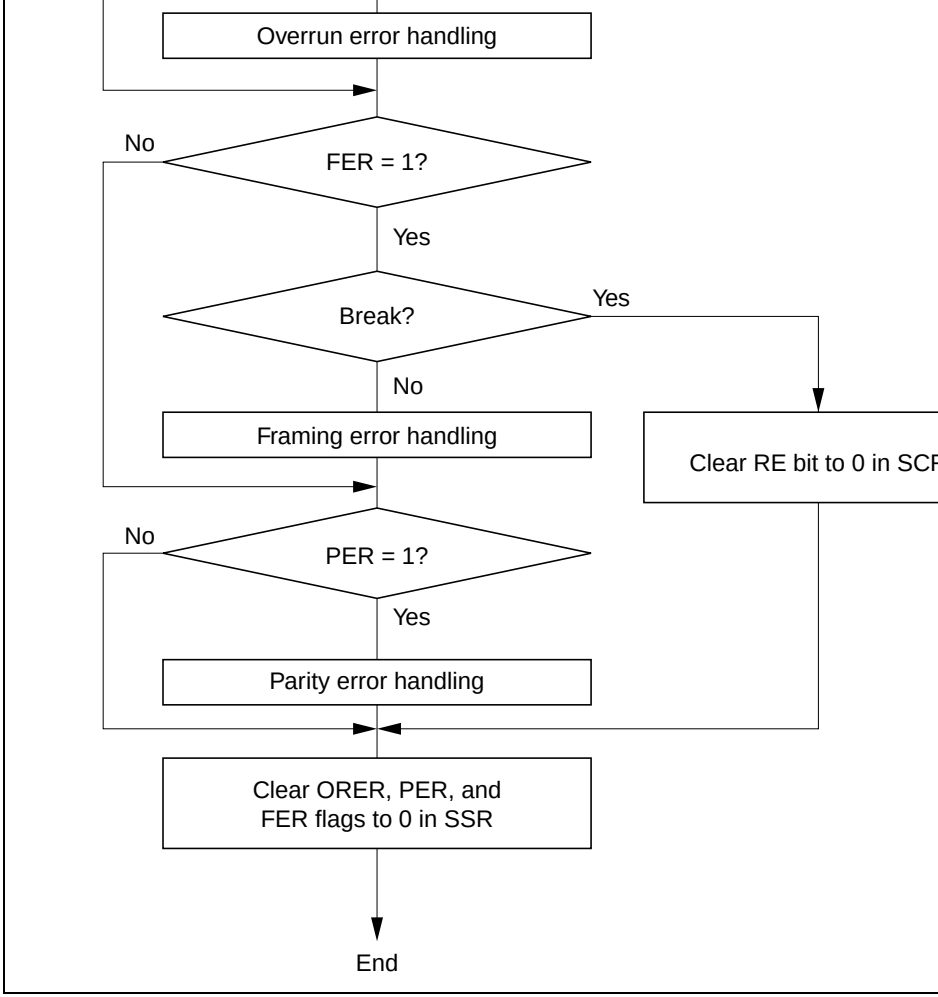


Figure 11.7 Sample Flowchart for Receiving Serial Data (2)

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 bit is checked.

— Status check: The RDRF flag must be 0 so that receive data can be transferre RSR into RDR.

If these checks all pass, the RDRF flag is set to 1 and the received data is stored in I 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 I 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 (RXI) is requested. If the ORER, PER, or FER flag is set to 1 and the RIE bit in SC set to 1, a receive-error interrupt (ERI) is requested.

Table 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 t from RSR to RDR
Framing error	FER	Stop bit is 0	Receive data tran from RSR to RDR
Parity error	PER	Parity of receive data differs from even/odd parity setting in SMR	Receive data tran from RSR to RDR

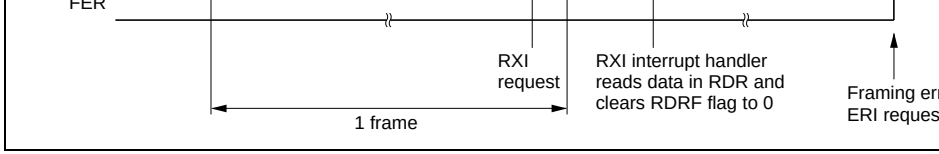


Figure 11.8 Example of SCI Receive Operation (8-Bit Data with Parity and On

11.3.3 Multiprocessor Communication

The multiprocessor communication function enables several processors to share a sing communication line. The processors communicate in asynchronous mode using a form additional multiprocessor bit (multiprocessor format).

In multiprocessor communication, each receiving processor is addressed by an ID. A s communication cycle consists of an ID-sending cycle that identifies the receiving proc data-sending cycle. The multiprocessor bit distinguishes ID-sending cycles from data- cycles.

The transmitting processor starts by sending the ID of the receiving processor with wh to communicate as data with the multiprocessor bit set to 1. Next the transmitting proc transmit data with the multiprocessor bit cleared to 0.

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

Figure 11.9 shows an example of communication among different processors using a multiprocessor format.

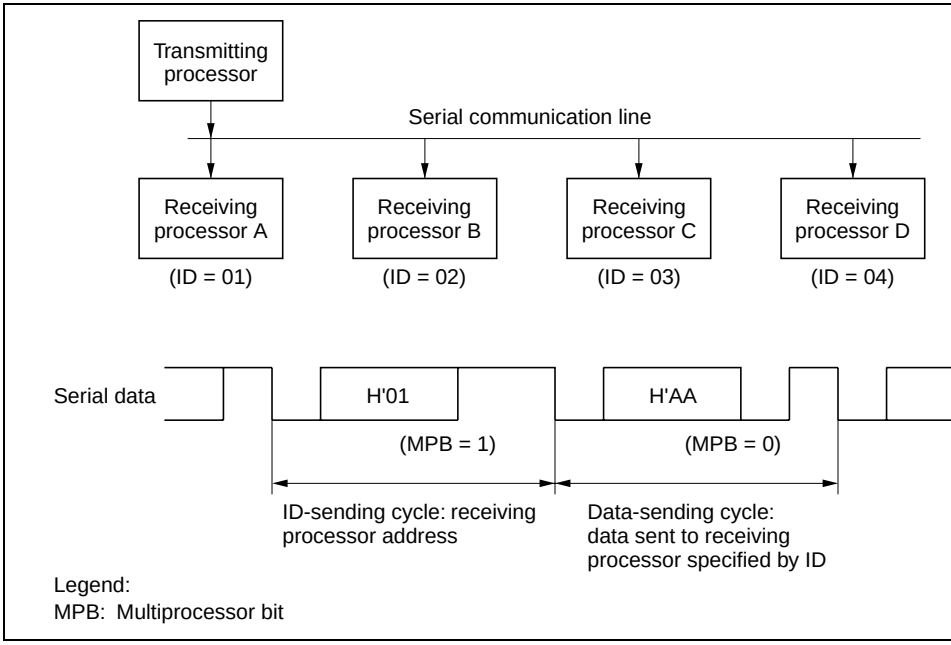
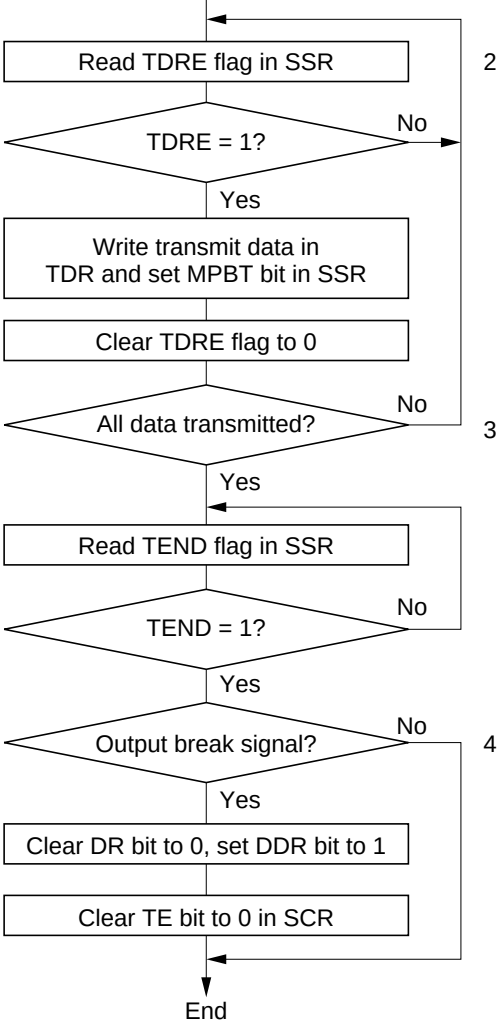


Figure 11.9 Example of Communication among Processors using Multiprocesso (Sending Data H'AA to Receiving Processor A)



2. SCI status check and transmi
write: read SSR, check that th
flag is 1, then write transmit
data in TDR. Also set the MPI
0 or 1 in SSR. Finally, clear th
flag to 0.
3. To continue transmitting seria
after checking that the TDRE
indicating that data can be wr
write data in TDR, then clear
flag to 0.
4. To output a break signal at th
serial transmission: set the DI
1 and clear the DR bit to 0 (D
DR are I/O port registers), the
the TE bit to 0 in SCR.

Figure 11.10 Sample Flowchart for Transmitting Multiprocessor Serial

— 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 tran

- The SCI checks the TDRE flag when it outputs the stop bit. If the TDRE flag is 0, t loads data from TDR into TSR, outputs the stop bit, then begins serial transmission frame. If the TDRE flag is 1, the SCI sets the TEND flag in SSR to 1, outputs the st continues output of 1 bits in the mark state. If the TEIE bit is set to 1 in SCR, a tran interrupt (TEI) is requested at this time.

Figure 11.11 shows an example of SCI transmit operation using a multiprocessor forma

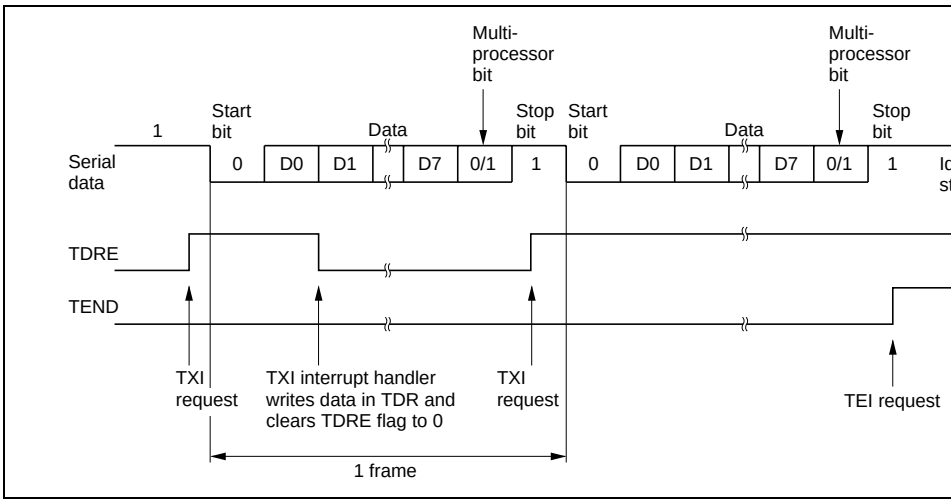
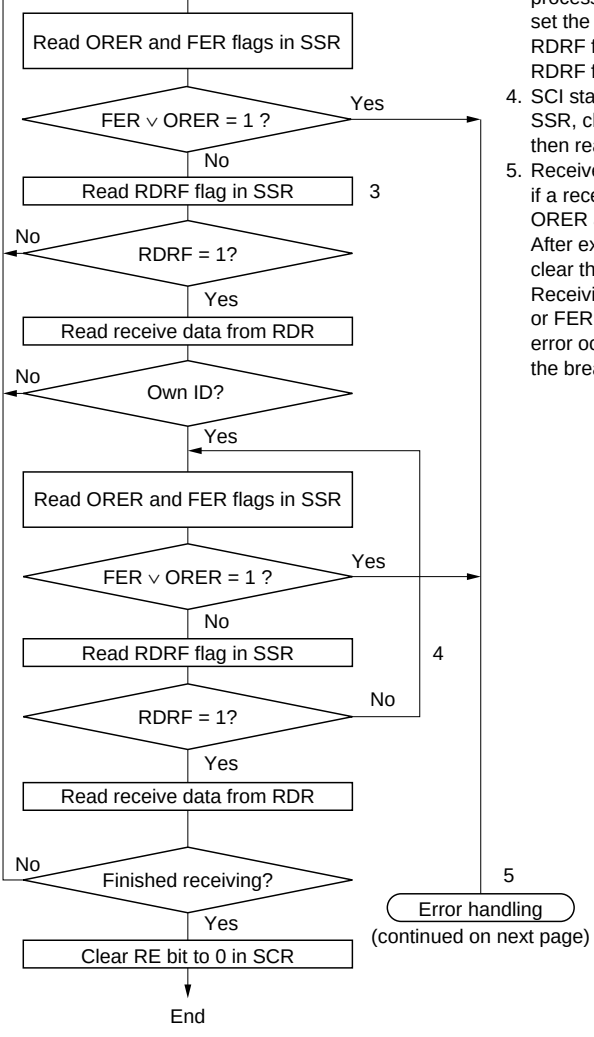


Figure 11.11 Example of SCI Transmit Operation (8-Bit Data with Multiprocess One Stop Bit)



4. SCI status check and data receiving: re
SSR, check that the RDRF flag is set to
then read data from RDR.
5. Receive error handling and break detec
if a receive error occurs, read the
ORER and FER flags in SSR to identify
After executing the necessary error han
clear the ORER and FER flags both to 0
Receiving cannot resume while either th
or FER flag remains set to 1. When a fr
error occurs, the RxD pin can be read to
the break state.

Figure 11.12 Sample Flowchart for Receiving Multiprocessor Serial Dat

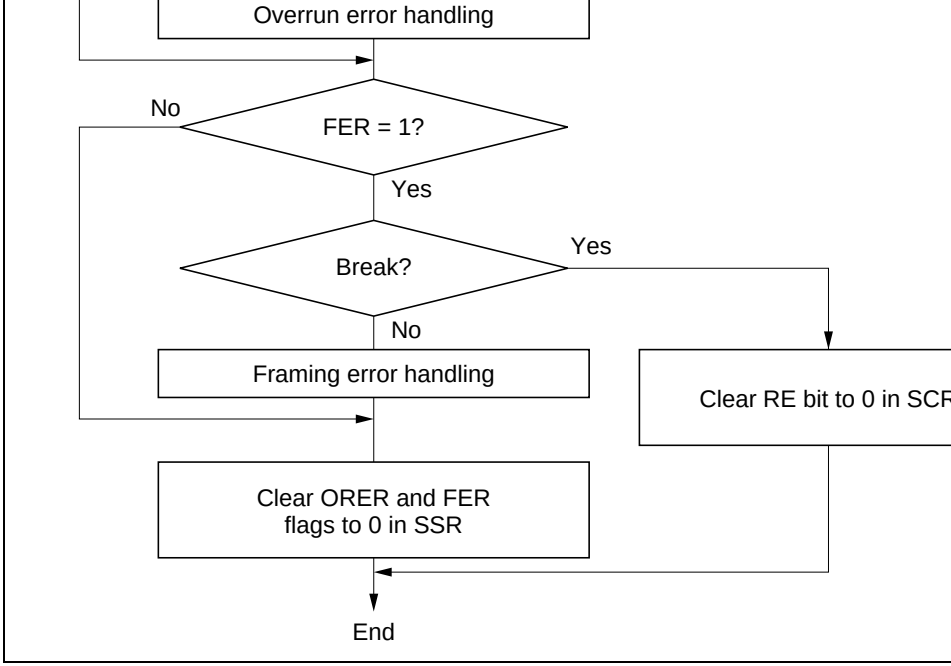


Figure 11.12 Sample Flowchart for Receiving Multiprocessor Serial Data

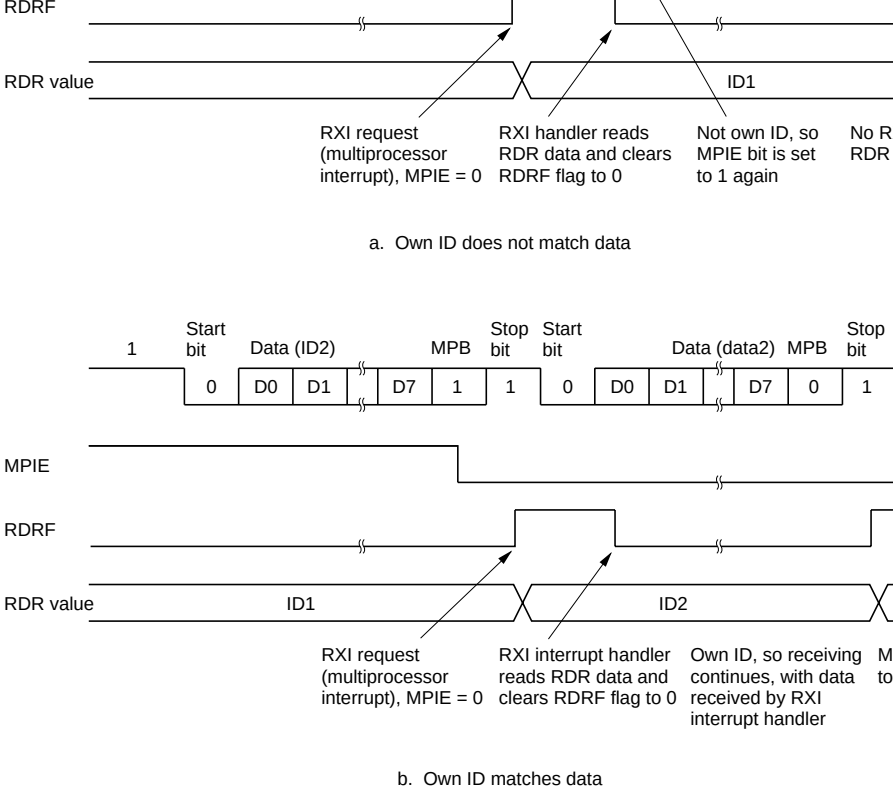


Figure 11.13 Example of SCI Receive Operation (8-Bit Data with Multiproces One Stop Bit)

Figure 11.14 shows the general format in synchronous serial communication.

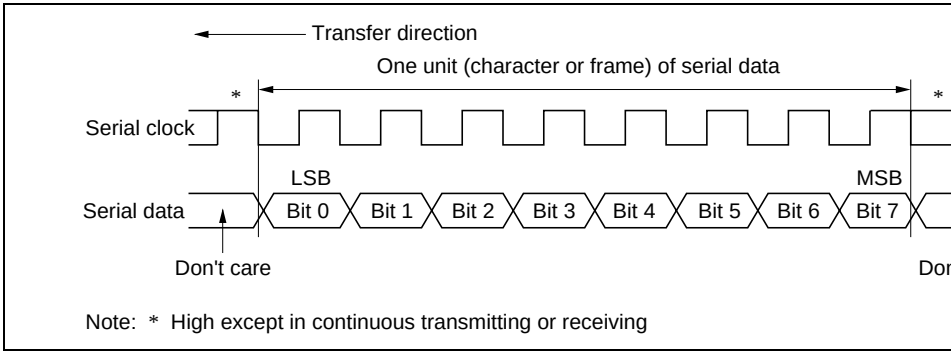


Figure 11.14 Data Format in Synchronous Communication

In synchronous serial communication, each data bit is placed on the communication lin falling edge of the serial clock to the next. Data is guaranteed valid at the rise of the ser In each character, the serial data bits are transmitted in order from LSB (first) to MSB (output of the MSB, the communication line remains in the state of the MSB. In synchro the SCI receives data by synchronizing with the rise of the serial clock.

Communication Format

The data length is fixed at 8 bits. No parity bit or multiprocessor bit can be added.

Clock

An internal clock generated by the on-chip baud rate generator or an external clock inp SCK pin can be selected by clearing or setting the CKE1 and CKE0 bits in SCR and th SMR. See table 11.9. When the SCI operates on an internal clock, it outputs the clock s SCK pin. Eight clock pulses are output per transmitted or received character. When the transmitting or receiving, the clock signal remains in the high state.

Figure 11.15 shows a sample flowchart for initializing the SCI.

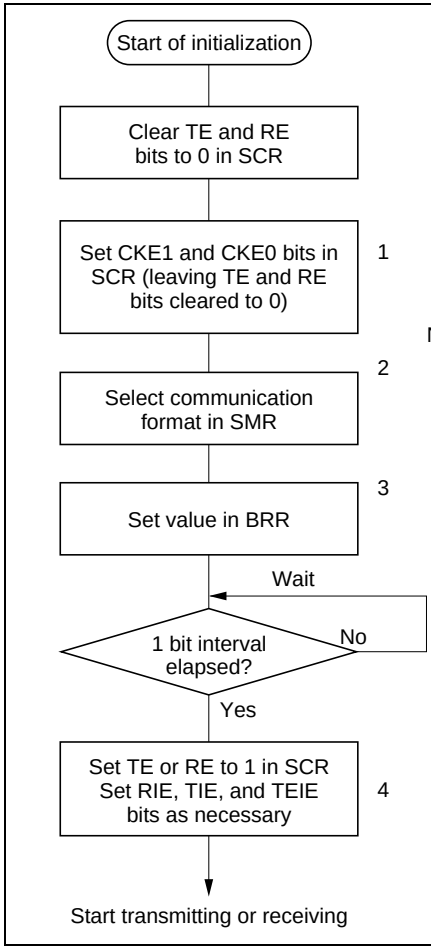
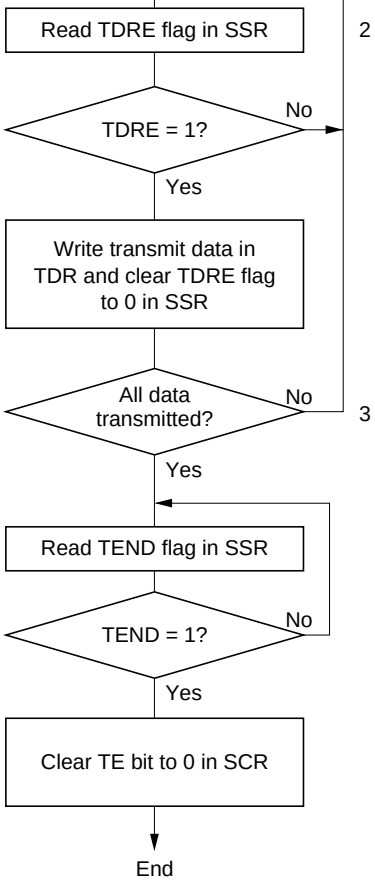


Figure 11.15 Sample Flowchart for SCI Initialization



that the TDRE flag is 1, indicating that data written, write data in TDR, then clear the TD to 0.

Figure 11.16 Sample Flowchart for Serial Transmitting

is selected, the SCI outputs data in synchronization with the input clock. Data is ou
the TxD pin in order from LSB (bit 0) to MSB (bit 7).

- The SCI checks the TDRE flag when it outputs the MSB (bit 7). If the TDRE flag
loads data from TDR into TSR and begins serial transmission of the next frame. If
flag is 1, the SCI sets the TEND flag to 1 in SSR, and after transmitting the MSB,
TxD pin in the MSB state. If the TEIE bit in SCR is set to 1, a transmit-end interru
requested at this time.
- 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.

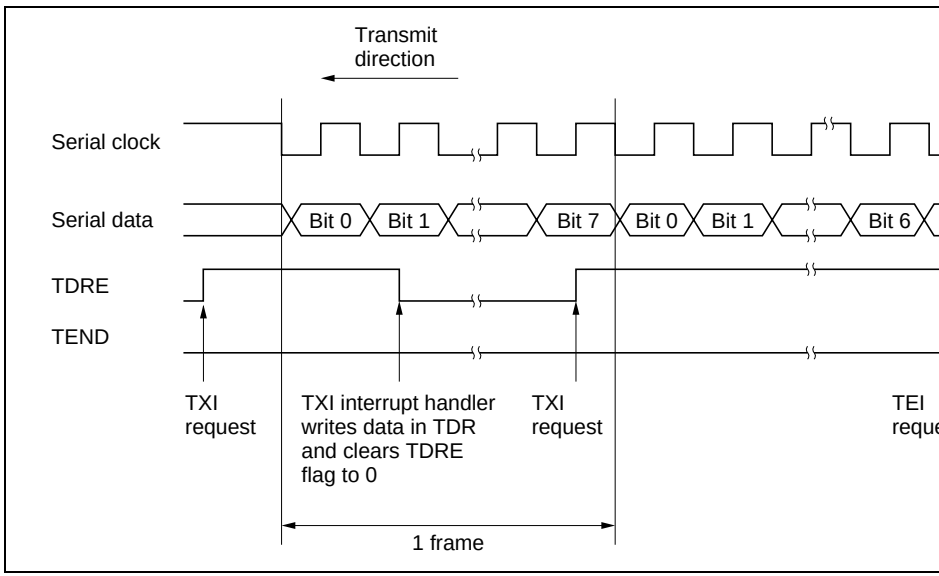
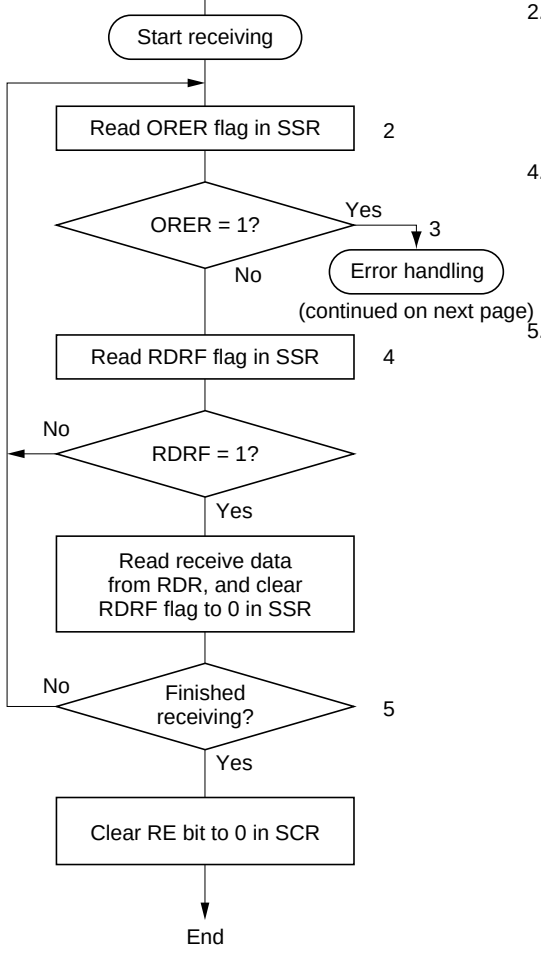


Figure 11.17 Example of SCI Transmit Operation



2., 3. Receive error handling: if a receive
occurs, read the ORER flag in SSR,
executing the necessary error handl
the ORER flag to 0. Neither transmit
receiving can resume while the ORI
remains set to 1.

4. SCI status check and receive data r
SSR, check that the RDRF flag is se
then read receive data from RDR an
the RDRF flag to 0. Notification that
flag has changed from 0 to 1 can als
given by the RXI interrupt.

5. To continue receiving serial data: ch
RDRF flag, read RDR, and clear the
flag to 0 before the MSB (bit 7) of th
frame is received.

Figure 11.18 Sample Flowchart for Serial Receiving (1)

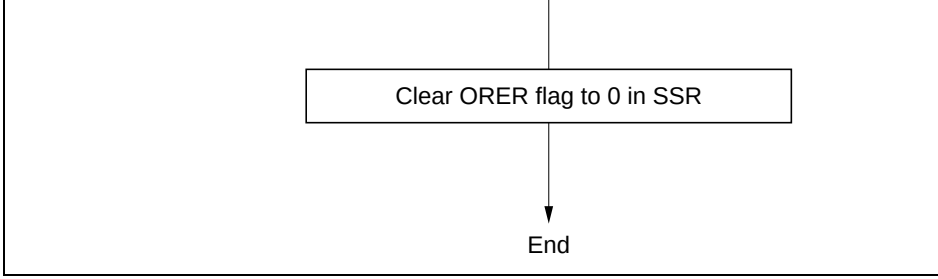


Figure 11.18 Sample Flowchart for Serial Receiving (2)

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 dat transferred from RSR to RDR. If this check passes, the RDRF flag is set to 1 and t data is stored in RDR. If the check does not pass (receive error), the SCI operates a in table 11.11. If any receive error is detected, the subsequent data transmission/re disabled.
- After setting the RDRF flag to 1, if the RIE bit is set to 1 in SCR, the SCI requests data-full interrupt (RXI). If the ORER flag is set to 1 and the RIE bit in SCR is als the SCI requests a receive-error interrupt (ERI).

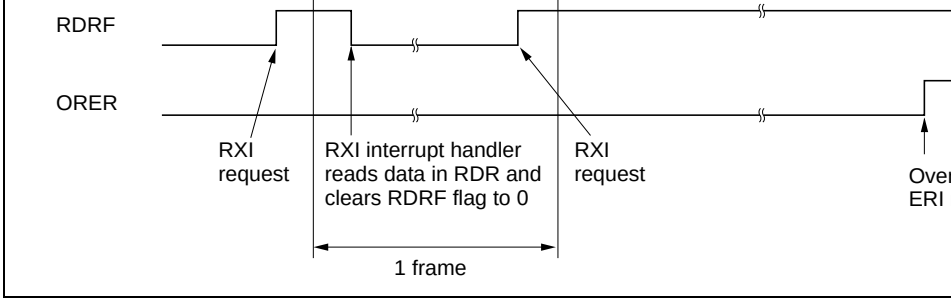
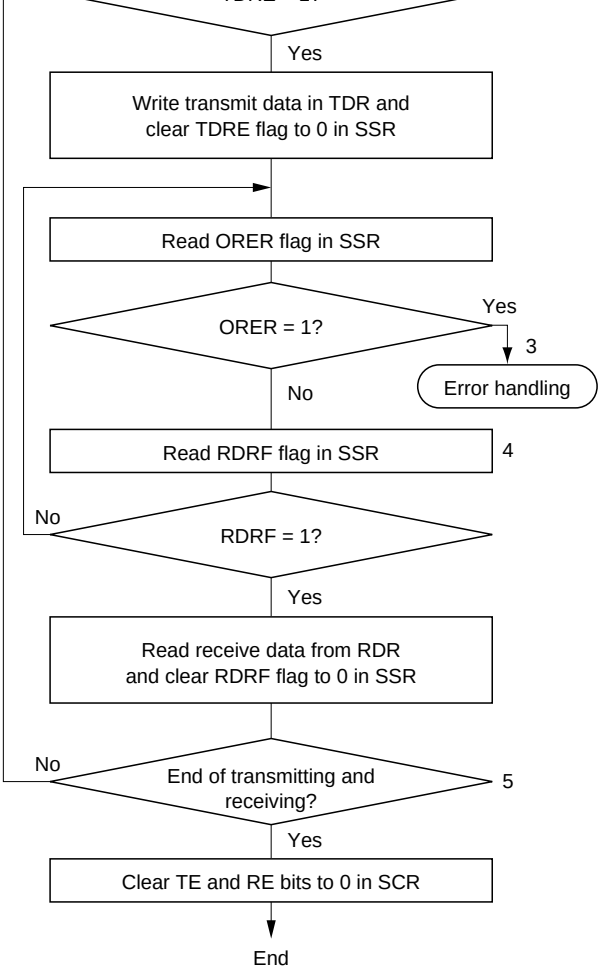


Figure 11.19 Example of SCI Receive Operation

Transmitting and Receiving Serial Data Simultaneously (Synchronous Mode): Fig shows a sample flowchart for transmitting and receiving serial data simultaneously and the procedure to follow.



Note: When switching from transmitting or receiving to simultaneous transmitting and receiving, clear the TE and RE bits both to 0, then set the TE and RE bits both to 1.

3. Receive error handling: if a error occurs, read the ORE SSR, then after executing sary error handling, clear t flag to 0.
   Neither transmitting nor re can resume while the ORE remains set to 1.
4. SCI status check and rece data read: read SSR, chec the RDRF flag is 1, then re receive data from RDR an the RDRF flag to 0. Notific that the RDRF flag has ch from 0 to 1 can also be giv by the RXI interrupt.
5. To continue transmitting a receiving serial data: chec RDRF flag, read RDR, and the RDRF flag to 0 before MSB (bit 7) of the current f is received. Also check tha the TDRE flag is set to 1, i ing that data can be writte data in TDR, then clear the flag to 0 before the MSB (b the current frame is transm

Figure 11.20 Sample Flowchart for Serial Transmitting

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 interru
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

Data can be written into TDR regardless of the state of the TDRE flag. If new data is w
TDR when the TDRE flag is 0, the old data stored in TDR will be lost because this da
yet been transferred to TSR. Before writing transmit data in TDR, be sure to check tha
flag is set to 1.

Simultaneous Multiple Receive Errors

Table 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
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 +

Notes: ○: Receive data is transferred from RSR to RDR.
×: Receive data is not transferred from RSR to RDR.

When the TE bit is cleared to 0 the TxD pin becomes an I/O port, the level and directio output) of which are determined by DR and DDR bits. This feature can be used to send signal.

After the serial transmitter is initialized, the DR value substitutes for the mark state unt bit is set to 1 (the TxD pin function is not selected until the TE bit is set to 1). The DDR bits should therefore both be set to 1 beforehand.

To send a break signal during serial transmission, clear the DR bit to 0, then clear the T When the TE bit is cleared to 0 the transmitter is initialized, regardless of its current sta TxD pin becomes an 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 transm even if the TDRE flag is cleared to 0. Be sure to clear the receive error flags to 0 when transmit. Note that clearing the RE bit to 0 does not clear the receive error flags to 0.

Receive Data Sampling Timing in Asynchronous Mode and Receive Margin

In asynchronous mode the SCI operates on a base clock with 16 times the bit rate frequ receiving, the SCI synchronizes internally with the fall of the start bit, which it samples base clock. Receive data is latched at the rising edge of the eighth base clock pulse. See 11.21.

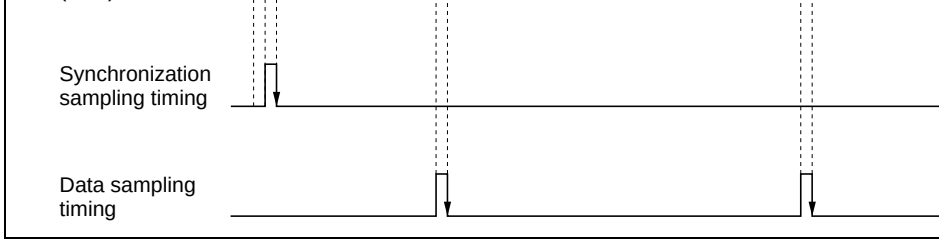


Figure 11.21 Receive Data Sampling Timing in Asynchronous Mode

The receive margin in asynchronous mode can therefore be expressed as shown in equ

$$M = \left| \left(0.5 - \frac{1}{2N} \right) - (L - 0.5)\, F \right| \times 100\% \quad \text{.....(1)}$$

$$\frac{|D - 0.5|}{N} (1 + F) \Big|$$

M: Receive margin (%)
N: Ratio of clock frequency to bit rate (N = 16)
D: Clock duty cycle (D = 0 to 1.0)
L: Frame length (L = 9 to 12)
F: Absolute deviation of clock frequency

From equation (1), if F = 0 and D = 0.5 the receive margin is 46.875%, as given by eq

When D = 0.5, F = 0:

$$M = [0.5 - 1/(2 \times 16)] \times 100\%$$
$$= 46.875\% \text{.....(2)}$$

This is a theoretical value. A reasonable margin to allow in system design is 20% to 3

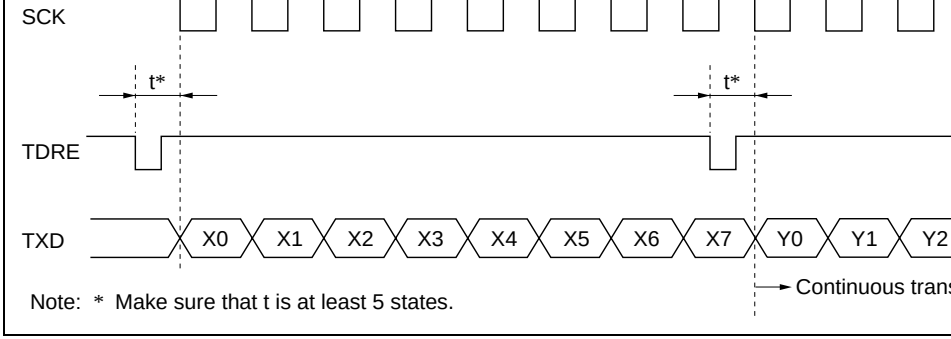


Figure 11.22 Transmission in Synchronous Mode (Example)

(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)

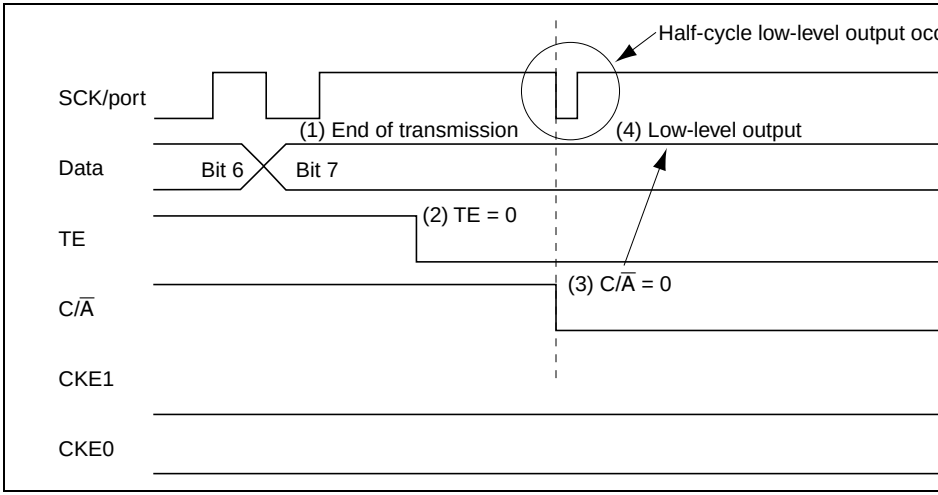


Figure 11.23 Operation when Switching from SCK Pin Function to Port Pin

(2) TE bit = 0

(3) CKE1 bit = 1

(4) C/$\overline{A}$ bit = 0 ... switchover to port output

(5) CKE1 bit = 0

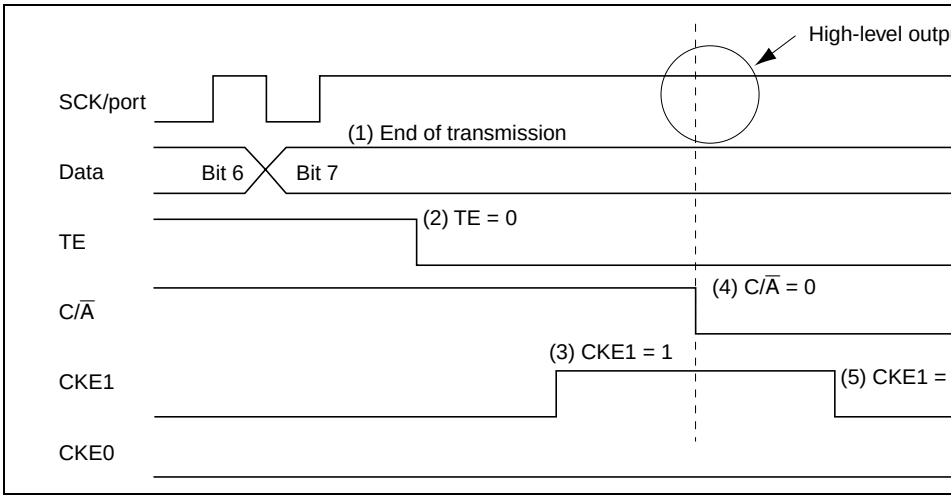


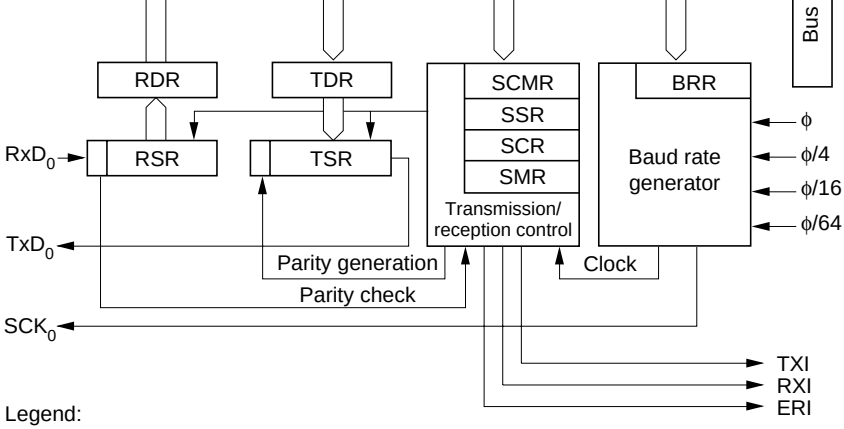
Figure 11.24 Operation when Switching from SCK Pin Function to Port Pin F (Preventing Low-Level Output)

Switching between the normal serial communication interface and the Smart Card inte
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/rec
    that can issue requests independently



Legend:
SCMR: Smart Card mode register
RSR: Receive shift register
RDR: Receive data register
TSR: Transmit shift register
TDR: Transmit data register
SMR: Serial mode register
SCR: Serial control register
SSR: Serial status register
BRR: Bit rate register

Figure 12.1 Block Diagram of Smart Card Interface

Transmit data pin 0	TxD_0	Output	SCI_0 transmit data outpu

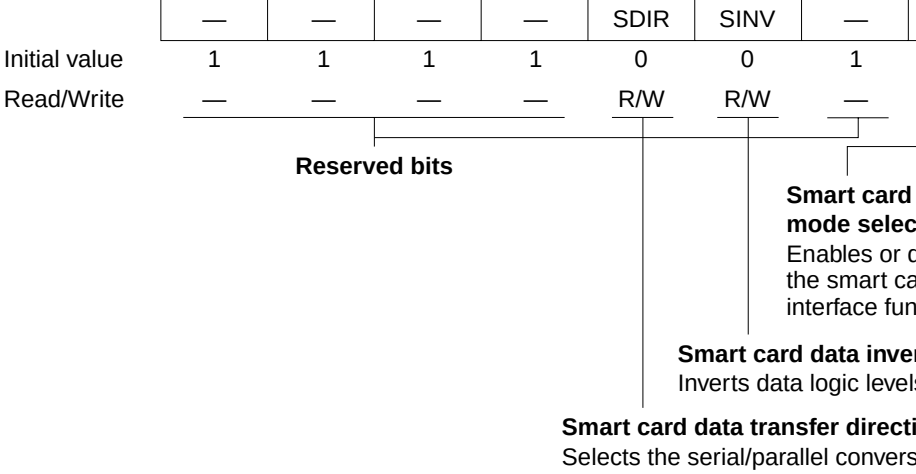
12.1.4 Register Configuration

Table 12.2 shows the registers used by the Smart Card interface. Details of SMR, BRI TDR, and RDR are the same as for the normal SCI function: see the register descriptio section 11, Serial Communication Interface.

Table 12.2 Smart Card Interface Registers

Address*[1]	Name	Abbreviation	R/W	Initial
H'FFB0	Serial mode register	SMR	R/W	H'00
H'FFB1	Bit rate register	BRR	R/W	H'FF
H'FFB2	Serial control register	SCR	R/W	H'00
H'FFB3	Transmit data register	TDR	R/W	H'FF
H'FFB4	Serial status register	SSR	R/(W)*[2]	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.



SCMR is an 8-bit readable/writable register that selects the Smart Card interface functi

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 conv
format.

Bit 3 SDIR	Description	
0	TDR contents are transmitted LSB-first	(In
	Receive data is stored in RDR LSB-first	
1	TDR contents are transmitted MSB-first	
	Receive data is stored in RDR MSB-first	

	Receive data is stored as it is in RDR
1	TDR contents are inverted before being transmitted Receive data is stored in inverted form in RDR

Bit 1—Reserved: 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 interface function.

Bit 0 SMIF	Description	
0	Smart Card interface function is disabled	(I
1	Smart Card interface function is enabled	

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, t
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
Status Register (SSR).

Bit 4—Error Signal Status (ERS): In Smart Card interface mode, bit 4 indicates the s
error signal sent back from the receiving end in transmission. Framing errors are not de
Smart Card interface mode.

Bit 4 ERS	Description
0	Indicates normal data transmission, with no error signal returned [Clearing conditions] (In • 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 err [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 pr
state.

	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 trans 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 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 o period, 10.5 etu after the start bit.
- If the error signal is sampled during transmission, the same data is transmitted auto after the elapse of 2 etu or longer.
- Only asynchronous communication is supported; there is no clocked synchronous communication function.

is input to the CLK pin of the IC card. No connection is needed if the IC card uses an in
clock.

LSI port output is used as the reset signal.

Other pins must normally be connected to the power supply or ground.

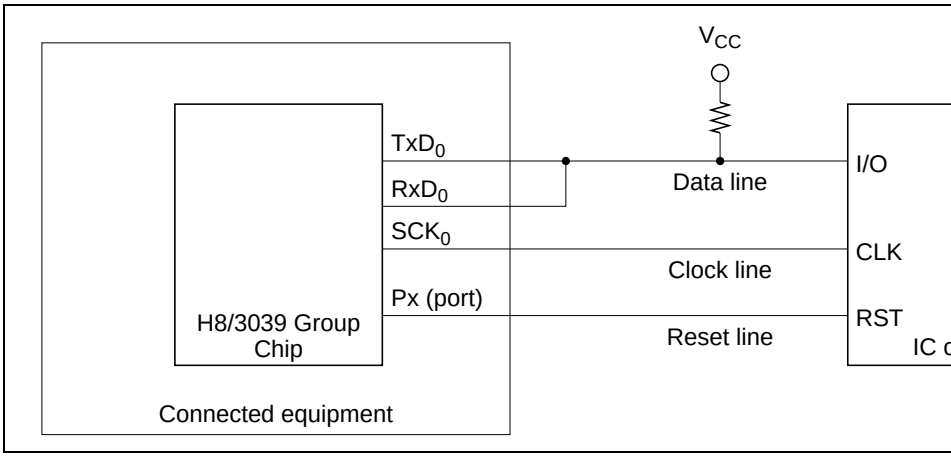


Figure 12.2 Schematic Diagram of Smart Card Interface Pin Connectio

Note: If an IC card is not connected, and the TE and RE bits are both set to 1, closed transmission/reception is possible, enabling self-diagnosis to be carried out.

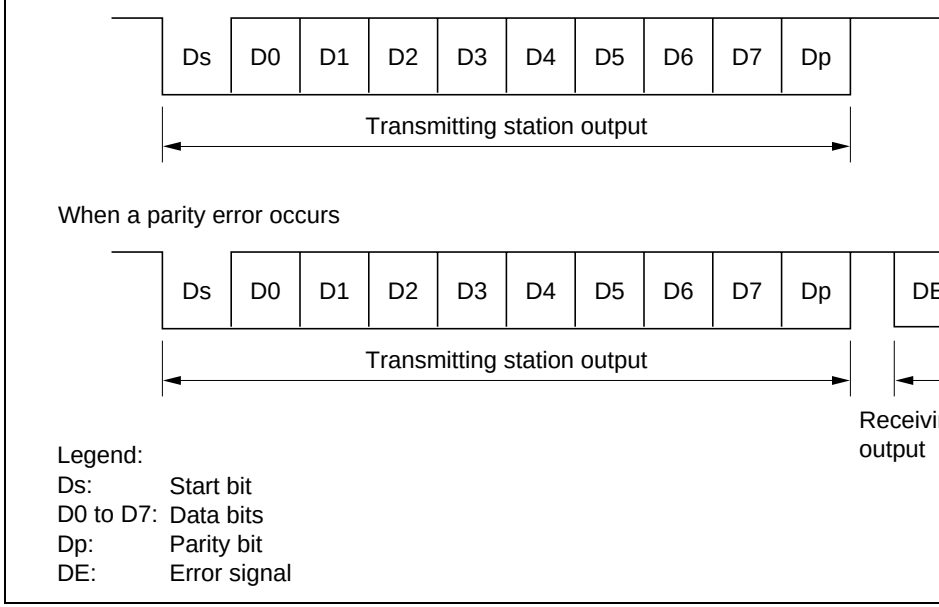


Figure 12.3 Smart Card Interface Data Format

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 wa reception of the next data.

If a parity error occurs, however, the receiving station outputs an error signal (DE, l to request retransmission of the data. After outputting the error signal for the prescr of time, the receiving station places the signal line in the high-impedance state agai 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 th frame.

If it does receive an error signal, however, it returns to step [2] and retransmits the e data.

	Bit						
Register	**Bit 7**	**Bit 6**	**Bit 5**	**Bit 4**	**Bit 3**	**Bit 2**	**Bit 1**
SMR	0	0	1	O/$\overline{E}$	1	0	CKS1
BRR	BRR7	BRR6	BRR5	BRR4	BRR3	BRR2	BRR1
SCR	TIE	RIE	TE	RE	0	0	0
TDR	TDR7	TDR6	TDR5	TDR4	TDR3	TDR2	TDR1
SSR	TDRE	RDRF	ORER	ERS	PER	TEND	0
RDR	RDR7	RDR6	RDR5	RDR4	RDR3	RDR2	RDR1
SCMR	—	—	—	—	SDIR	SINV	—

Note: —: Unused bit.

SMR Setting: The O/$\overline{E}$ bit is cleared to 0 if the IC card is of the direct convention typ
1 if of the inverse convention type.

Bits CKS1 and CKS0 select the clock source of the on-chip baud rate generator. See s
12.3.5, Clock.

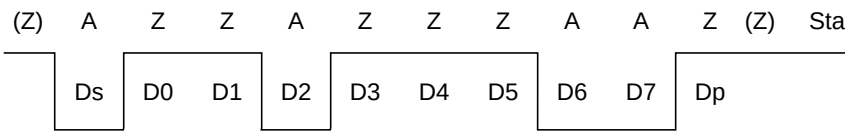
BRR Setting: BRR is used to set the bit rate. See section 12.3.5, Clock, for the metho
calculating the value to be set.

SCR Setting: The function of the TIE, RIE, TE, and RE bits is the same as for the no
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, o
clock is to be output.

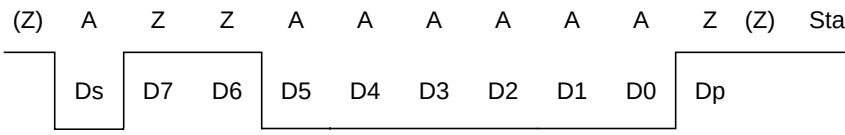
types of IC card (direct convention and inverse convention).

- Direct convention (SDIR = SINV = O/$\overline{E}$ = 0)



With the direct convention type, the logic 1 level corresponds to state Z and the log
state A, and transfer is performed in LSB-first order. The start character data above
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 lo
to state Z, and transfer is performed in MSB-first order. The start character data abo
The parity bit is 0, corresponding to state Z, since even parity is stipulated for the S
With the H8/3039 Group, inversion specified by the SINV bit applies only to the da
to D0. For parity bit inversion, the O/$\overline{E}$ bit in SMR is set to odd parity mode (the sa
to both transmission and reception).

$$B = \frac{\phi}{1488 \times 2^{2n-1} \times (N + 1)} \times 10^6$$

Where: N = Value set in BRR ($0 \leq N \leq 255$)
B = Bit rate (bit/s)
ϕ = Operating frequency* (MHz)
n = See table 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 di
frequency to calculate the bit rate. The equation above applies directly to 1
division.

Table 12.5 Examples of Bit Rate B (bit/s) for Various BRR Settings (When n = (

	ϕ (MHz)					
N	**7.1424**	**10.00**	**10.7136**	**13.00**	**14.2848**	**16.00**
0	9600.0	13440.9	14400.0	17473.1	19200.0	21505.4
1	4800.0	6720.4	7200.0	8736.6	9600.0	10752.7
2	3200.0	4480.3	4800.0	5824.4	6400.0	7168.5

Note: Bit rates are rounded off to one decimal place.

	7.1424		10.00		10.7136		13.00		14.2848		16.00	
bit/s	N	Error	N	Error	N	Error	N	Error	N	Error	N	Error
9600	0	0.00	1	30	1	25	1	8.99	1	0.00	1	12.01

Table 12.7 Maximum Bit Rate at Various Frequencies (Smart Card Interface M

ϕ (MHz)	Maximum Bit Rate (bit/s)	N	n
7.1424	9600	0	0
10.00	13441	0	0
10.7136	14400	0	0
13.00	17473	0	0
14.2848	19200	0	0
16.00	21505	0	0
18.00	24194	0	0

The bit rate error is given by the following formula:

$$\text{Error}\ (\%) = \left(\frac{\phi}{1488 \times 2^{2n-1} \times B \times (N + 1)} \times 10^{6} - 1\right) \times 100$$

[2] Clear the error flags ERS, PER, and ORER in SSR to 0.

[3] Set the O/$\overline{E}$ bit and CKS1 and CKS0 bits in SMR. Clear the C/$\overline{A}$, CHR, and MP bi
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 por
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
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 n
bit and RE bit at the same time, except for self-diagnosis.

[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 t

[4] Write the transmit data to TDR, clear the TDRE flag to 0, and perform the transmit
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 inter
requests are enabled, a transmit data empty interrupt (TXI) request will be generated. If
occurs in transmission and the ERS flag is set to 1 while the RIE bit is set to 1 and inter
requests are enabled, a transfer error interrupt (ERI) request will be generated.

For details, see the following Interrupt Operations.

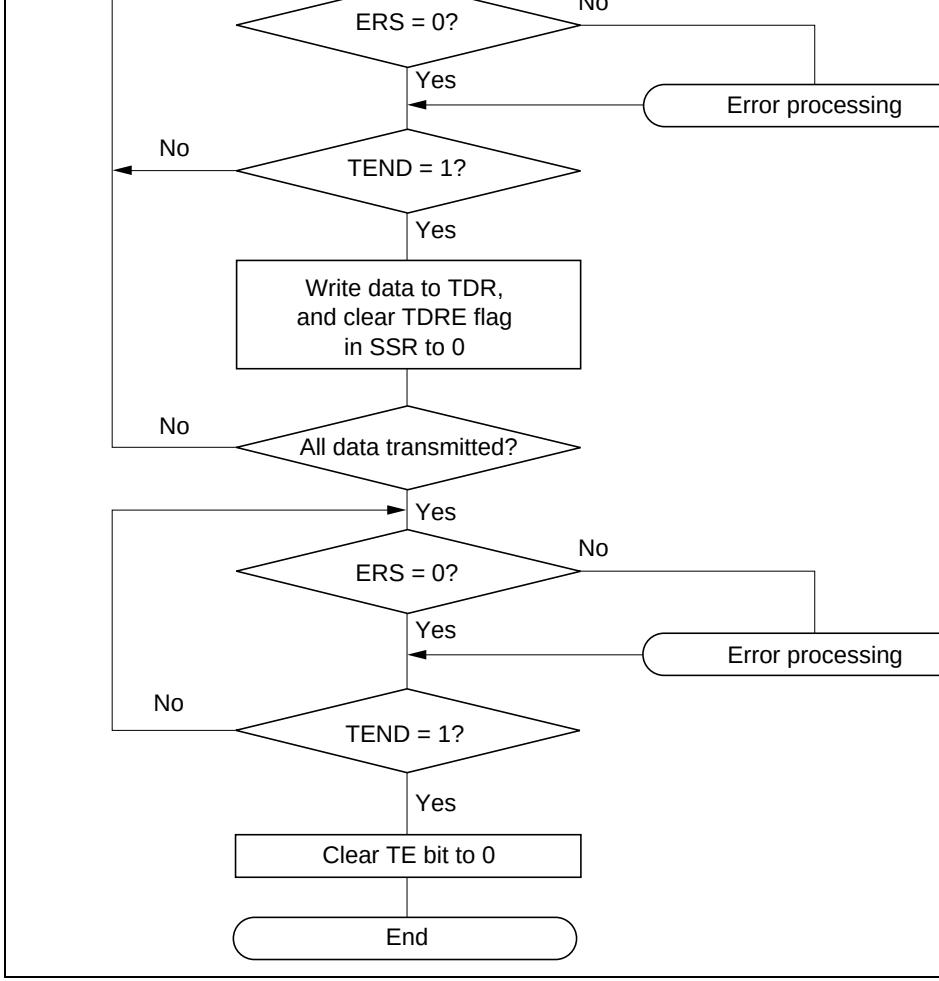


Figure 12.4 Example of Transmission Processing Flow

In case of normal transmission: TEND flag is set
In case of transmit error: ERS flag is set
Steps (2) and (3) above are repeated until the TEN

Note: When the ERS flag is set, it should be cleared until transfer of the last bit (D7 in L transmission, D0 in MSB-first transmission) of the next transfer data has been co

Figure 12.5 Relation Between Transmit Operation and Internal Registe

Serial Data Reception

Data reception in Smart Card mode uses the same processing procedure as for the norm Figure 12.6 shows an example of the transmission processing flow.

[1] Perform Smart Card interface mode initialization as described above in Initializatio

[2] Check that the ORER flag and PER flag in SSR are cleared to 0. If either flag is set, the appropriate receive error processing, then clear both the ORER and the PER flag

[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.

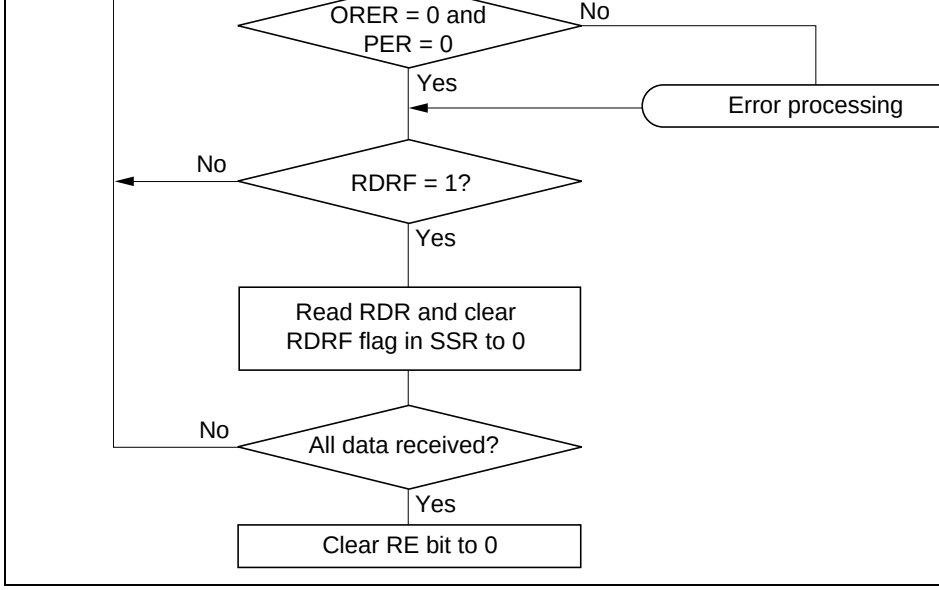


Figure 12.6 Example of Reception Processing Flow

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 interru
are enabled, a receive data full interrupt (RXI) request will be generated. If an error oc
reception and either the ORER flag or the PER flag is set to 1, a transfer error interrup
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 sti
transferred to RDR, and therefore this data can be read.

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 inte
requests, transfer error interrupt (ERI) requests, and receive data full interrupt (RXI) re
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
The relationship between the operating states and interrupt sources is shown in table 12

Table 12.8 Smart Card Mode Operating States and Interrupt Sources

Operating State		Flag	Mask Bit	Interrupt So
Transmit Mode	Normal operation	TEND	TIE	TXI
	Error	ERS	RIE	ERI
Receive Mode	Normal operation	RDRF	RIE	RXI
	Error	PER, ORER	RIE	ERI

In reception, the SCI samples the falling edge of the start bit using the basic clock, and
internal synchronization. Receive data is latched internally at the rising edge of the 18
the basic clock. This is illustrated in figure 12.7.

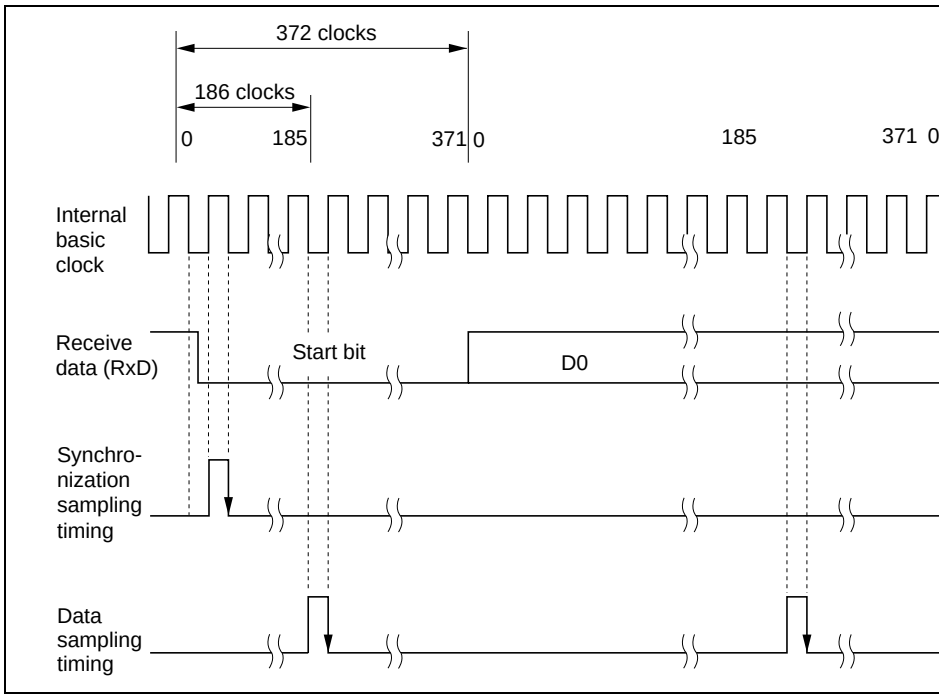


Figure 12.7 Receive Data Sampling Timing in Smart Card Mode

F: Absolute value of clock frequency deviation

Assuming values of F = 0 and D = 0.5 in the above formula, the reception margin form follows.

When D = 0.5 and F = 0,

$$M = (0.5 - 1/2 \times 372) \times 100\%$$
$$= 49.866\%$$

automatically set to 1. If the RIE bit in SCR is enabled at this time, an ERI interrup
generated. The PER bit in SSR should be kept cleared to 0 until the next parity bit

[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 r

[4] If no error is found when the received parity bit is checked, the receive operation i
have been completed normally, and the RDRF flag in SSR is automatically set to 1
bit in SCR is enabled at this time, an RXI interrupt request is generated.

[5] When a normal frame is received, the pin retains the high-impedance state at the ti
error signal transmission.

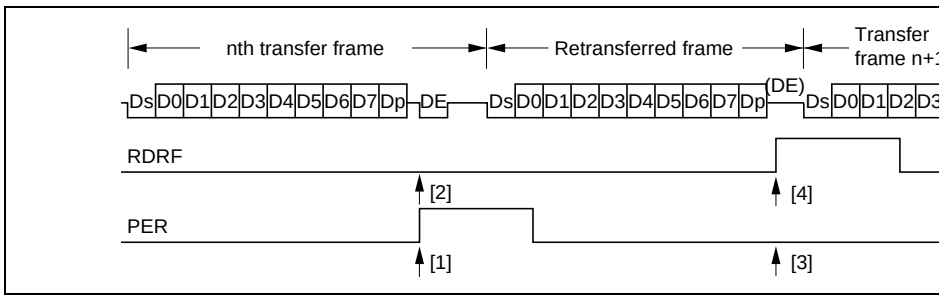


Figure 12.8 Retransfer Operation in SCI Receive Mode

is received.

[8] If an error signal is not sent back from the receiving end, the ERS bit in SSR is not s

[9] If an error signal is not sent back from the receiving end, transmission of one frame a retransfer, is judged to have been completed, and the TEND bit in SSR is set to 1. bit in SCR is enabled at this time, a TXI interrupt request is generated.

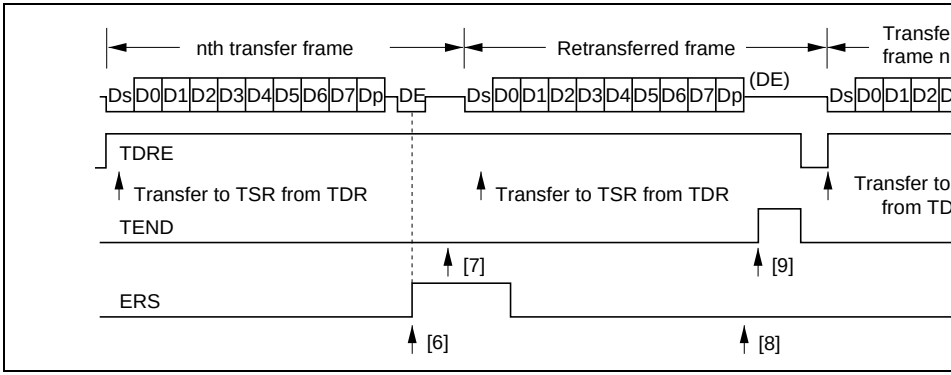


Figure 12.9 Retransfer Operation in SCI Transmit Mode

When the A/D converter is not used, it can be halted independently to conserve power 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 ref voltage at the AV_{CC} pin.
- High-speed conversion

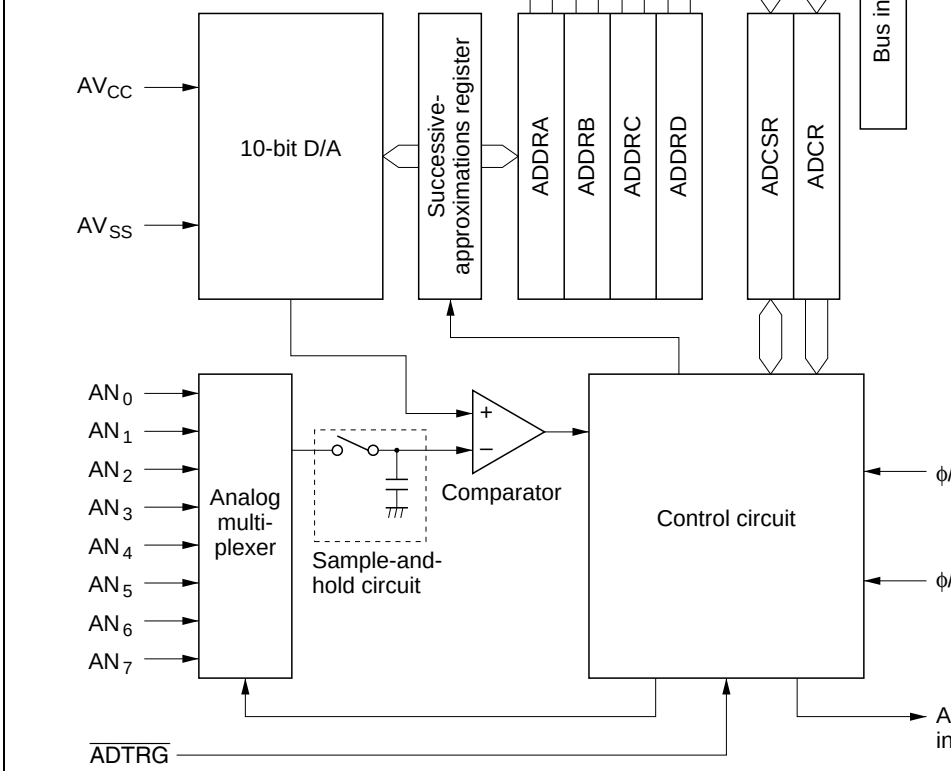
  Conversion time: minimum 7.4 µs per channel (with 18 MHz system clock)
- Two conversion modes

  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 channels.
- Sample-and-hold function
- A/D conversion can be externally triggered
- A/D interrupt requested at end of conversion

  At the end of A/D conversion, an A/D end interrupt (ADI) can be requested.



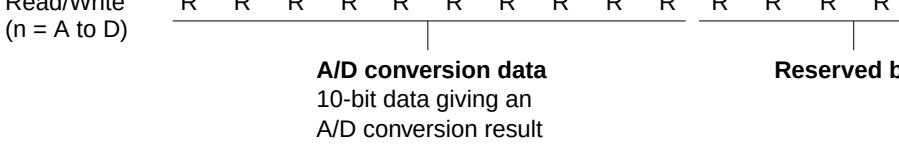
Legend:
ADCR: A/D control register
ADCSR: A/D control/status register
ADDRA: A/D data register A
ADDRB: A/D data register B
ADDRC: A/D data register C
ADDRD: A/D data register D

Figure 13.1 A/D Converter Block Diagram

Pin Name	ation	I/O	Function
Analog power supply pin	AV_{CC}	Input	Analog power supply and referer
Analog ground pin	AV_{SS}	Input	Analog ground and reference vol
Analog input pin 0	AN_0	Input	Group 0 analog inputs
Analog input pin 1	AN_1	Input	
Analog input pin 2	AN_2	Input	
Analog input pin 3	AN_3	Input	
Analog input pin 4	AN_4	Input	Group 1 analog inputs
Analog input pin 5	AN_5	Input	
Analog input pin 6	AN_6	Input	
Analog input pin 7	AN_7	Input	
A/D external trigger input pin	$\overline{ADTRG}$	Input	External trigger input for starting conversion

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.



The four A/D data registers (ADDRA to ADDRD) are 16-bit read-only registers that s
results of A/D conversion.

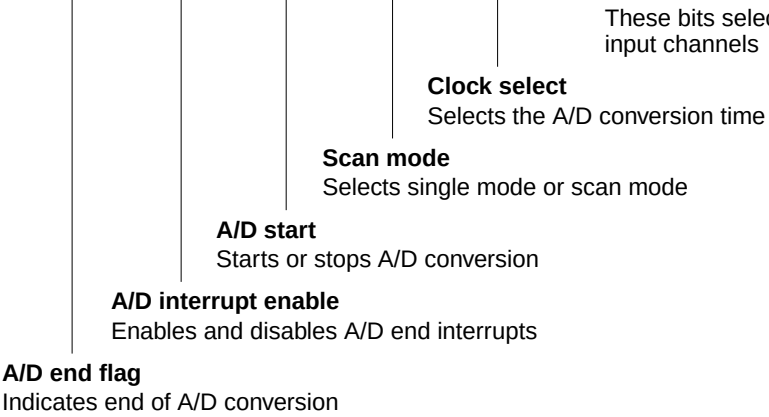
An A/D conversion produces 10-bit data, which is transferred for storage into the A/D
register corresponding to the selected channel. The upper 8 bits of the result are stored
byte of the A/D data register. The lower 2 bits are stored in the lower byte. Bits 5 to 0
data register are reserved bits that always read 0. Table 13.3 indicates the pairings of a
channels and A/D data registers.

The CPU can always read the A/D data registers. The upper byte can be read directly,
lower byte is read through a temporary register (TEMP). For details see section 13.3,
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



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

ADCSR is an 8-bit readable/writable register that selects the mode and controls the A/D
ADCSR is initialized to H'00 by a reset and in standby mode.

Bit 7—A/D End Flag (ADF): Indicates the end of A/D conversion.

Bit 7 ADF	Description	
0	[Clearing condition]	(Ini
	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 5—A/D Start (ADST): Starts or stops A/D conversion. The ADST bit remains se
A/D conversion. It can also be set to 1 by external trigger input at the $\overline{\text{ADTRG}}$ pin.

Bit 5 ADST	Description	
0	A/D conversion is stopped	(I
1	Single mode: A/D conversion starts; ADST is automatically cleared to 0 w conversion ends.	
	Scan mode: A/D conversion starts and continues, cycling among the sel channels, until ADST is cleared to 0 by software, by a reset transition to standby mode.	

Bit 4—Scan Mode (SCAN): Selects single mode or scan mode. For further informati
operation in these modes, see section 13.4, Operation. Clear the ADST bit to 0 before
the conversion mode.

Bit 4 SCAN	Description	
0	Single mode	(I
1	Scan mode	

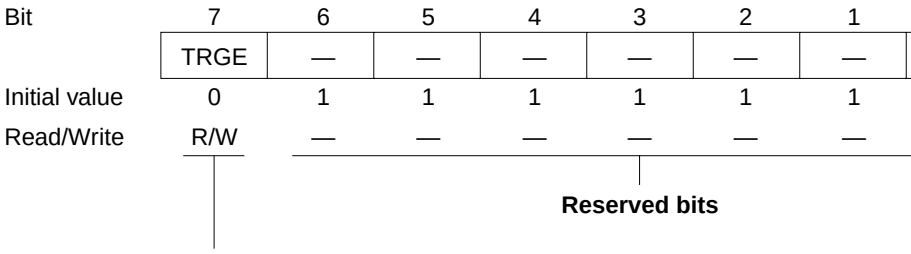
Bit 3—Clock Select (CKS): Selects the A/D conversion time. Clear the ADST bit to
switching the conversion time.

Bit 3 CKS	Description	
0	Conversion time = 266 states (maximum)	(I
1	Conversion time = 134 states (maximum)	

Bits 2 to 0—Channel Select 2 to 0 (CH2 to CH0): These bits and the SCAN bit sele
input channels. Clear the ADST bit to 0 before changing the channel selection.

1	0	0	AN_4	AN_4
		1	AN_5	AN_4, AN_5
	1	0	AN_6	AN_4 to AN_6
		1	AN_7	AN_4 to AN_7

13.2.3 A/D Control Register (ADCR)



ADCR is an 8-bit readable/writable register that enables or disables external triggering conversion. ADCR is initialized to H'7F by a reset and in standby mode.

Bit 7—Trigger Enable (TRGE): Enables or disables external triggering of A/D conve

Bit 7 TRGE	Description	
0	A/D conversion cannot be externally triggered	(In
1	A/D conversion starts at the falling edge of the external trigger signal ($\overline{\text{ADTI}}$	

Bits 6 to 0—Reserved: These bits cannot be modified and are always read as 1.

When reading an A/D data register, always read the upper byte before the lower byte.
to read only the upper byte, but if only the lower byte is read, incorrect data may be ob

Figure 13.2 shows the data flow for access to an A/D data register.

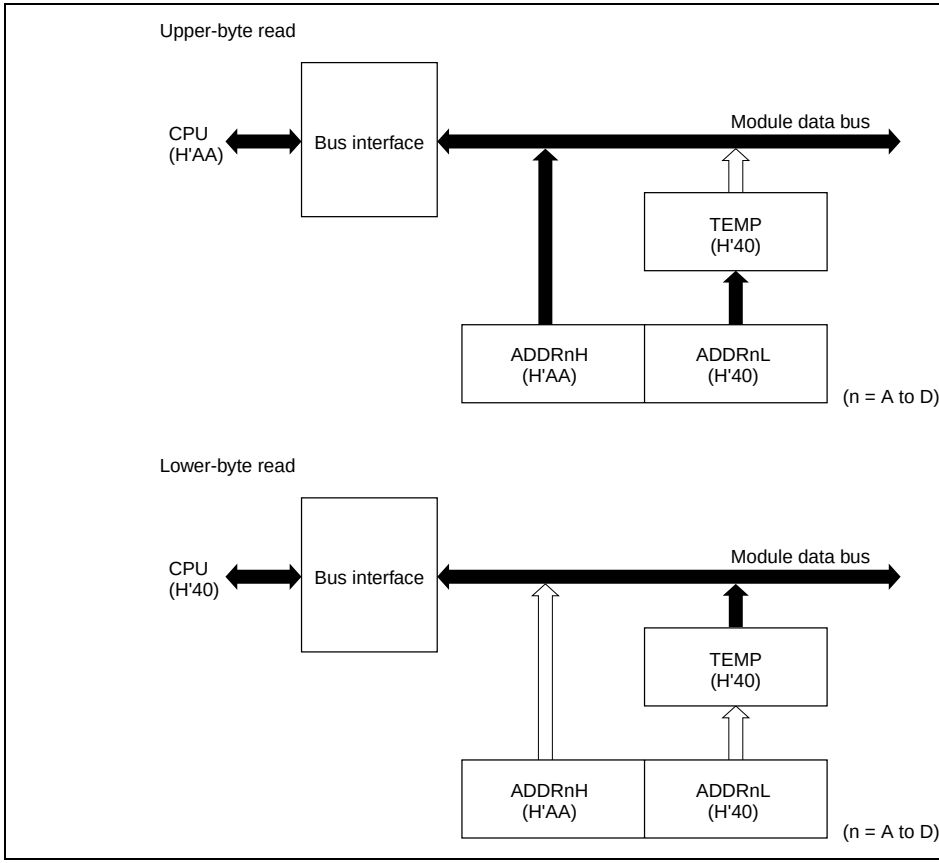


Figure 13.2 A/D Data Register Access Operation (Reading H'AA40)

conversion starts when the ADST bit is set to 1 by software, or by external trigger inpu
ADST bit remains set to 1 during A/D conversion and is automatically cleared to 0 whe
conversion ends.

When conversion ends the ADF bit is set to 1. If the ADIE bit is also set to 1, an ADI i
requested at this time. To clear the ADF flag to 0, first read ADCSR, then write 0 in AI

When the mode or analog input channel must be switched during analog conversion, to
incorrect operation, first clear the ADST bit to 0 in ADCSR to halt A/D conversion. Af
the necessary changes, set the ADST bit to 1 to start A/D conversion again. The ADST
set at the same time as the mode or channel is changed.

Typical operations when channel 1 (AN_1) is selected in single mode are described next.

Figure 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
   1), the A/D interrupt is enabled (ADIE = 1), and A/D conversion is started (ADST =
2. When A/D conversion is completed, the result is transferred into ADDRB. At the sa
   the ADF flag is set to 1, the ADST bit is cleared to 0, and the A/D converter becom
3. Since ADF = 1 and ADIE = 1, an ADI interrupt is requested.
4. The A/D interrupt handling routine starts.
5. The routine reads ADCSR, then writes 0 in the ADF flag.
6. The routine reads and processes the conversion result (ADDRB).
7. Execution of the A/D interrupt handling routine ends. After that, if the ADST bit is
   A/D conversion starts again and steps 2 to 7 are repeated.

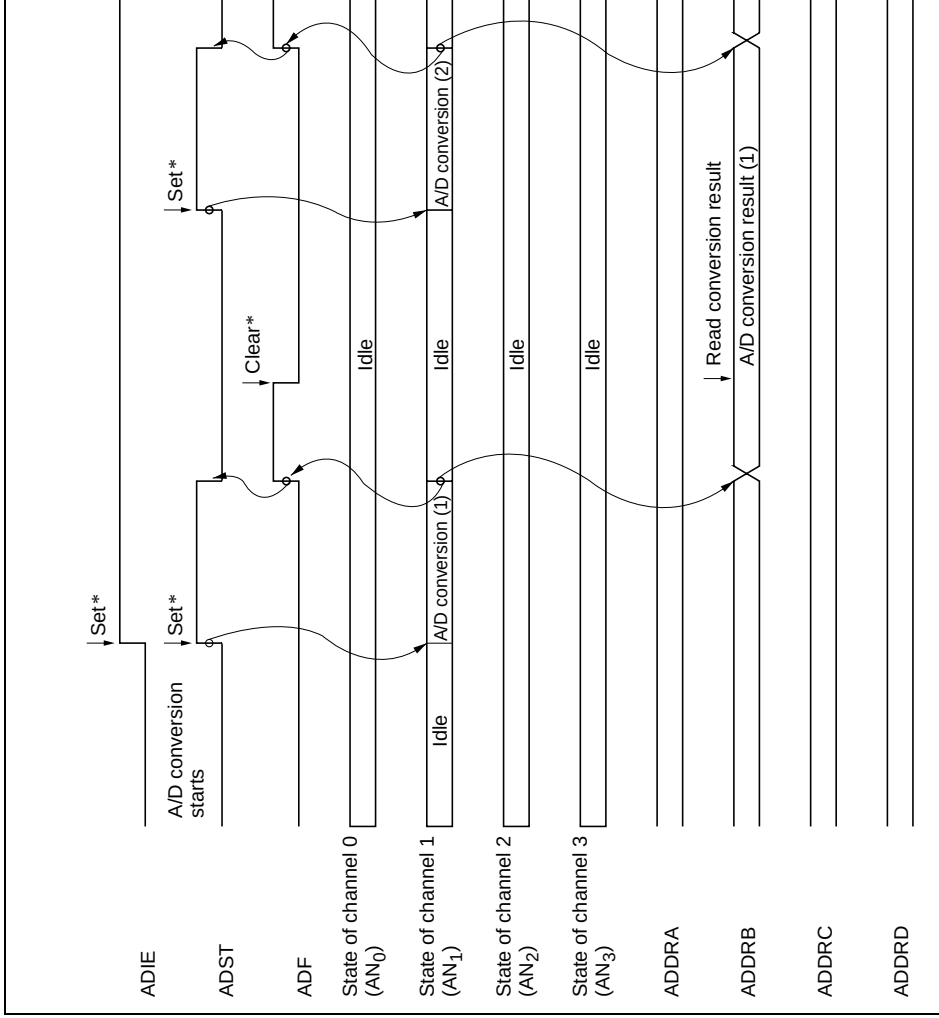


Figure 13.3 Example of A/D Converter Operation (Single Mode, Channel 1 Selected)

corresponding to the channels.

When the mode or analog input channel selection must be changed during analog conv prevent incorrect operation, first clear the ADST bit to 0 in ADCSR to halt A/D conver making the necessary changes, set the ADST bit to 1. A/D conversion will start again f first channel in the group. The ADST bit can be set at the same time as the mode or cha selection is changed.

Typical operations when three channels in group 0 (AN_0 to AN_2) are selected in scan m described next. Figure 13.4 shows a timing diagram for this example.

1. Scan mode is selected (SCAN = 1), scan group 0 is selected (CH2 = 0), analog inpu AN_0 to AN_2 are selected (CH1 = 1, CH0 = 0), and A/D conversion is started (ADST
2. When A/D conversion of the first channel (AN_0) is completed, the result is transfer ADDRA. Next, conversion of the second channel (AN_1) starts automatically.
3. Conversion proceeds in the same way through the third channel (AN_2).
4. When conversion of all selected channels (AN_0 to AN_2) is completed, the ADF flag and conversion of the first channel (AN_0) starts again. If the ADIE bit is set to 1, an interrupt is requested at this time.
5. Steps 2 to 4 are repeated as long as the ADST bit remains set to 1. When the ADST cleared to 0, A/D conversion stops. After that, if the ADST bit is set to 1, A/D conv starts again from the first channel (AN_0).

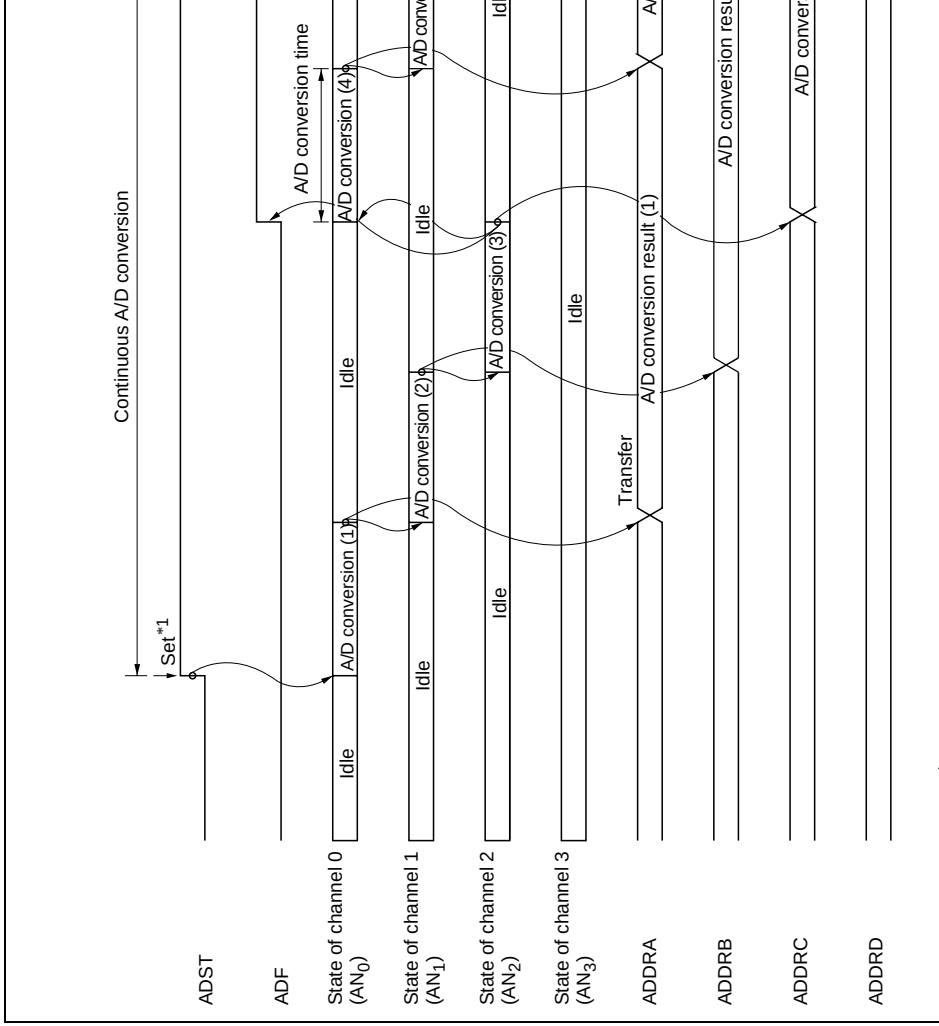


Figure 13.4 Example of A/D Converter Operation (Scan Mode, Channels AN_0 to AN_2 Selected)

In scan mode, the values given in table 13.4 apply to the first conversion. In the second subsequent conversions the conversion time is fixed at 256 states when CKS = 0 or 128 when CKS = 1.

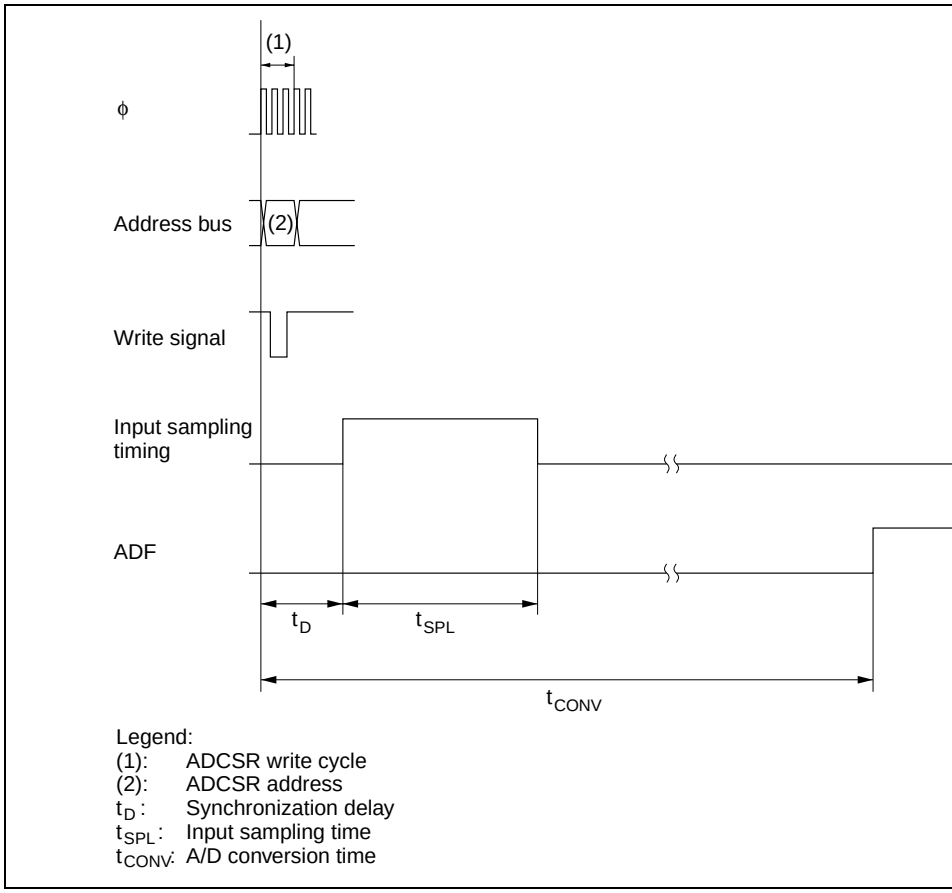


Figure 13.5 A/D Conversion Timing

13.4.4 External Trigger Input Timing

A/D conversion can be externally triggered. When the TRGE bit is set to 1 in ADCR, trigger input is enabled at the $\overline{\text{ADTRG}}$ pin. A high-to-low transition at the $\overline{\text{ADTRG}}$ pi ADST bit to 1 in ADCSR, starting A/D conversion. Other operations, in both single a modes, are the same as if the ADST bit had been set to 1 by software. Figure 13.6 sho timing.

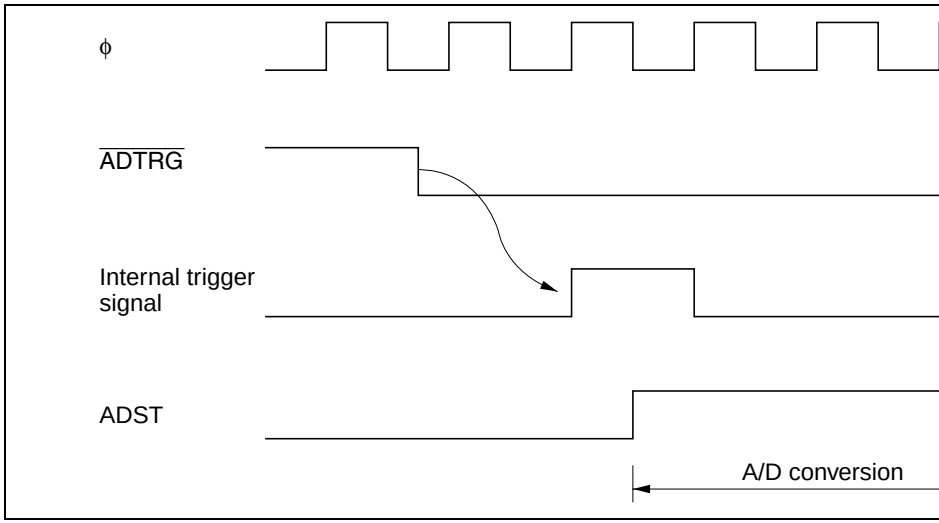


Figure 13.6 External Trigger Input Timing

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 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 con 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 adversel

Notes on Board Design

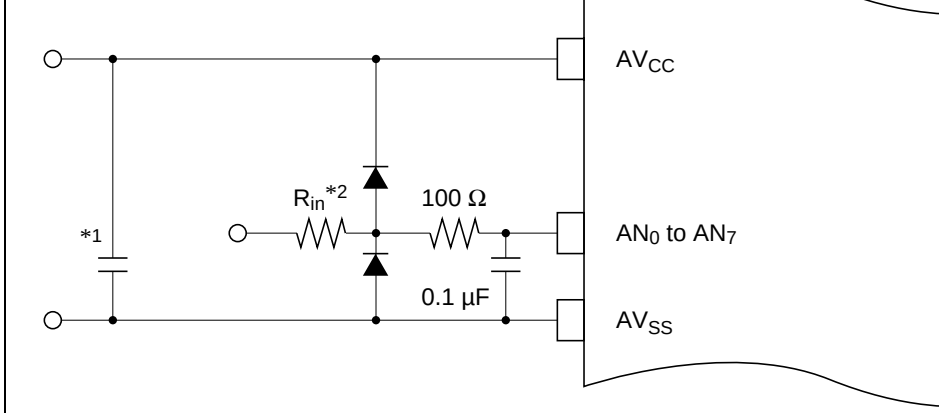
In board design, digital circuitry and analog circuitry should be as mutually isolated as and layout in which digital circuit signal lines and analog circuit signal lines cross or ar proximity should be avoided as far as possible. Failure to do so may result in incorrect of the analog circuitry due to inductance, adversely affecting A/D conversion values.

Also, digital circuitry must be isolated from the analog input signals (AN_0 to AN_7), and power supply and reference voltage (AV_{CC}) by the analog ground (AV_{SS}). Also, the anal (AV_{SS}) should be connected at one point to a stable digital ground (V_{SS}) on the board.

Notes on Noise Countermeasures

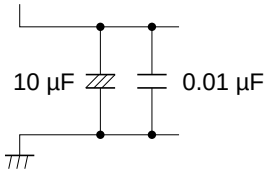
A protection circuit connected to prevent damage due to an abnormal voltage such as a surge at the analog input pins (AN_0 to AN_7) and analog power supply (AV_{CC}) should be between AV_{CC} and AV_{SS} as shown in figure 13.7.

Also, the bypass capacitors connected to AV_{CC} and the filter capacitor connected to AN must be connected to AV_{SS}.



Notes: Values are reference values.

1.



2. R_{in}: Input impedance

Figure 13.7 Example of Analog Input Protection Circuit

Table 13.5 Analog Pin Specifications

Item	Min	Max	Un
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

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 char
  when the digital output changes from the minimum voltage value B'0000000000 (H
  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 char
  when the digital output changes from B'1111111110 (H'3FE) to B'1111111111 (H'3
  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 v
  the full-scale voltage. Does not include the offset error, full-scale error, or quantizat

- Absolute precision

  The deviation between the digital value and the analog input value. Includes the off
  full-scale error, quantization error, and nonlinearity error.

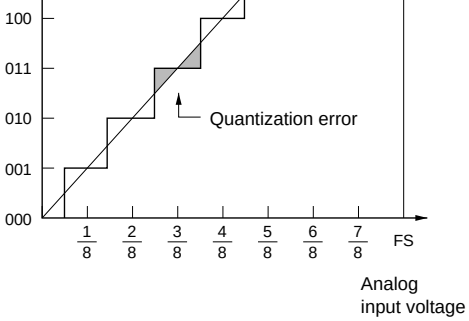


Figure 13.9 A/D Conversion Precision Definitions (1)

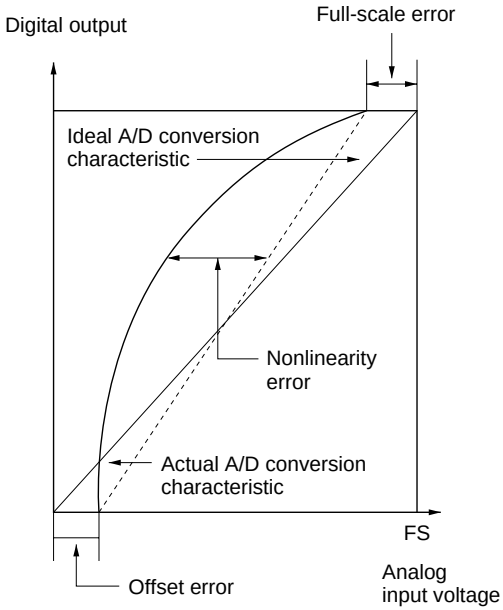


Figure 13.10 A/D Conversion Precision Definitions (2)

will essentially comprise only the internal input resistance of 10 kΩ, and the signal sou
impedance is ignored.

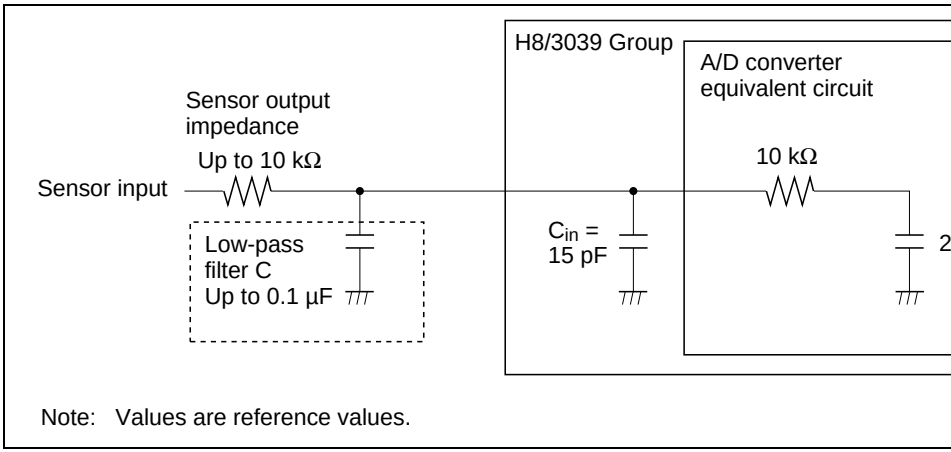
However, since a low-pass filter effect is obtained in this case, it may not be possible to
analog signal with a large differential coefficient (e.g., voltage regulation 5 mV/µs or g

When converting a high-speed analog signal and when performing conversion in the sc
low-impedance buffer should be inserted.

Influences on Absolute Precision

Adding capacitance results in coupling with GND, and therefore noise in GND may ad
affect absolute precision. Be sure to make the connection to an electrically stable GND
AV_{SS}.

Care is also required to insure that filter circuits do not communicate with digital signal
mounting board, thus acting as antennas.



Note: Values are reference values.

Figure 13.11 Example of Analog Input Circuit

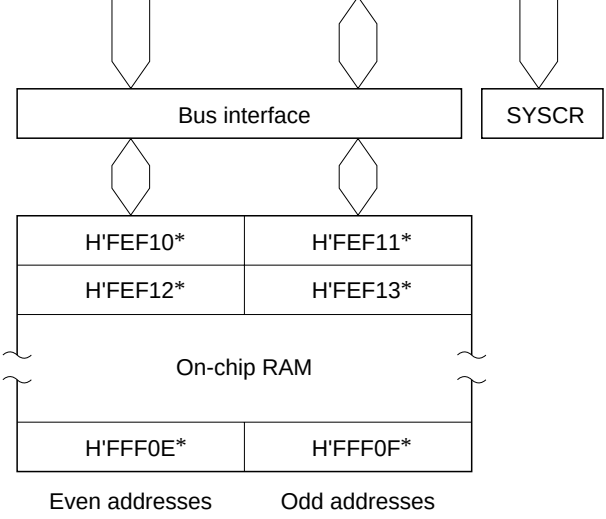
transfer.

The RAM enable bit (RAME) in the system control register (SYSCR) can enable or d
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)	H8/303 (512 b
Modes 1, 5, 7	H'FEF10 to H'FFF0F	H'FF710 to H'FFF0F	H'FFB10 to H'FFF0F	H'FFD H'FFF
Mode 3	H'FFEF10 to H'FFFF0F	H'FFF710 to H'FFFF0F	H'FFFB10 to H'FFFF0F	H'FFF H'FFF
Mode 6	H'F710 to H'FF0F	H'F710 to H'FF0F	H'FB10 to H'FF0F	H'FD1 H'FF0



Legend:
SYSCR: System control register

Note: * Lower 20 bits of the address

Figure 14.1 RAM Block Diagram (H8/3039 in Modes 1, 5 and 7)

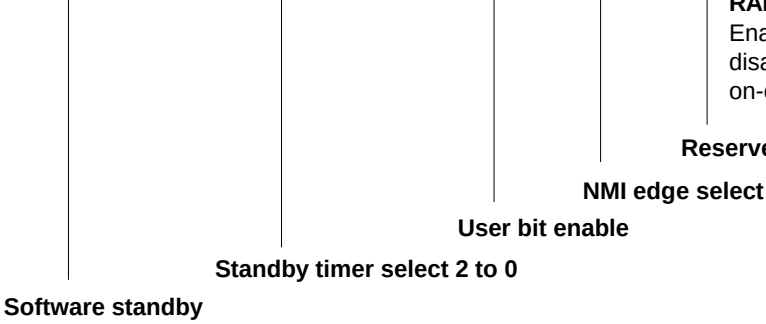
14.1.2 Register Configuration

The on-chip RAM is controlled by the system control register (SYSCR). Table 14.2 giv
address and initial value of SYSCR.

Table 14.2 RAM Control Register

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

Note: * Lower 16 bits of the address



One function of SYSCR is to enable or disable access to the on-chip RAM. The on-ch
enabled or disabled by the RAME bit in SYSCR. For details about the other bits, see s
System Control Register (SYSCR).

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

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

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 da
beginning from an even address can be accessed in two states using the 16-bit data bus.

transfer.

The mode pins (MD_2 to MD_0) can be set to enable or disable the on-chip ROM. See ta

The on-chip flash memory product (H8/3039F-ZTAT) can be erased and programmed 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-Ch**
Mode 1 (1-Mbyte expanded mode with on-chip ROM disabled)	0	0	1	Disabl addres
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	Enable
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 mo

— 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
  block to be erased can be specified by setting the corresponding bit. There are block
  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 prog
  equivalent to 300 µs (typ.) per byte, and the erase time is 100 ms (typ.) per block.
- Reprogramming capability

  The flash memory can be reprogrammed up to 100 times.
- On-board programming modes

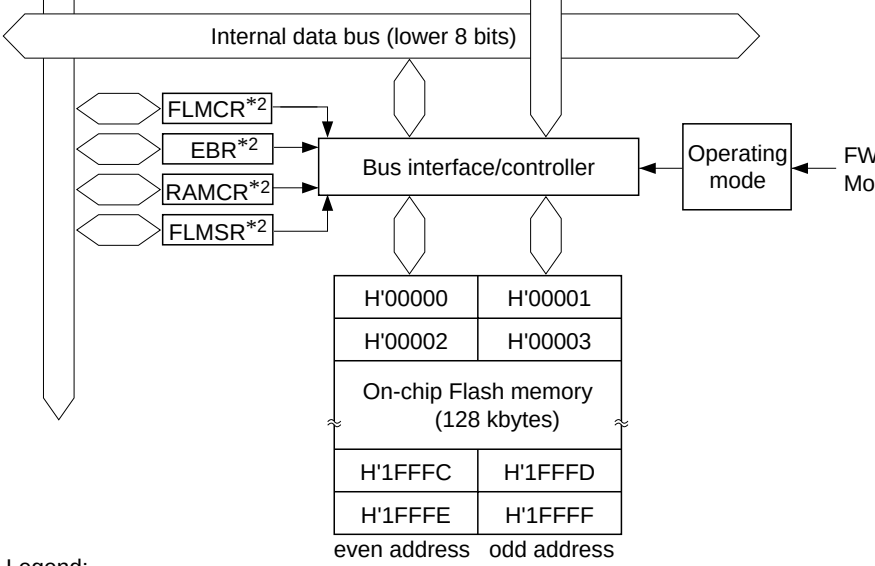
  There are two modes in which flash memory can be programmed/erased/verified on
  — Boot mode
  — User program mode
- Automatic bit rate adjustment

  With data transfer in boot mode, the this LSI's bit rate can be automatically adjusted
  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 memo
  in real time.
- PROM mode

  Flash memory can be programmed/erased in PROM mode, using a PROM program
  well as in on-board programming mode.
- Protect modes

  There are three protect modes, hardware, software, and error protect, which allow p
  status to be designated for flash memory program/erase/verify operations.



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 $\overline{\text{RESO}}$ p
mask ROM versions.
2. The registers that control the flash memory versions (FLMCR, EBR, RAMCI
FLMSR) are used in the flash memory versions only. They are not provided
mask ROM versions. Reading the corresponding addresses in a mask ROM
will always return 1s, and writes to these addresses are disabled.

Figure 15.1 Block Diagram of Flash Memory

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 $\overline{\text{RESO}}$ pin.

15.2.4 Register Configuration

The registers used to control the on-chip flash memory when enabled are shown in tabl

Table 15.3 Flash Memory Registers

Register Name	Abbreviation	R/W	Initial Value	Addr
Flash memory control register	FLMCR	R/W	H'00*[2]	H'FF4
Erase block register	EBR	R/W	H'00	H'FF4
RAM control register	RAMCR	R/W	H'F1	H'FF4
Flash memory status register	FLMSR	R	H'7F	H'FF4

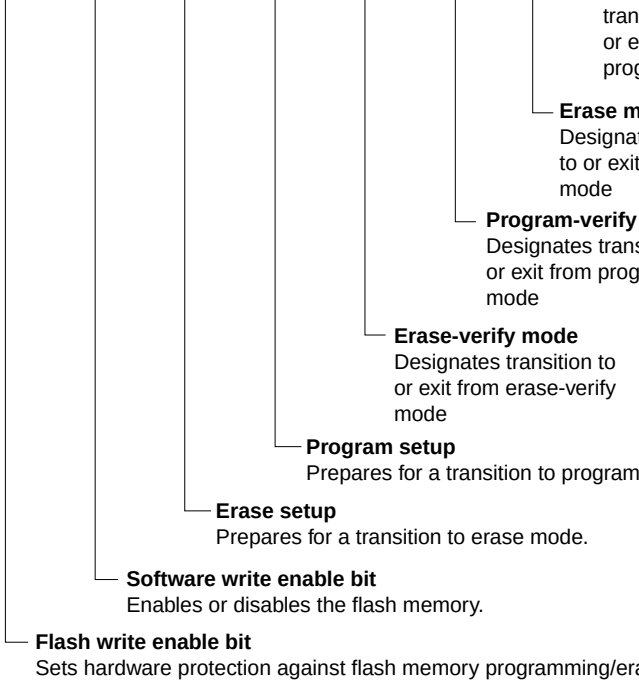
Notes: 1. Lower 16 bits of the address.

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 cor addresses in a mask ROM version will always return 1s, and writes to these addresses a

the E bit. FLMCR is initialized by a reset, and in hardware standby mode and softwar
mode. Its initial value is H'80 when a high level is input to the FWE pin, and H'00 wh
level is input. In mode 6 the FWE pin must be fixed low, as flash memory on-board p
is not supported. Therefore, bits in this register cannot be set to 1 in mode 6. When on
memory is disabled, a read will return H'00, and writes are invalid. When setting bits
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
1; writes to the E bit only when FWE = 1, SWE = 1, and ESU = 1; and writes to the P
when FWE = 1, SWE = 1, and PSU = 1.



Bit 7—Flash Write Enable Bit (FWE): Sets hardware protection against flash memor programming/erasing. When using this bit, refer to section 15.9, Notes on Flash Memo Programming/Erasing.

Bit 7 FWE	Description
0	When a low level is input to the FWE pin (hardware-protected state)
1	When a high level is input to the FWE pin

	[Setting condition]
	When FWE = 1

Bit 5—Erase Setup Bit (ESU)*[1]: Prepares for a transition to erase mode. Do not set t
PSU, EV, PV, E, or P bit at the same time.

Bit 5 ESU	Description	
0	Erase setup cleared	(I
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 n
SWE, ESU, EV, PV, E, or P bit at the same time.

Bit 4 PSU	Description	
0	Program setup cleared	(I
1	Program setup [Setting condition] When FWE = 1, and SWE = 1	

	When FWE = 1, and SWE = 1

Bit 2—Program-Verify (PV)*[1]: Selects program-verify mode transition or clearing. D
the SWE, ESU, PSU, EV, E, or P bit at the same time.

Bit 2 PV	Description	
0	Program-verify mode cleared	(Ini
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, E
EV, PV, or P bit at the same time.

Bit 1 E	Description	
0	Erase mode cleared	(Ini
1	Transition to erase mode [Setting condition] When FWE = 1, SWE = 1, and ESU = 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, E
3. Set the P and E bits according to the program and erase algorithms shown
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 in
H'00 by a reset, in hardware standby mode, or software standby mode, when a high le
input to the FWE terminal, or when the FLMCR SWE bit is 0 when a high level is app
FWE terminal. When a bit is set in EBR, the corresponding block can be erased. Othe
erase - protected. The blocks are erased block by block. Therefore, set only one bit in
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
configuration is shown in table 15.4. To erase all the blocks, erase each block sequent

This LSI does not support the on-board programming mode in mode 6, so bits in this
cannot be set to 1 in mode 6.

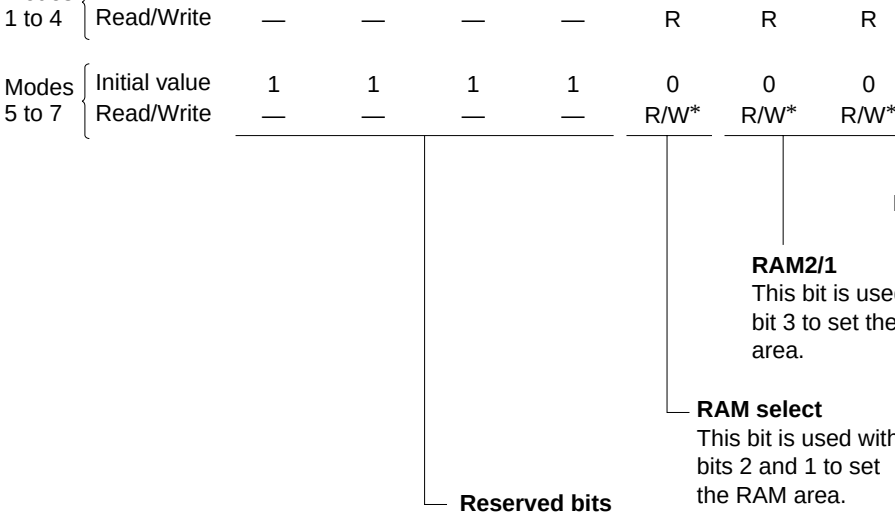
Bits 7 to 0—Block 7 to 0 (EB7 to EB0): These bits select blocks (EB7 to EB0) to be e

Bits 7 to 0 EB7 to EB0	Description	
0	Block EB7 to EB0 is not selected.	(Ini
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



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 (se
The initial setting for this bit is 0 in modes 5, 6, and 7 (internal flash memory enabled
programming is enabled.* In modes other than 5 to 7, 0 is always read and writing is d
It is initialized by a reset and in hardware standby mode. It is not initialized in softwar
mode.
When bit 3 is set, all flash-memory blocks are protected from programming and erasi

Bits 2 to 1—RAM2 to RAM1: These bits are used with bit 3 to reassign an area to R
table 15.5). The initial setting for this bit is 0 in modes 5, 6, and 7 (internal flash mem
and programming is enabled.* In modes other than 5 to 7, 0 is always read and writing
They are initialized by a reset and in hardware standby mode. They are not initialized
standby mode.

RAM Area	Bit 3 RAMS	Bit 2 RAM2	Bit 1 RAM1	RAM Emulation St
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	

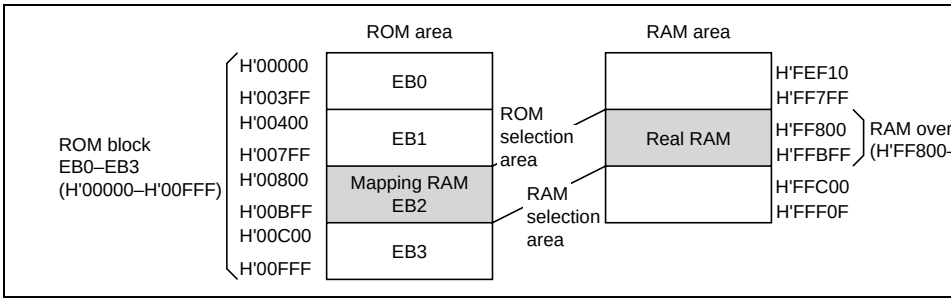
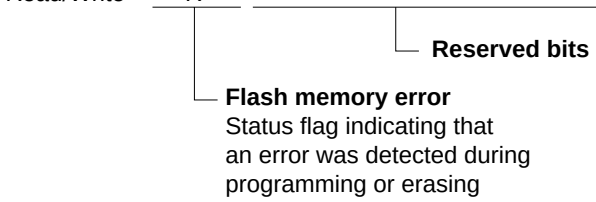


Figure 15.2 Example of Overlap ROM Area and RAM Area



Bit 7—Flash Memory Error (FLER): Indicates that an error occurred while flash m
being programmed or erased. When bit 7 is set, flash memory is placed in an error-pro

Bit 7 FLER	Description
0	Flash memory program/erase protection (error protection*[1]) is disabled [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 (includin instruction fetch, but not including reading of a RAM area overlapped c memory). 2. A hardware exception-handling sequence (other than a reset, invalid in trap instruction, or zero-divide exception) was executed just before pro erasing.*[3] 3. The SLEEP instruction (including software standby mode) was execut programming or erasing.

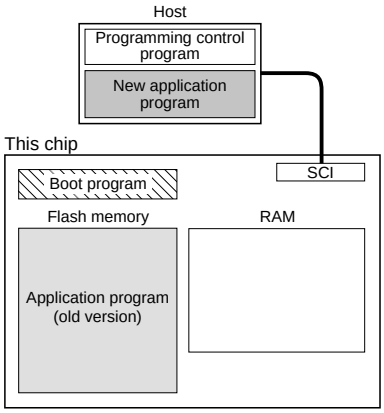
Notes: 1. For details, see section 15.6.3, Error Protection.
2. The read data has undetermined values.
3. Before stack and vector read by exception handling.

Bits 6 to 0—Reserved: These bits cannot be modified and are always read as 1.

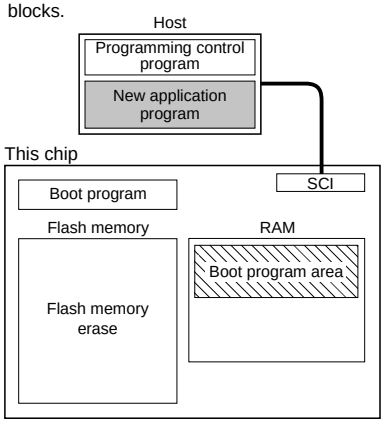
Table 15.6 Setting On-Board Programming Modes

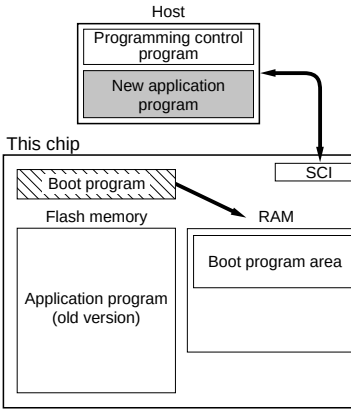
Mode		FWE	MD_2	MD_1	MD_0
Boot mode	mode 5	1*[1]	0*[2]	0	1
	mode 7		0*[2]	1	1
User program mode	mode 5		1	0	1
	mode 7		1	1	1

Notes: 1. For the High level input timing, see items (6) and (7) of Notes on Using the E
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 of modes 5 and 7 in the same way as in the normal mode.



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 transfe
from the host to RAM by SCI communicat
executed, and the new application progra
host is written into the flash memory.

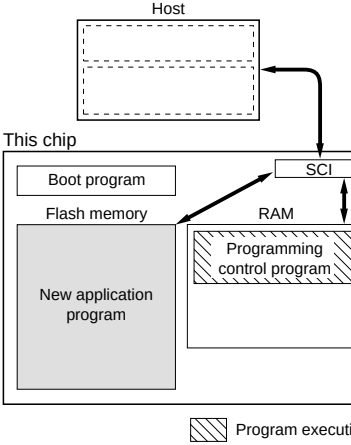
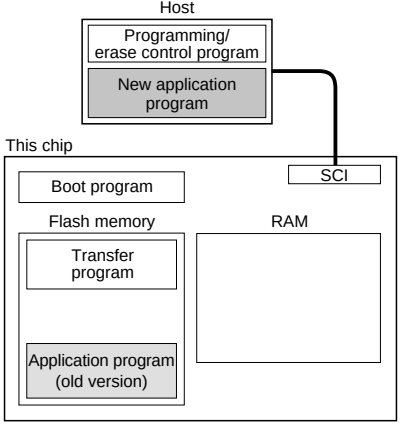
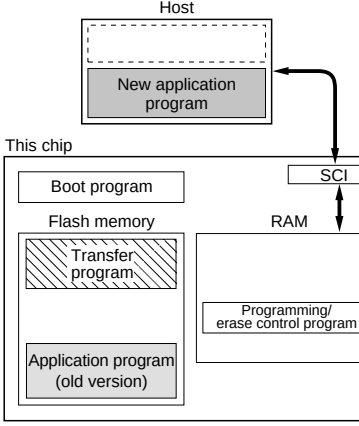


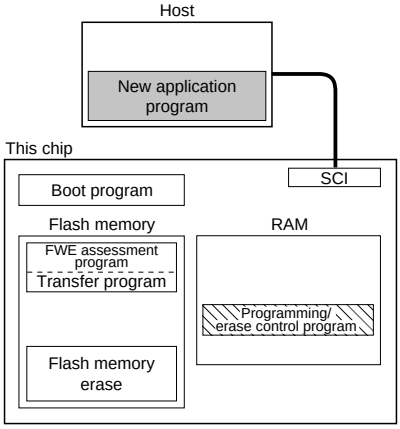
Figure 15.3 Boot Mode



3. Flash memory initialization
The programming/erase program in RAM is executed, and the flash memory is initialized (to H'FF). Erasing can be performed in block units, but not in byte units.



4. Writing new application program
Next, the new application program in the
written into the erased flash memory bloc
not write to unerased blocks.



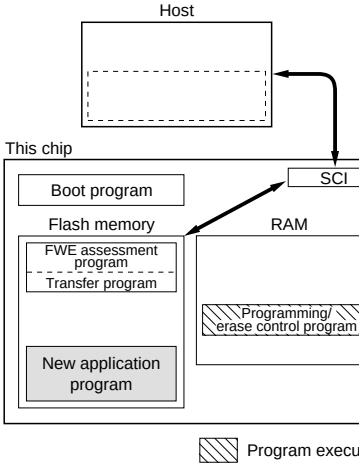


Figure 15.4 User Program Mode (Example)

After the program has been stored the end of writing, execution branches to the top ad
(H'FF300) of the on-chip RAM, execute the program written on the on-chip RAM, an
flash memory program/erase.

The system configuration in boot mode is shown in figure 15.5, and the boot program
execution procedure in figure 15.6.

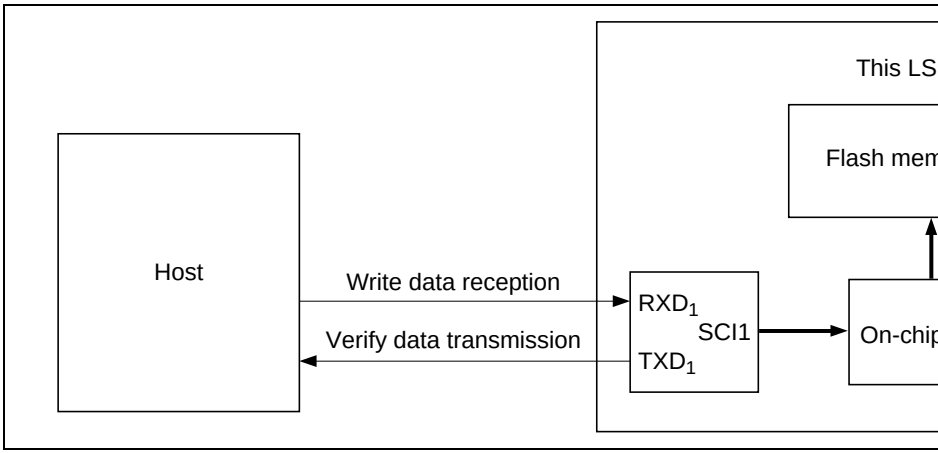
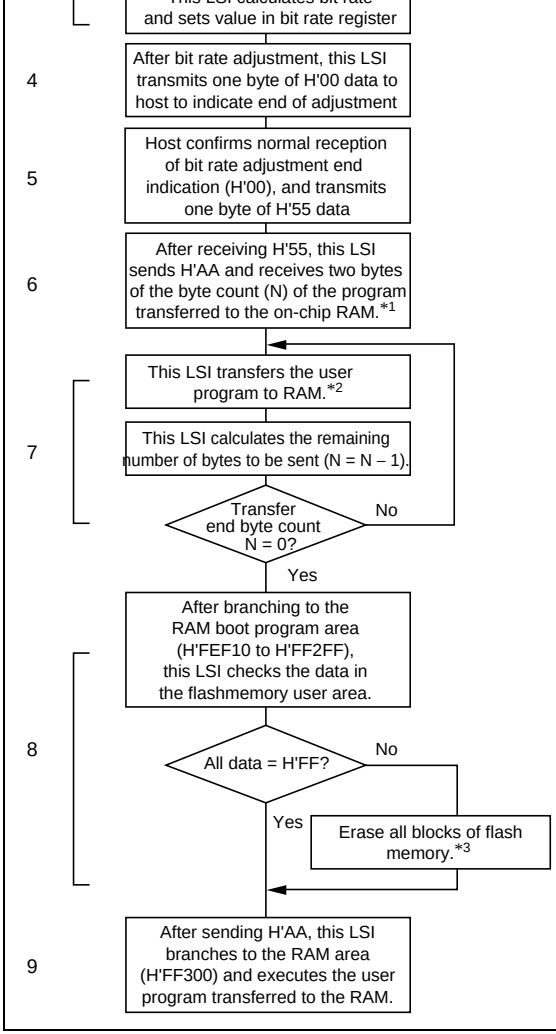


Figure 15.5 System Configuration in Boot Mode



performs transfer.

4 At the end of SCI bit rate adjustment, this L one byte of H'00 data to signal the end of a

5 Check if the host normally received the one rate adjustment end signal sent from this L sent one byte of H'55 data.

6 After H'55 is sent, the host receives H'AA a the byte count of the user program that is transferred to this LSI.
Send the 2-byte count in upper byte and lo order. Then sequentially send the program the user.
This LSI sequentially sends (echo back) ea the received byte count and user program as verification data.

7 This LSI sequentially writes the received us program to the on-chip RAM area (H'FF30

8 Before executing the transferred user prog checks if data was written to flash memory control branched to the RAM boot program (H'FEF10–H'FF2FF). If data was already w flash memory, all the blocks are erased.

9 After sending H'AA, this LSI branches to th RAM area (H'FF300) and executes the use written to that area.

Notes: 1. The RAM area that can be used by the u 3.0 kbytes. Set the transfer byte count to 3.0 kbytes. Always send the 2-byte trans count in upper byte and lower byte order Transfer byte count example: For 256 by upper byte H'01, lower byte H'00.

2. Set the part that controls the user progra memory at the program according to the memory programming/erase algorithms

3. When a memory cell malfunctions and c erased, this LSI sends one H'FF byte as error and stops the erase operation and operations.

Figure 15.6 Boot Mode Execution Procedure

Figure 15.7 Measuring the Low Period of the Communication Data from th

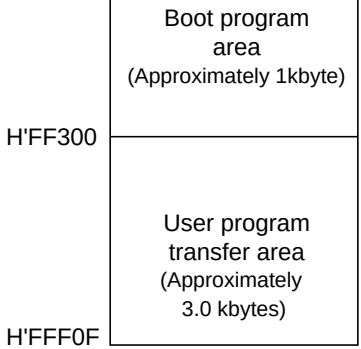
When boot mode is initiated, this LSI measures the low period of the asynchronous SC communication data (H'00) transmitted continuously from the host (figure 15.7). The transmit/receive format should be set as follows: 8-bit data, 1 stop bit, no parity. This calculates the bit rate of the transmission from the host from the measured low period, transmits one H'00 byte to the host to indicate the end of bit rate adjustment. The host confirm that this adjustment end indication (H'00) has been received normally, and tra H'55 byte to the LSI. If reception cannot be performed normally, initiate boot mode ag and repeat the above operations. Depending on the host's transmission bit rate and the clock frequency of this LSI, there will be a discrepancy between the bit rates of the ho LSI. To ensure correct SCI operation, the host's transfer bit rate should be set to 4800 bps*[1].

Table 15.7 shows typical host transfer bit rates and system clock frequencies for whic adjustment of this LSI bit rate is possible. The boot program should be executed withi clock range*[2].

Table 15.7 System Clock Frequencies for which Automatic Adjustment of This Rate Is Possible

Host Bit Rate (bps)	System Clock Frequency for which Automatic Adjustr 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
2. This LSI may automatically adjusts the bit rate except for bit rate and syste combinations as shown in table 15.7. However, the bit rate of the host and be different and subsequent transfers will not be carried out normally. Ther always execute the boot mode within the range of the bit rate and system c combinations shown in table 15.7.



Notes: 1. The boot program area cannot be used until a transition is made to the exe
state for the user program transferred to RAM. Note also that the boot prog
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 pr
transfer area (H'FF800 to H'FFBFF) is used as an area for carrying out emu
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 inpu
SCI's RXD_1 pin. The reset should end with RXD_1 high. After the reset ends, it takes
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 m
blocks are erased when this mode is executed. Therefore, boot mode should be used
on-board programming, or for forced recovery if the program to be activated in user
mode is accidentally erased and user program mode cannot be executed, for exampl

(3) Interrupts cannot be used during programming or erasing of flash memory.

branches to the user program. Since the stack pointer (SP) is implicitly used during 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 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 t 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 reset release*[1]. The following points must be noted:

(a) Before making a transition from the boot mode to the regular mode, the microc boot mode must be reset by reset input via the $\overline{RES}$ pin. At this time, the $\overline{RES}$ p 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 boot mode. When making a mode transition, first enter the reset state by inputt level to the $\overline{RES}$ pin. When a watchdog timer reset was generated in the boot m microcomputer mode is not reset and the on-chip boot program is restarted reg the state of the mode pin.

(c) Do not input low level to the FWE pin while the boot program is executing and 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_{C} during a reset (while a low level is being input to the $\overline{RES}$ pin), the microcomputer mode will change.

Therefore, since the state of the address dual port and bus control output signals ($\overline{A}$ changes, use of these pins as output signals during reset must be disabled outside t microcomputer.

Notes: 1. The mode pin and FWE pin input must satisfy the mode programming setu relative to the reset clear timing.

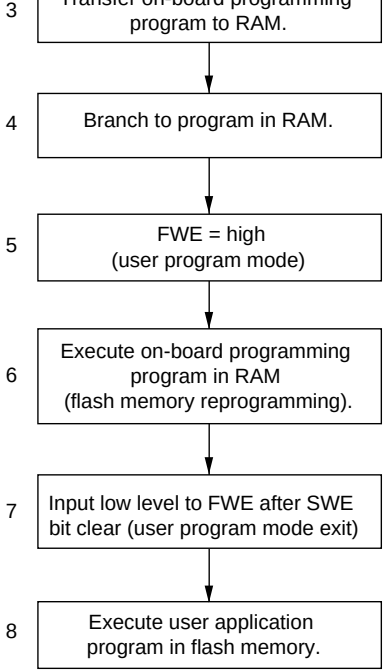
2. For notes on FWE pin High/Low, see section 15.9, Notes on Flash Memor Programming/Erasing.

a user program. Therefore, on-chip flash memory on-board programming can be perfor
providing a means of controlling FWE and supplying the write data on the board and p
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 hig
the FWE pin. In this mode, the peripheral functions, other than flash memory, are perfo
same as in modes 5 and 7.

Since the flash memory cannot be read while it is being programmed/erased, place a pr
program on external memory, or transfer the programming program to RAM area, and
in the RAM. In mode 6, do not program/erase the flash memory. When setting mode 6,
input low level to the FWE pin.

Figure 15.9 shows the procedure for executing when transferred to on-chip RAM. Duri
start, starting from the user program mode is possible.



that is, during the period the RES pin is a lo

3 Transfers the on-board programming progra

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 l
the on-board programming program in RAM
reprograms the user application program in

7 At the end of reprograming, clears the SWE
the user program mode by switching the FW
a high level to a low level.*

8 Branches to, and executes, the user applica
reprogrammed in flash memory.

Note: * For notes on FWE pin High/Low, see sectio
on Flash Memory Programming/Erasing.

Figure 15.9 User Program Mode Execution Procedure (Example)

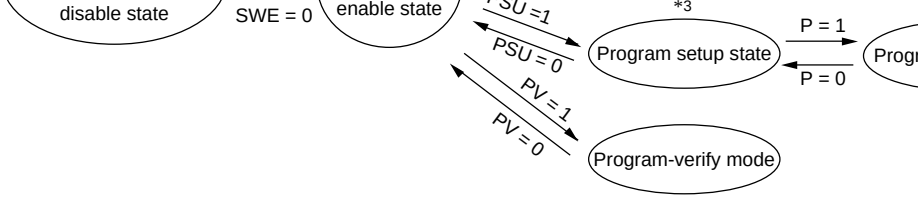
Note: Normally do not apply a high level to the FWE pin. To prevent erroneous prog
erasing in the event of program runaway, etc., apply a high level to the FWE p
when programming/erasing flash memory (including flash memory emulation
If program runaway, etc. causes overprogramming or overerasing of flash me
memory cells will not operate normally.
Also, while a high level is applied to the FWE pin, the watchdog timer should
to prevent overprogramming or overerasing due to program runaway, etc.
In mode 6, do not reprogram flash memory. When setting mode 6, always set
to a low level.

The flash memory cannot be read while being programmed or erased. Therefore, the pr controls flash memory programming/erasing (the programming control program) shoul 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 18.2.5, Flash Memory Characteristics.

Notes: 1. Operation is not guaranteed if setting/resetting of the SWE, ESU, PSU, EV, 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 FWE to 1. (programming/erasing will not be executed if FWE = 0).



Notes: : Normal mode : On-board programming mode

1. Do not make a state transition by setting or clearing two or more bits at the same time.
2. After transition from the erase mode to the erase setup state, do not make a transition to the e without going through the software reprogramming enable state.
3. After transition from the programming mode to the program setup state, do not switch to the pr mode without going through the software reprogramming enable state.

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 or programs to flash memory. Performing program operations according to this flowcl enable data or programs to be written to flash memory without subjecting the device t stress or sacrificing program data reliability. Programming should be carried out 32 by time.

For the wait time (x, y, z, α, β, γ, ε, η) after setting or clearing each bit in the flash me 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 co register (FLMCR), 32-byte program data is stored in the program data area and reprog area, and the 32-byte data in the reprogram data area written consecutively to the write (The lower 8 bits of the first address written to must be H'00, H'20, H'40, H'60, H'80, or H'E0.) 32 consecutive byte data transfers are performed. The program address and p are latched in the flash memory. A 32-byte data transfer must be performed even if wr than 32 bytes; in this case, H'FF data must be written to the extra addresses.

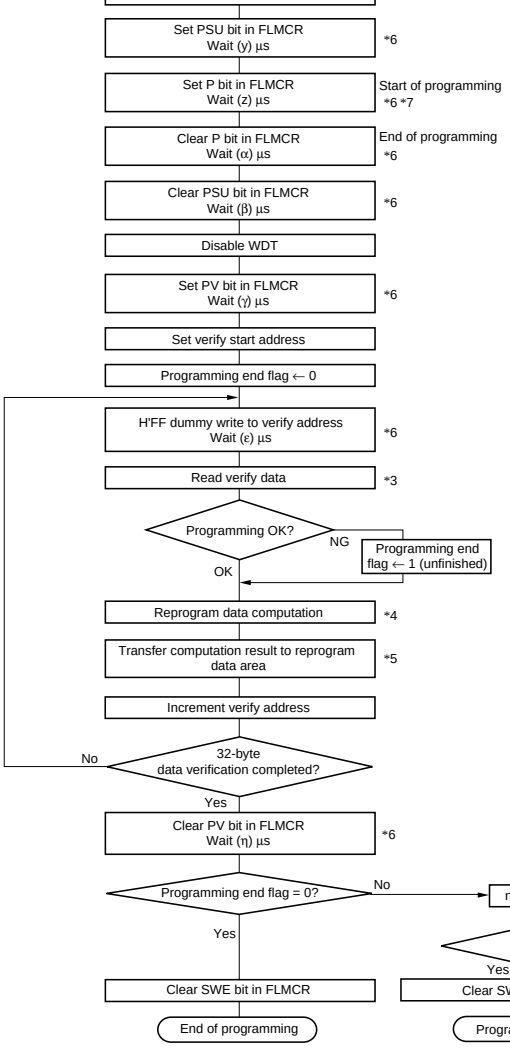
The wait time after P bit setting must be changed according to the number of reprogram loops. For details, see section 18.2.5, Flash Memory Characteristics.

15.5.2 Program-Verify Mode

In program-verify mode, the data written in program mode is read to check whether it h 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 ex mode. After exiting program mode, the watchdog timer setting is also cleared. Then the mode is switched to program-verify mode by setting the PV bit in FLMCR. Before read program-verify mode, a dummy write of H'FF data should be made to the addresses to b The dummy write should be executed after the elapse of (γ) µs or more. When the flash read in this state (verify data is read in 16-bit units), the data at the latched address is re least (ε) µs after the dummy write before performing this read operation. Next, the origi written data is compared with the verify data, and reprogram data is computed (see figu and transferred to RAM. After verification of 32 bytes of data has been completed, exit verify mode, wait for at least (η) µs, then determine whether 32-byte programming has reprogramming is necessary, set program mode again, and repeat the program/program sequence as before. However, ensure that the program/program-verify sequence is not r 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 required in RAM.



possible), with the write start address at a 32-b
The lower 8 bits of the first address written to
H'20, H'40, H'60, H'80, H'A0, H'C0, or H'E0. A
transfer must be performed even if writing few
in this case, H'FF data must be written to the e

3. Verify data is read in 16-bit (word) units. (Byte also possible.)
4. Reprogram data is determined by the computa table below (comparison of data stored in the area with verify data). Programming of reprogr executed in the next programming loop. There which programming has been completed will b again if the result of the subsequent verify ope
5. An area for storing write data (32 bytes) and a reprogram data (32 bytes) must be provided in contents of the latter are rewritten in accordan reprogramming data computation.
6. The values of x, y, z, α, β, γ, ε, η, and N are sh 18.2.5, Flash Memory Characteristics.
7. The value of z depends on the number of repr (n). Details are given in section 18.2.5, Flash Characteristics.

Write Data	Verify Data	Reprogram Data	Comments
0	0	1	Programming complete
0	1	0	Programming incomple
1	0	1	—
1	1	1	Still in erased state; no

Figure 15.11 Program/Program-Verify Flowchart

in erase block register (EBR) at least (x) µs after setting the SWE bit to 1 in FLMCR. N
watchdog timer (WDT) is set to prevent overerasing due to program runaway, etc. Set a
greater than (z) ms + (y + α + ß) µs as the WDT overflow period. Preparation for enter
mode (erase setup) is performed next by setting the ESU bit in FLMCR. The operating
then switched to erase mode by setting the E bit in FLMCR after the elapse of at least (

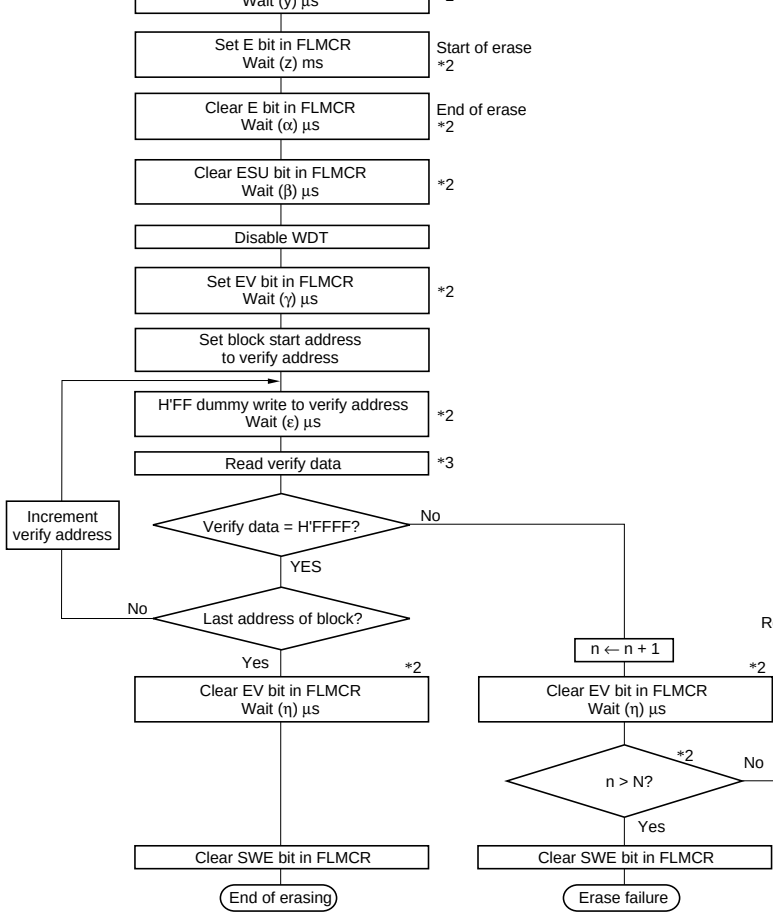
The time during which the E bit is set is the flash memory erase time. Ensure that the e
does not exceed (z) ms.

Note: With flash memory erasing, preprogramming (setting all data in the memory to
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
correctly erased.

After the elapse of the fixed erase time, clear the E bit in FLMCR, then wait for at least
before clearing the ESU bit to exit erase mode. After exiting erase mode, the watchdog
setting is also cleared. The operating mode is then switched to erase-verify mode by set
bit in FLMCR. Before reading in erase-verify mode, a dummy write of H'FF data shoul
to the addresses to be read. The dummy write should be executed after the elapse of (γ)
When the flash memory is read in this state (verify data is read in 16-bit units), the data
latched address is read. Wait at least (ε) µs after the dummy write before performing th
operation. If the read data has been erased (all "1"), a dummy write is performed to the
address, and erase-verify is performed. If the read data is unerased, set erase mode agai
repeat the erase/erase-verify sequence as before. However, do not repeat the erase/erase
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)

disabled or aborted. Hardware protection is reset by settings in the flash memory contro
(FLMCR) and erase block register (EBR). In the case of error protection, the P bit and
set, but a transition is not made to program mode or erase mode. (See table 15.8.)

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]
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]

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 Programm
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 oper system clocks.

		Function	
Item	**Description**	**Program**	**Erase**
Emulation protection*[2]	Setting the RAMS bit in RAMCR sets the program/erase-protected state for all blocks.	No*[2]	No*[3]
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

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.

EBR settings*[3] are retained, but program mode or erase mode is aborted at the point at error occurred. When 1 is set in the FLER bit, transition to the program mode or erase cannot be made even by setting the P and E bits in FLMCR. However, PV and EV bit 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 h 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, disabled. For more information, see section 15.6.4, NMI Input Disable Cor
2. For a detailed description of the FLER bits setting conditions, see section 1 Memory Status Register (FLMSR).
3. Data can be written to FLMCR and EBR. However, when transition to the standby mode was made in the error protection state, the registers are initia

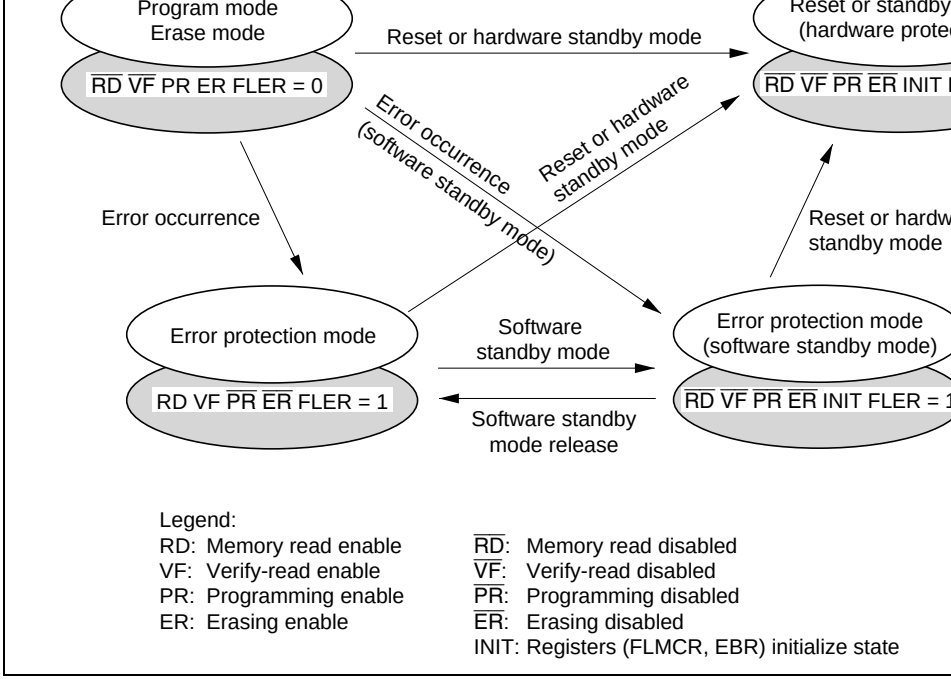


Figure 15.13 Flash Memory State Transitions
(When High Level Apply to FWE Pin in Modes 5 and 7 (On-Chip ROM Ena

The error protection function is disabled for errors other than the FLER bit set conditio considerable time elapses up to transit to this protection state, the flash memory may al damaged. As a result, this function cannot completely protect the flash memory against

Therefore, to prevent such erroneous operation, operation must be carried out correctly according with the program/erase algorithms in the state that flash write enable (FWE) addition, the operation must be always carried out correctly by supervising microcomp inside and outside the chip with the watchdog timer, etc. At transition to this protection flash memory may be erroneously programmed or erased, or its abort may result in inc

This is done to avoid the following operation states:

1. Generation of an NMI input during programming/erasing violates the program/era algorithms and normal operation can not longer be assured.
2. Vector-read cannot be carried out normally*[2] during NMI exception handling duri programming/erasing and the microcomputer runs away as a result.
3. If an NMI input is generated during boot program execution, the normal boot mod cannot be executed.

Therefore, this LSI has conditions that exceptionally disable NMI inputs only in the o 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 ar disabled in the error protection state and the state that holds the P or E bit in FLMCR memory emulation by RAM.

Notes: 1. Indicates the period up to branching to the on-chip RAM boot program are to H'FF2FF). (This branch occurs immediately after user program transfer completed.)

Therefore, after branching to RAM area, NMI input is enabled in states oth program/erase state. Thus, interrupt requests inside and outside the microc must be disabled until initial writing by user program (writing of vector tab processing program, etc.) is completed.

2. In this case, vector read is not performed normally for the following two re
   a. The correct value cannot be read even by reading the flash memory dur programming/erasing. (Value is undefined.)
   b. If a value has not yet been written to the NMI vector table, NMI except will not be performed correctly.

original RAM area (H'FF800 to H'FFBFF). For a description of the RAMCR and RAM 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 memory area is shown below.

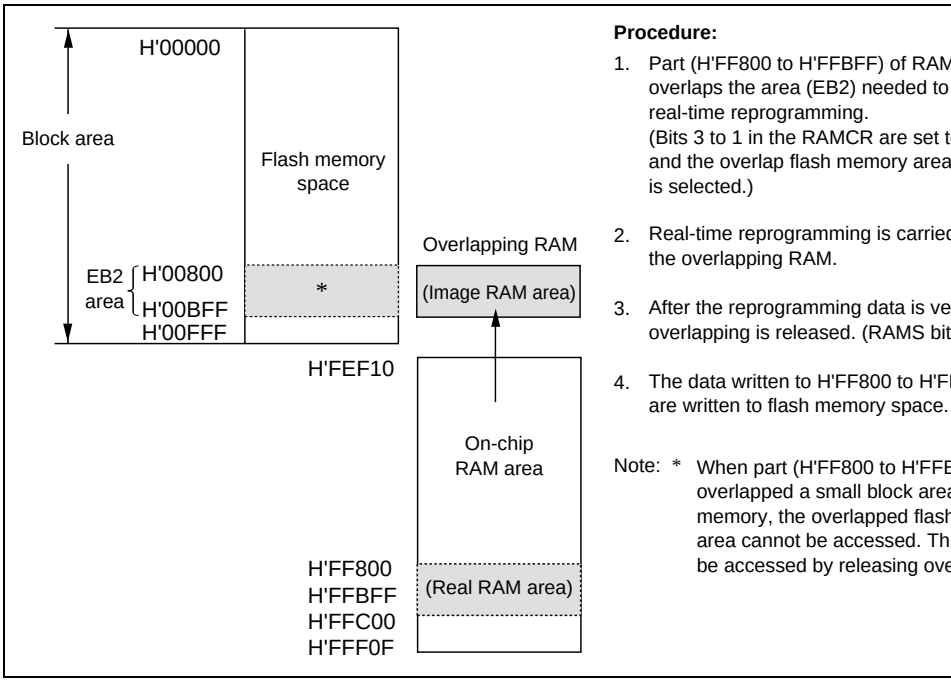


Figure 15.14 Example of RAM Overlapping Operation

(2) NMI input disable conditions

When the P and E bits in FLMCR are set, NMI input is disabled, the same as norm program/erase even when using the emulation function.

NMI input is cleared when the P and E bits are reset (including watchdog timer res standby mode, when a high level is not applied to FWE, and when the SWE bit in 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 mem program/erase mode. In the PROM mode, a program can be freely written to the on-cl using a PROM programmer that supports the Renesas Technology 128 kbytes flash m chip microcomputer device type.

For notes on PROM mode use, see sections 15.8.9, Notes on Memory Programming a Notes on Flash Memory Programming/Erasing.

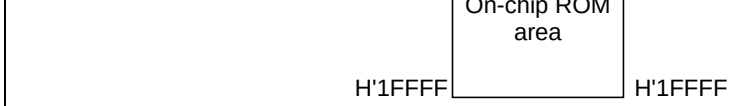


Figure 15.15 PROM Mode Memory Map

15.8.3 PROM Mode Operation

Table 15.10 shows how the different operating modes are set when using PROM mode 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 confirm the end of auto-programming.
- Auto-Erase Mode

  Auto-erase mode supports automatic erasing of the entire flash memory. Status poll to confirm the end of auto-erasing.
- Status Read Mode

  Status polling is used for auto-programming and auto-erasing, and normal terminati confirmed by reading the I/O 6 signal. In status read mode, error information is outp error occurs.

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 m
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

Command Name	Number of Cycles	1st Cycle			2nd Cy	
		Mode	Address	Data	Mode	Addres
Memory read mode	1	Write	X	H'00	Read	RA
Auto-program mode	129	Write	X	H'40	Write	WA
Auto-erase mode	2	Write	X	H'20	Write	X
Status read mode	2	Write	X	H'71	Write	X

Legend:

RA: Read address

WA: Program address

Dout: Read data

Din: Program data

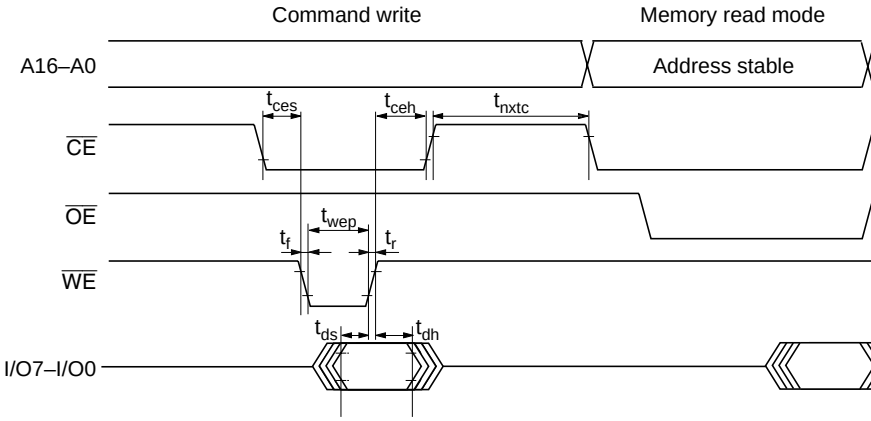
Note: In auto-program mode, 129 cycles are required for command writing by a simul
128-byte write.

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} = –2
Output low voltage	0_7–0_0	V_{OL}	—	—	0.45	V	I_{OL} = 1.
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 Maximum Ratings.

Exceeding the absolute maximum ratings may cause permanent damage to the

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

Output disable delay time	t_{df}	—	100	ns
Data output hold time	t_{oh}	5	—	ns

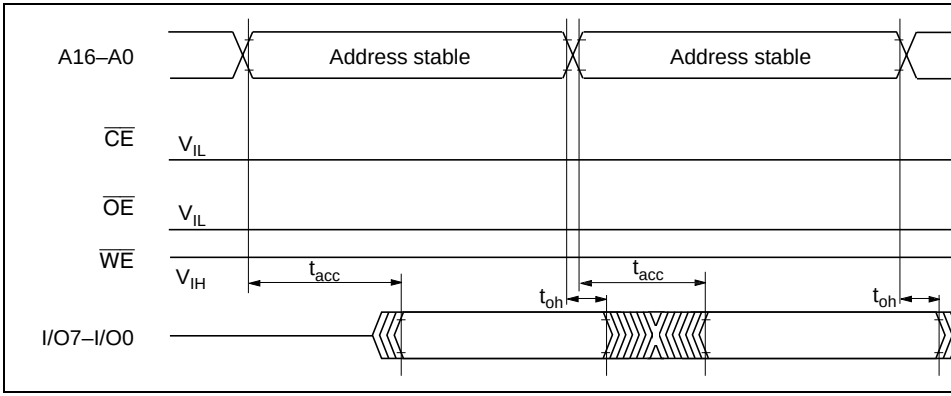


Figure 15.17 $\overline{CE}$/$\overline{OE}$ Enable State Read

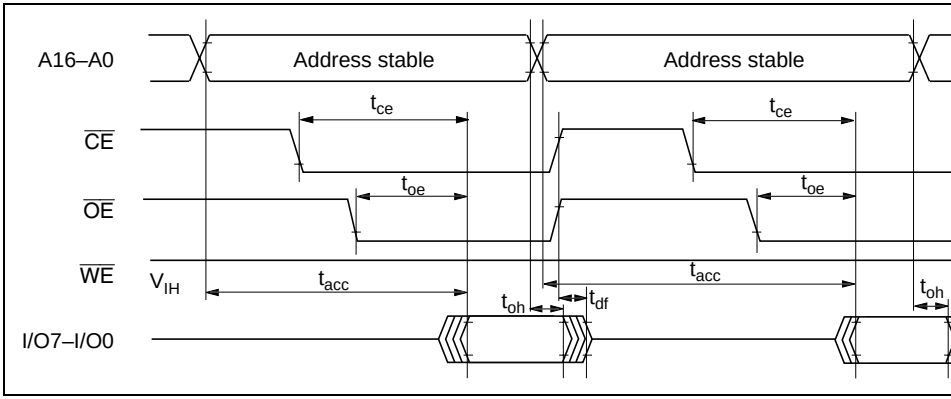
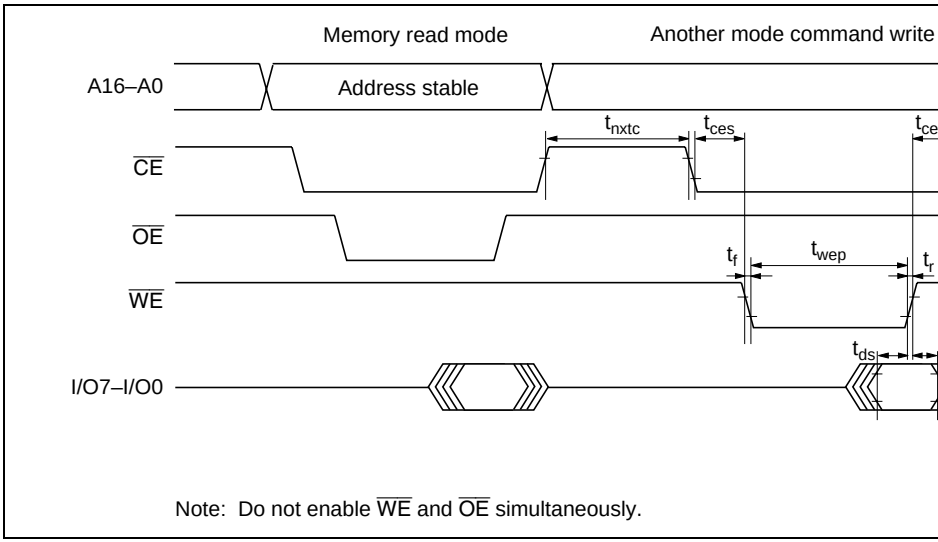


Figure 15.18 $\overline{CE}$/$\overline{OE}$ Clock Read

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 Mod

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

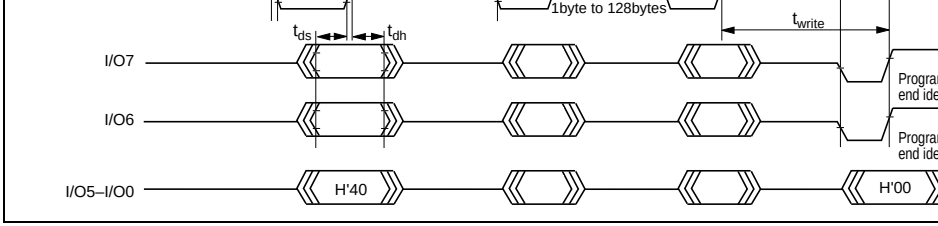


Figure 15.20 Auto-Program Mode Timing Waveforms

Cautions on Use of Auto-Program Mode

- In auto-program mode, 128 bytes are programmed simultaneously. This should be by executing 128 consecutive byte transfers.
- A 128-byte data transfer is necessary even when programming fewer than 128 byt 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 mem operation but a write error will be flagged.
- Memory address transfer is performed in the second cycle (figure 15.20). Do not p 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 can also be used for this purpose.

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

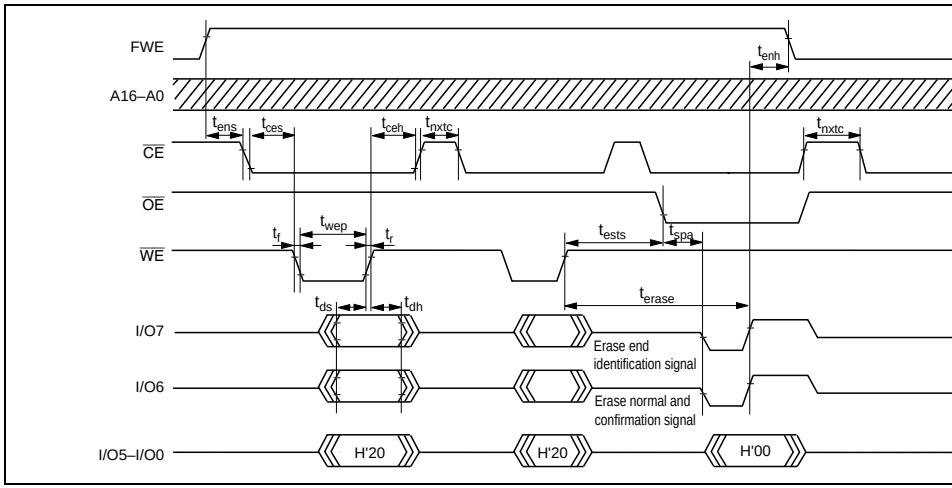
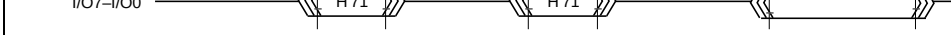


Figure 15.21 Auto-Erase Mode Timing Waveforms

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	



Note: I/O3 and I/O2 are undefined.

Figure 15.22 Status Read Mode Timing Waveforms

Table 15.19 Status Read Mode Return Commands

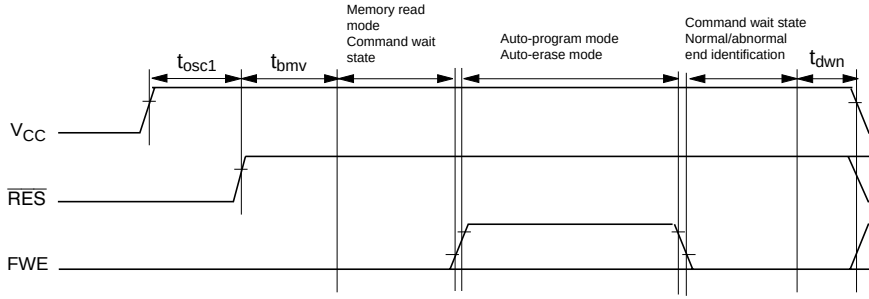
Pin Name	I/O7	I/O6	I/O5	I/O4	I/O3	I/O2	I/O1
Attribute	Normal end identification	Command error	Program-ming error	Erase error	—	—	Program-ming or erase count exceeded
Initial value	0	0	0	0	0	0	0
Indications	Normal end: 0 Abnormal end: 1	Command error: 1 Otherwise: 0	Program-ming error: 1 Otherwise: 0	Erase error: 1 Otherwise: 0	—	—	Count exceeded: 1 Otherwise: 0

Notes on status read mode

After exiting auto-program mode or auto-erase mode, status read mode must be execute
dropping the power supply.

Immediately after powering on, or once powering off, the return command is undefined

settling time)				
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 Ti

Notes: 1. The flash memory is initially in the erased state when the device is shipped
For other chips for which the erasure history is unknown, it is recommended
erasing be executed to check and supplement the initialization (erase) level.

2. In the PROM mode, auto-programming to a 128-byte programming unit blo
be performed only once.

   Do not perform additional programming to a programmed 128-byte progra
block.

   To reprogram, perform auto-programming after auto-erasing.

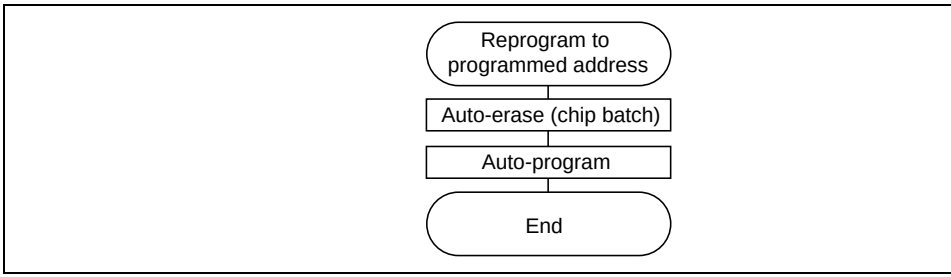


Figure 15.24 Reprogramming to Programmed Address

15.9 Notes on Flash Memory Programming/Erasing

The following describes notes when using the on-board programming mode, RAM emu
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
microcomputer device type.

Do not set the PROM writer at the HN28F101. If the PROM writer is set to the HN
mistake, a high level can be input to the FWE pin and the LSI can be destroyed.

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 satisfy the operation confirmation state, fix the FWE pin at a low level to set the pr mode.

To prevent erroneous programming/erasing of flash memory, note the following in High/Low switching:

- Apply an input to the FWE pin after the Vcc voltage has stabilized within the r

  If an input is applied to the FWE pin when the microcomputer Vcc voltage doe the rated voltage, flash memory may be erroneously programmed or erased bec microcomputer is in the unconfirmed state.
- Apply an input to the FWE pin when the oscillation has stabilized (after the osc stabilization time).

  When turning on the Vcc power, apply an input to the FWE pin after holding th at a low level during the oscillation stabilization time (t_{osc1}=20ms). Do not apply 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 wh input is low. At this time, the FWE and MD_2 to MD_0 inputs must satisfy the mo programming setup time (t_{MDS}) relative to the reset clear timing. The mode prog setup time is necessary for $\overline{RES}$ reset timing even in transition from the boot m another mode.

  In reset during operation, the $\overline{RES}$ pin must be held at a low level for at least 20 clocks.
- In the user program mode, FWE = High/Low switching is possible regardless o input.

  FWE input switching is also possible during program execution on flash memo

To prevent erroneous programming/erasing in the event of program runaway, etc., i level to the FWE pin only when programming/erasing flash memory (including flas emulation by RAM). Avoid system configurations that constantly input a high level FWE pin. Handle program runaway, etc. by starting the watchdog timer so that flas is not overprogrammed/overerased even while a high level is input to the FWE pin.

(5) Program/erase the flash memory in accordance with the recommended algorithms.

The recommended algorithms can program/erase the flash memory without applyin stress to the device or sacrificing the reliability of the program data.

When setting the PSU and ESU bits in FLMCR, set the watchdog timer for program 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 bi

If the SWE bit is set, the flash data can be reprogrammed, but flash memory cannot accessed for purposes other than verify (verify during programming/erase).

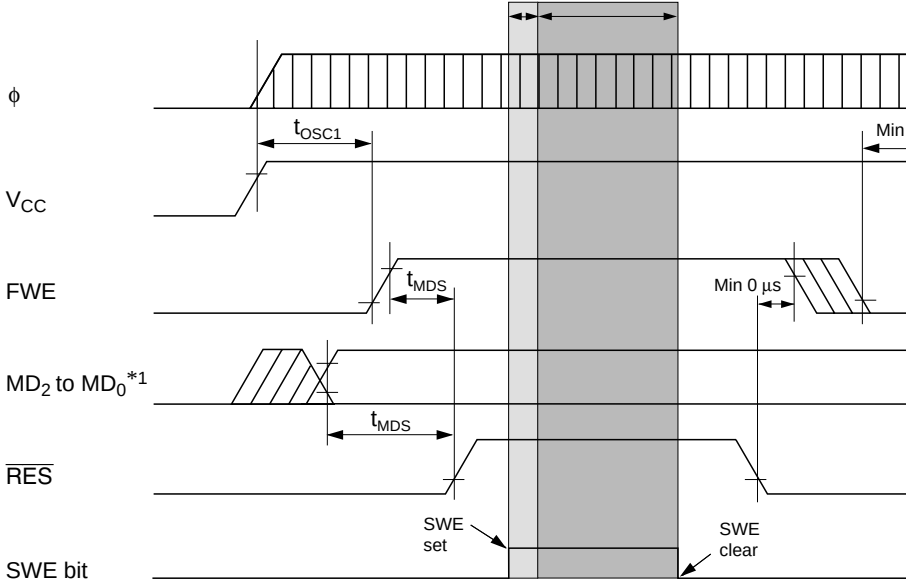
Similarly perform flash memory program execution and data read after clearing the even when using the RAM emulation function with a high level input to the FWE p However, RAM area that overlaps flash memory space can be read/programmed wh SWE bit is set or cleared.

(7) Do not use an interrupt during flash memory programming or erasing.

Since programming/erase operations (including emulation by RAM) have priority v 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 tim Program to 128-byte programming unit blocks one time only even in the PROM mo all the programming unit blocks before reprogramming.
Bus release must also be disabled.

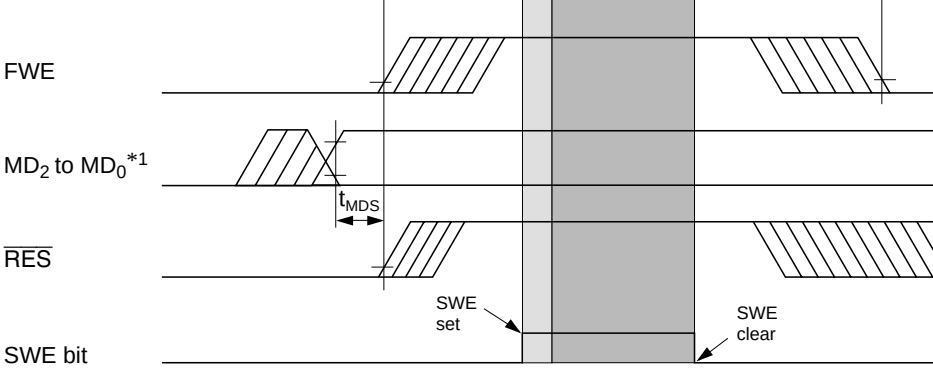


: Flash memory access disabled period
(x: Wait time after SWE setting)*2

: 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) 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)

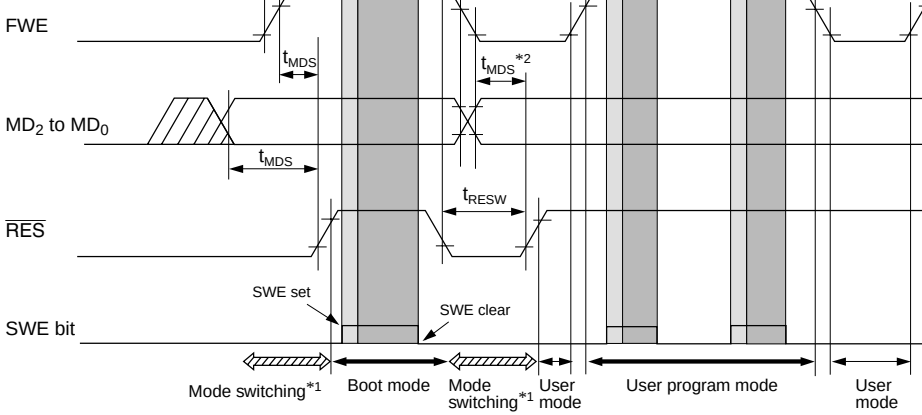


: Flash memory access disabled period
(x: Wait time after SWE setting)*2

: 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 switchin is necessary.
During this switching period (period during which a low level is input to the $\overline{\text{RES}}$ pin), the state of dual port and bus control output signals ($\overline{\text{AS}}$,$\overline{\text{RD}}$,$\overline{\text{WR}}$) 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 tim to the $\overline{\text{RES}}$ 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)

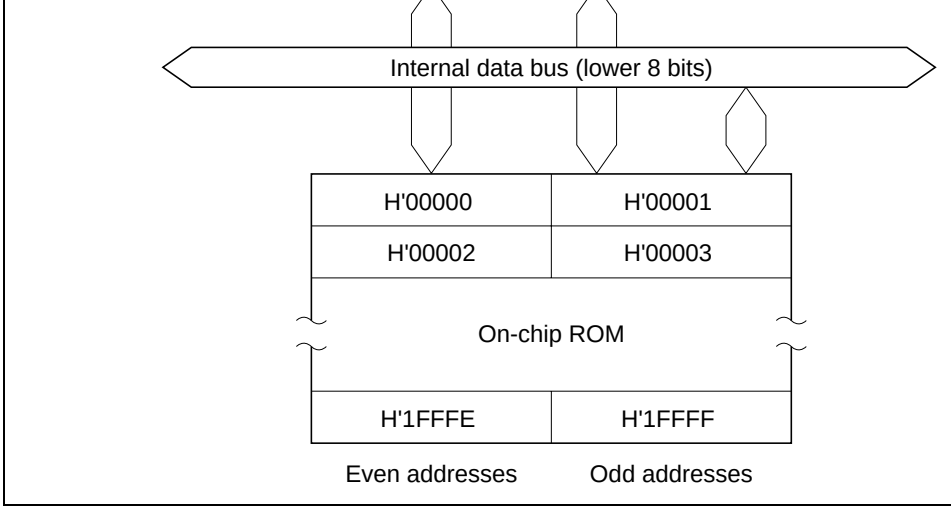


Figure 15.28 ROM Block Diagram (H8/3039)

- The flash memory versions only registers for flash memory control (FLMCR, EBR and FLMSR) are not provided in the mask ROM versions. Reading the correspond addresses in a mask ROM version will always return 1s, and writes to these addres disabled. This must be borne in mind when switching from the flash memory versi mask ROM version.

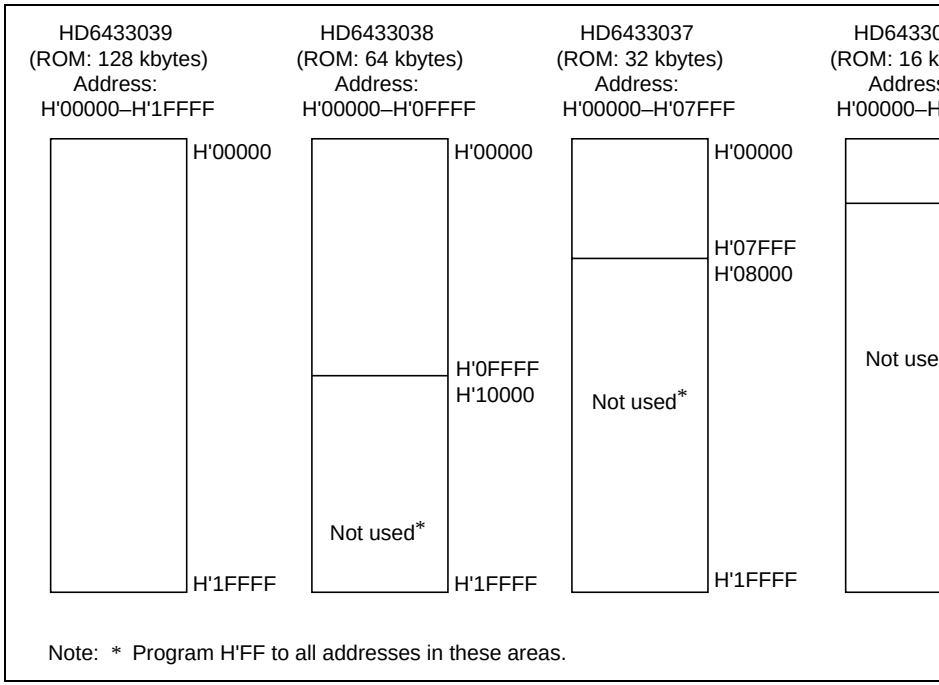


Figure 15.29 Mask ROM Addresses and Data

as a master clock to prescalers that supply clock signals to the on-chip supporting mod
Frequency division ratios of 1/1, 1/2, 1/4, and 1/8 can be selected for the frequency di
settings in a division control register (DIVCR). Power consumption in the chip is redu
almost direct proportion to the frequency division ratio*[2].

Notes: 1. Usage of the ϕ pin differs depending on the chip operating mode and the P
setting in the module standby control register (MSTCR). For details, see se
System Clock Output Disabling Function.

2. The division ratio of the frequency divider can be changed dynamically du
operation. The clock output at the ϕ pin also changes when the division rat
changed. The frequency output at the ϕ pin is shown below.

$$\phi = \mathrm{EXTAL} \times n$$

EXTAL: Frequency of crystal resonator or external clock sign
n: Frequency division ratio (n = 1/1, 1/2, 1/4, or 1/8)

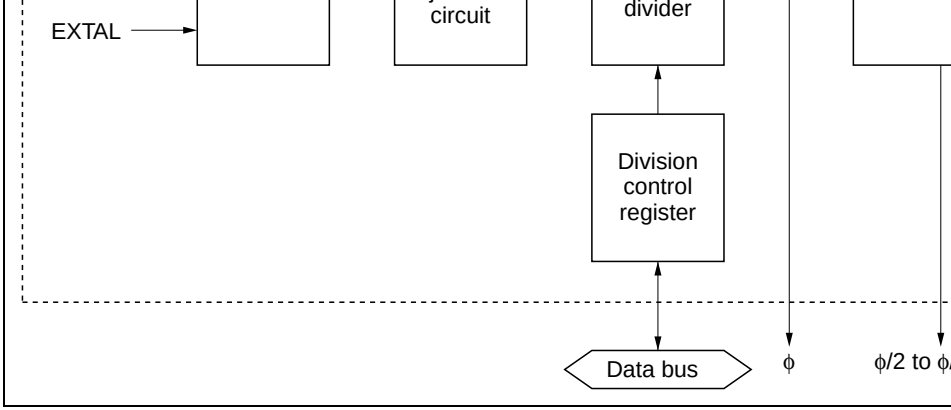


Figure 16.1 Block Diagram of Clock Pulse Generator

16.2 Oscillator Circuit

Clock pulses can be supplied by connecting a crystal resonator, or by input of an extern
signal.

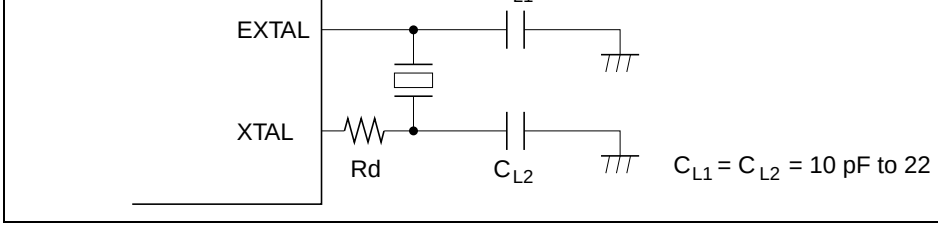


Figure 16.2 Connection of Crystal Resonator (Example)

Table 16.1 Damping Resistance Value (Example)

Frequency (MHz)	2	4	8	10	12	16
Rd (Ω)	1 k	500	200	0	0	0

Crystal Resonator

Figure 16.3 shows an equivalent circuit of the crystal resonator. The crystal resonator
the characteristics listed in table 16.2.

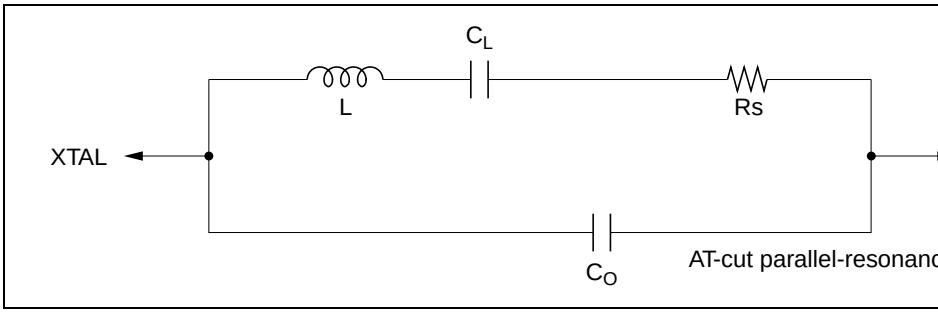


Figure 16.3 Crystal Resonator Equivalent Circuit

Notes on Board Design

When a crystal resonator is connected, the following points should be noted:

Other signal lines should be routed away from the oscillator circuit to prevent induction interfering with correct oscillation. See figure 16.4.

When the board is designed, the crystal resonator and its load capacitors should be plac as possible to the XTAL and EXTAL pins.

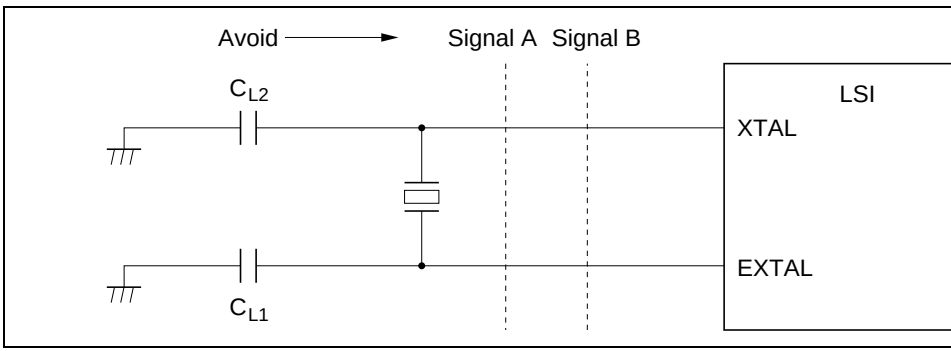


Figure 16.4 Example of Incorrect Board Design

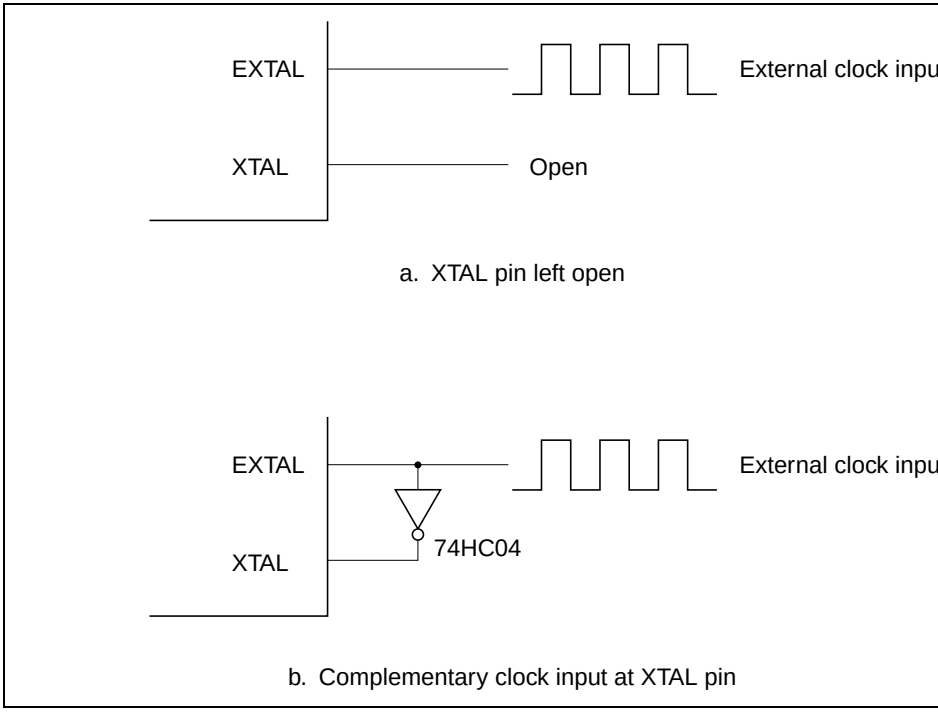


Figure 16.5 External Clock Input (Examples)

Item	Symbol	Min	Max	Min	Max	Unit	Test Cond
External clock rise time	t_{EXr}	—	10	—	5	ns	Figure 16.
External clock fall time	t_{EXf}	—	10	—	5	ns	
External clock	—	30	70	30	70	%	$\phi \geq 5$ MHz
input duty (a/t_{cyc})		40	60	40	60	%	$\phi < 5$ MHz
ϕ clock width duty (b/t_{cyc})	—	40	60	40	60	%	

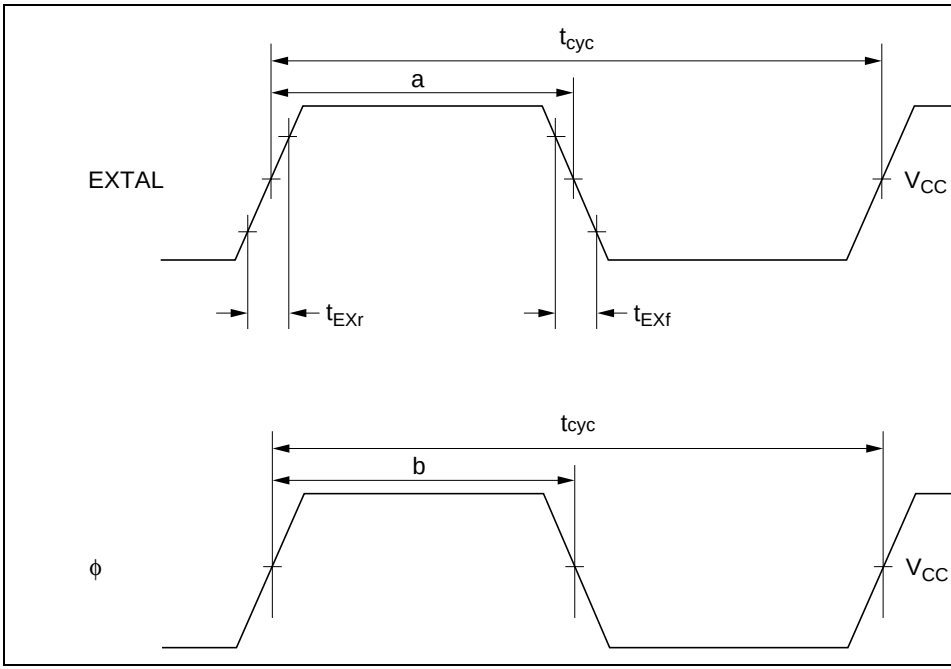


Figure 16.6 External Clock Input Timing

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	No
External clock output stabilization delay time	t_{DEXT}*	500	—	µs	Fig

Note: * t_{DEXT} includes a 10 t_{cyc} $\overline{RES}$ pulse width (t_{RESW}).

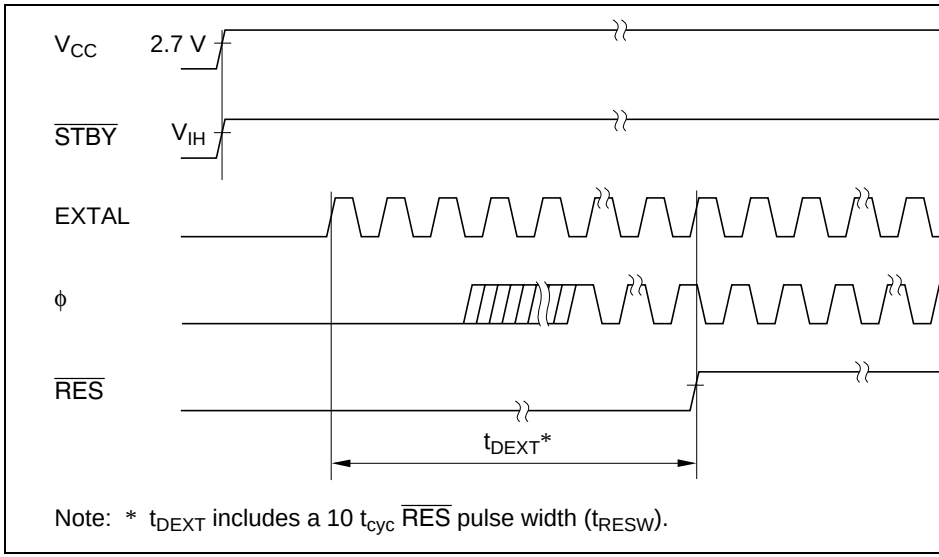


Figure 16.7 External Clock Output Stabilization Delay Time

16.5 Frequency Divider

The frequency divider divides the duty-adjusted clock signal to generate the system clo
frequency division ratio can be changed dynamically by modifying the value in DIVCR
described below. Power consumption in the chip is reduced in almost direct proportion
frequency division ratio. The system clock generated by the frequency divider can be o
ϕ pin.

16.5.1 Register Configuration

Table 16.5 summarizes the frequency division register.

Table 16.5 Frequency Division Register

Address*	Name	Abbreviation	R/W	Initia
H'FF5D	Division control register	DIVCR	R/W	H'FC

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

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

Bits 7 to 2—Reserved: These bits cannot be modified and are always read as 1.

Bits 1 and 0—Divide (DIV1 and DIV0): These bits select the frequency division rati
follows.

Bit 1 DIV1	Bit 0 DIV0	Frequency Division Ratio	
0	0	1/1	(I
0	1	1/2	
1	0	1/4	
1	1	1/8	

communication, and other time-dependent processing differs before and after any chan
division ratio. The waiting time for exit from software standby mode also changes whe
division ratio is changed. For details, see section 17.4.3, Selection of Oscillator Waiting
Exit from Software Standby Mode

- Sleep mode
- Software standby mode
- Hardware standby mode

The module standby function can halt on-chip supporting modules independently of th
down state. The modules that can be halted are the ITU, SCI0, SCI1, and A/D convert

Table 17.1 indicates the methods of entering and exiting these power-down modes and
of the CPU and on-chip supporting modules in each mode.

Mode	Entering Conditions	State										
		Clock	CPU	CPU Registers	ITU	SCI0	SCI1	A/D	Supporting Modules	RAM	φ clock output	I/O Ports
Sleep mode	SLEEP instruction executed while SSBY = 0 in SYSCR	Active	Halted	Held	Active	Active	Active	Active	Active	Held	φoutput	Held
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
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 impeda
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	—

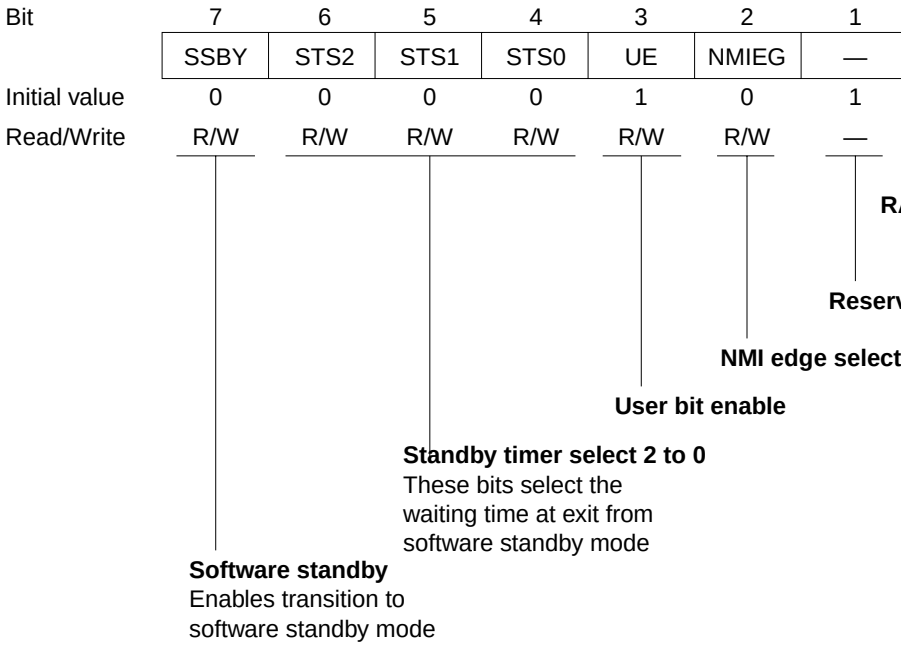
Legend:
SYSCR: System control register
SSBY: Software standby bit
MSTCR: Module standby control register

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, fi
clear the MSTCR bit to 0, then set up the module registers again.

H'FFF2	System control register	SYSCR	R/W	H'0
H'FF5E	Module standby control register	MSTCR	R/W	H'4

Note: * Lower 16 bits of the address.

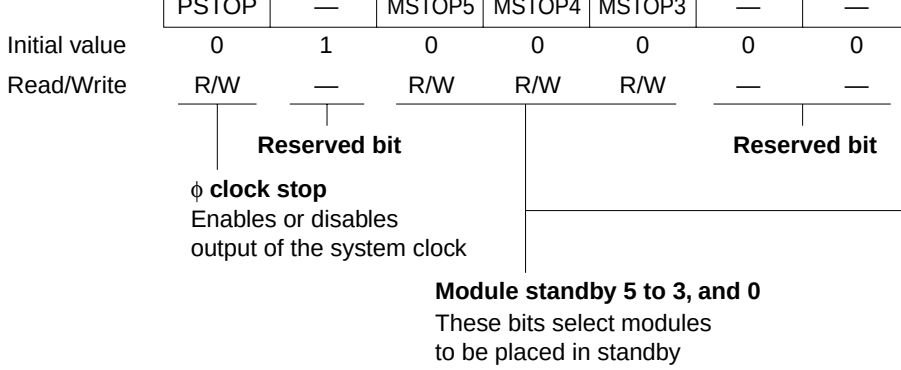
17.2.1 System Control Register (SYSCR)



SYSCR is an 8-bit readable/writable register. Bit 7 (SSBY) and bits 6 to 4 (STS2 to S the power-down state. For information on the other SYSCR bits, see section 3.3, Syste Register (SYSCR).

Bits 6 to 4—Standby Timer Select (STS2 to STS0): These bits select the length of tir and on-chip supporting modules wait for the clock to settle when software standby mod by an external interrupt. If the clock is generated by a crystal resonator, set these bits ac the clock frequency so that the waiting time (for the clock to stabilize) will be at least 7 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
		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	



MSTCR is initialized to H'40 by a reset and in hardware standby mode. It is not initial software standby mode.

Bit 7—ϕ Clock Stop (PSTOP): Enables or disables output of the system clock (ϕ).

Bit 7 PSTOP	Description	
0	System clock output is enabled	(I
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	(I
1	ITU is in standby state	

Bit 3 MSTOP3	Description	
0	SCI1 operates normally	(Ini
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

Bit 0 MSTOP0	Description	
0	A/D converter operates normally	(Ini
1	A/D converter is in standby state	

supporting modules which have been placed in standby by the module standby functio
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 i
exception handling state. Sleep mode is not exited by an interrupt source in an on-chip
module if the interrupt is disabled in the on-chip supporting module. Sleep mode is no
an interrupt other then NMI if the interrupt is masked by interrupt priority settings (IP
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
mode.

CPU, clock, and on-chip supporting modules all halt. The on-chip supporting modules
and halted. As long as the specified voltage is supplied, however, CPU register content
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, $\overline{\text{IRQ}}$
by input at the $\overline{\text{RES}}$ or $\overline{\text{STBY}}$ pin.

Exit by Interrupt: When an NMI, IRQ_0, or IRQ_1 interrupt request signal is received, th
oscillator begins operating. After the oscillator settling time selected by bits STS2 to S
SYSCR, stable clock signals are supplied to the entire chip, software standby mode end
interrupt exception handling begins. Software standby mode is not exited if the interrup
bits of interrupts IRQ_0, and IRQ_1 are cleared to 0, or if these interrupts are masked in th

Exit by $\overline{\text{RES}}$ Input: When the $\overline{\text{RES}}$ input goes low, the clock oscillator starts and clock
supplied immediately to the entire chip. The $\overline{\text{RES}}$ signal must be held low long enough
clock oscillator to stabilize. When $\overline{\text{RES}}$ goes high, the CPU starts reset exception handl

Exit by $\overline{\text{STBY}}$ Input: Low input at the $\overline{\text{STBY}}$ pin causes a transition to hardware stand

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
0	0	0	0	0	8192 states	0.46	0.51	0.65	0.8	1.0	1.3	2.0	4.1
		0	0	1	16384 states	0.91	1.0	1.3	1.6	2.0	2.7	4.1	8.2
		0	1	0	32768 states	1.8	2.0	2.7	3.3	4.1	5.5	8.2	16.4
		0	1	1	65536 states	3.6	4.1	5.5	6.6	8.2	10.9	16.4	32.8
		1	0	0	131072 states	7.3	8.2	10.9	13.1	16.4	21.8	32.8	65.5
		1	0	1	1024 states	0.057	0.064	0.085	0.10	0.13	0.17	0.26	0.51
		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
		0	0	1	16384 states	1.8	2.0	2.7	3.3	4.1	5.5	8.2	16.4
		0	1	0	32768 states	3.6	4.1	5.5	6.6	8.2	10.9	16.4	32.8
		0	1	1	65536 states	7.3	8.2	10.9	13.1	16.4	21.8	32.8	65.5
		1	0	0	131072 states	14.6	16.4	21.8	26.2	32.8	43.7	65.5	131.1
		1	0	1	1024 states	0.11	0.13	0.17	0.20	0.26	0.34	0.51	1.0
		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
		0	0	1	16384 states	3.6	4.1	5.5	6.6	8.2	10.9	16.4	32.8
		0	1	0	32768 states	7.3	8.2	10.9	13.1	16.4	21.8	32.8	65.5
		0	1	1	65536 states	14.6	16.4	21.8	26.2	32.8	43.7	65.5	131.1
		1	0	0	131072 states	29.1	32.8	43.7	52.4	65.5	87.4	131.1	262.1
		1	0	1	1024 states	0.23	0.26	0.34	0.41	0.51	0.68	1.02	2.0
		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
		0	0	1	16384 states	7.3	8.2	10.9	13.1	16.4	21.8	32.8	65.5
		0	1	0	32768 states	14.6	16.4	21.8	26.2	32.8	43.7	65.5	131.1
		0	1	1	65536 states	29.1	32.8	43.7	52.4	65.5	87.4	131.1	262.1
		1	0	0	131072 states	58.3	65.5	87.4	104.9	131.1	174.8	262.1	524.3
		1	0	1	1024 states	0.46	0.51	0.68	0.82	1.0	1.4	2.0	4.1
		1	1	—	Illegal setting								

: Recommended setting

Software standby mode is exited at the next rising edge of the NMI signal.

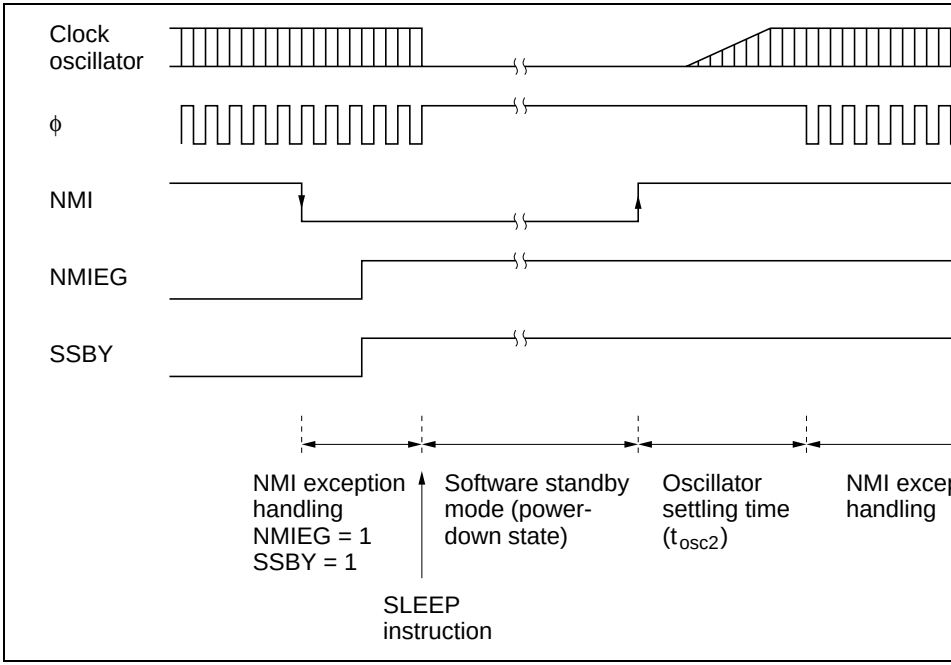


Figure 17.1 NMI Timing for Software Standby Mode (Example)

17.4.5 Usage Note

The I/O ports retain their existing states in software standby mode. If a port is in the hig state, its output current is not reduced.

high-impedance state.

Clear the RAME bit to 0 in SYSCR before $\overline{\text{STBY}}$ goes low to retain on-chip RAM da

The inputs at the mode pins (MD_2 to MD_0) should not be changed during hardware sta

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 $\overline{\text{STBY}}$ goes high, the clock oscillator starts running. $\overline{\text{RES}}$ should be held low long eno clock oscillator to settle. When $\overline{\text{RES}}$ goes high, reset exception handling begins, follo transition to the program execution state.

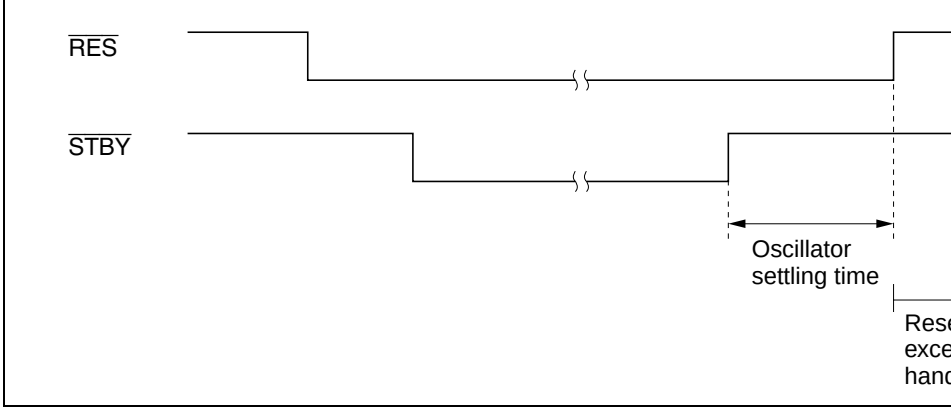


Figure 17.2 Hardware Standby Mode Timing

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 reg
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
by the module standby function, its registers are initialized, including registers with in
request flags. Consequently, if an interrupt occurs just before the MSTOP bit is set to
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 wi
module is placed in module standby. What happens after that depends on the particula
details, see section 7, I/O Ports. Pins that change from the input to the output state req
care. For example, if SCI1 is placed in module standby, the receive data pin loses its r
function and becomes a generic I/O pin. If its data direction bit is set to 1, the pin beco
output pin, and its output may collide with external serial data. Data collisions should
by clearing the data direction bit to 0 or taking other appropriate action.

Register Resetting: When an on-chip supporting module is halted by the module stan
function, all its registers are initialized. To restart the module, after its MSTOP bit is c
its registers must be set up again. It is not possible to write to the registers while the M
set to 1.

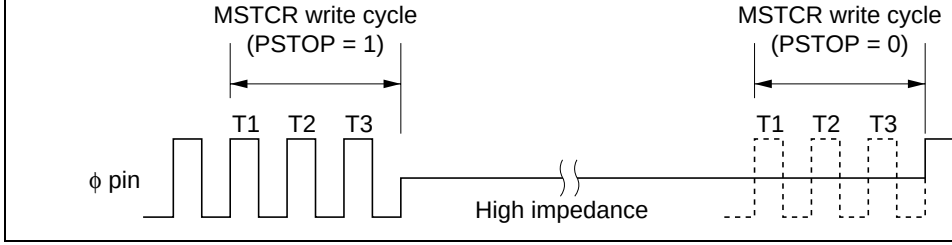


Figure 17.3 Starting and Stopping of System Clock Output

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

Table 18.1 Absolute Maximum Ratings

Item	Symbol	Value
Power supply voltage	V_{CC}	–0.3 to +7.0
Input voltage (except port 7)*	V_{in}	–0.3 to V_{CC} +0.3
Input voltage (port 7)	V_{in}	–0.3 to AV_{CC} +0.3
Analog power supply voltage	AV_{CC}	–0.3 to +7.0
Analog input voltage	V_{AN}	–0.3 to AV_{CC} +0.3
Operating temperature	T_{opr}	Regular specifications: –20 to +75
		Wide-range specifications: –40 to +85
Storage temperature	T_{stg}	–55 to +125

Caution: Permanent damage to the chip may result if absolute maximum ratings are

Note: * 12 V must not be applied to any pin, as this will cause permanent damage t

Item		Symbol	Min	Typ	Max	Unit	Test C
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} = -
			3.5	—	—	V	I_{OH} = -
Output low voltage	All output pins (except $\overline{RESO}$)	V_{OL}	—	—	0.4	V	I_{OL} = 1
	Ports 1, 2, 5, B		—	—	1.0	V	I_{OL} = 1
	$\overline{RESO}$		—	—	0.4	V	I_{OL} = 2

(off state)	RESO		—	—	10.0	µA	
Input pull-up MOS current	Ports 2, 5	$-I_p$	50	—	300	µA	V_{in} =
Input capacitance	NMI, $\overline{RES}$	C_{in}	—	—	50	pF	V_{in} =
	All input pins except NMI and $\overline{RES}$		—	—	20		f = 1 T_a =
Current dissipation*[2]	Normal operation	I_{CC}	—	50	70	mA	f = 1
	Sleep mode		—	35	50		f = 1
	Standby mode*[3]		—	0.01	5.0	µA	$T_a \leq$
			—	—	20.0		50°C
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 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

		$V_T - V_T$	$V_{CC} \times 0.04$	—	—	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} <
					0.8	V	V_{CC} = 4.0 V
Output high voltage	All output pins (except $\overline{RESO}$)	V_{OH}	V_{CC} –0.5	—	—	V	I_{OH} = –
			V_{CC} –1.0	—	—	V	I_{OH} = –
Output low voltage	All output pins (except $\overline{RESO}$)	V_{OL}	—	—	0.4	V	I_{OL} = 1
	Ports 1, 2, 5, B		—	—	1.0	V	$V_{CC} \leq$ I_{OL} = 5 4 V < I_{OL} = 1
	$\overline{RESO}$		—	—	0.4	V	I_{OL} = 1
Input leakage current	$\overline{STBY}$, NMI, $\overline{RES}$, MD_2, MD_1, MD_0	$\lvert I_{in} \rvert$	—	—	1.0	µA	V_{in} = 0 V_{CC} –0
	Port 7		—	—	1.0	µA	V_{in} = 0 AV_{CC} +

Input capacitance	NMI, RES	C_{in}	—	—	30	pF	V_{in}
	All input pins except NMI and $\overline{RES}$		—	—	20		f = 1 Ta =
Current dissipation*2	Normal operation	I_{CC}*4	—	12 (3.0 V)	33.8 (5.5 V)	mA	f = 8
	Sleep mode		—	8 (3.0 V)	25.0 (5.5 V)	mA	f = 8
	Standby mode*3		—	0.01	5.0	µA	$T_a \leq 5$
			—	—	20.0	µA	50°C
Analog power supply current	During A/D conversion	AI_{CC}	—	1.3	2.5	mA	AV_{CC}
			—	1.7	2.8	mA	AV_{CC}
	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 w output pins unloaded and the on-chip pull-up MOS transistors in the off stat

3. The values are for $V_{RAM} \leq V_{CC} < 2.7$ V, V_{IH} min = $V_{CC} \times 0.9$, and V_{IL} max = 0.3

4. I_{CC} depends on V_{CC} and f as follows:
   I_{CC} max = 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]

		$V_T - V_T$	$V_{CC} \times 0.04$	—	—	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} <
					0.8	V	V_{CC} = 4 5.5 V
Output high voltage	All output pins (except $\overline{RESO}$)	V_{OH}	V_{CC} –0.5	—	—	V	I_{OH} = –
			V_{CC} –1.0	—	—	V	I_{OH} = –
Output low voltage	All output pins (except $\overline{RESO}$)	V_{OL}	—	—	0.4	V	I_{OL} = 1
	Ports 1, 2, 5, B		—	—	1.0	V	$V_{CC} \leq$ I_{OL} = 5 4 V < I_{OL} = 1
	$\overline{RESO}$		—	—	0.4	V	I_{OL} = 1
Input leakage current	$\overline{STBY}$, NMI, $\overline{RES}$, MD_2, MD_1, MD_0	$\lvert I_{in} \rvert$	—	—	1.0	µA	V_{in} = 0 V_{CC} –0
	Port 7		—	—	1.0	µA	V_{in} = 0 AV_{CC} +

Input capacitance	NMI, RES	C_{in}	—	—	30	pF	V_{in} =
	All input pins except NMI and $\overline{RES}$		—	—	20		f = 1 T_a =
Current dissipation*[2]	Normal operation	I_{CC}*[4]	—	15 (3.0 V)	41.5 (5.5 V)	mA	f = 1
	Sleep mode		—	10 (3.0 V)	30.5 (5.5 V)	mA	f = 1
	Standby mode*[3]		—	0.01	5.0	µA	$T_a \leq$
			—	—	20.0	µA	50°C
Analog power supply current	During A/D conversion	AI_{CC}	—	1.3	2.5	mA	AV_{CC}
			—	1.7	—	mA	AV_{CC}
	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 w output pins unloaded and the on-chip pull-up transistors in the off state.

3. The values are for $V_{RAM} \leq V_{CC} < 3.0$ V, V_{IH} min = $V_{CC} \times 0.9$, and V_{IL} max = 0.3

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]

Permissible output low current (total)	Total of 27 pins including ports 1, 2, 5 and B	ΣI_{OL}	—	—	80
	Total of 23 pins, including ports 8, 9, A and B		—	—	75*[2]/65*
	Total of all output pins, including the above		—	—	120
Permissible output high current (per pin)	All output pins	I_{OH}	—	—	2.0
Permissible output high current (total)	Total of all output pins	ΣI_{OH}	—	—	40

Notes: 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 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

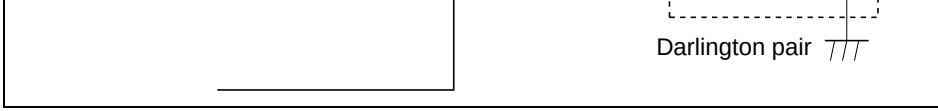


Figure 18.1 Darlington Pair Drive Circuit (Example)

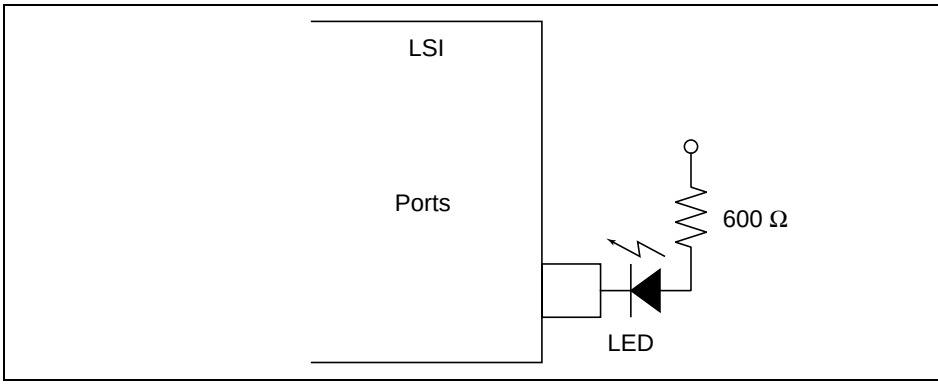


Figure 18.2 LED Drive Circuit (Example)

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 M
10 MHz, T_a = –20°C to +75°C (regular specifications), T_a = –40°C to +8
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
T_a = –20°C to +75°C (regular specifications), T_a = –40°C to +85°C (wid
specifications)

		Condition A		Condition B		Condition C		
		8 MHz		10 MHz		18 MHz		
Item	**Symbol**	**Min**	**Max**	**Min**	**Max**	**Min**	**Max**	**Unit**
Clock cycle time	t_{cyc}	125	500	100	500	55.5	500	ns
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	—	

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	—
Wait hold time	t_{WTH}	10	—	10	—	5	—

Note: * For Condition A, the following times depend on the clock cycle time as shown

$t_{ACC1} = 1.5 \times t_{cyc} - 68$ (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

$t_{ACC1} = 1.5 \times t_{cyc} - 50$ (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 show

$t_{ACC1} = 1.5 \times t_{cyc} - 34$ (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)

Condition C: V_{CC} = 5.0 V ±10%, AV_{CC} = 5.0 V ±10%, V_{SS} = AV_{SS} = 0 V, φ = 2 MHz to
T_a = –20°C to +75°C (regular specifications), T_a = –40°C to +85°C (wid
specifications)

		Condition A		Condition B		Condition C			
		8 MHz		10 MHz		18 MHz			
Item	**Symbol**	**Min**	**Max**	**Min**	**Max**	**Min**	**Max**	**Unit**	**Tes**
$\overline{RES}$ setup time	t_{RESS}	200	—	200	—	200	—	ns	Figu
$\overline{RES}$ 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	Figu
$\overline{RESO}$ 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	Figu
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	Figu
Clock oscillator settling time in software standby (crystal)	t_{OSC2}	8	—	8	—	7	—	ms	Figu

Condition C: V_{CC} = 5.0 V ±10%, AV_{CC} = 5.0 V ±10%, V_{SS} = AV_{SS} = 0 V, ϕ = 2 MHz
T_a = –20°C to +75°C (regular specifications), T_a = –40°C to +85°C (wi
specifications)

				Condition A		Condition B		Condition C		
				8 MHz		10 MHz		18 MHz		
Item			**Symbol**	**Min**	**Max**	**Min**	**Max**	**Min**	**Max**	**Unit**
ITU	Timer output delay time		t_{TOCD}	—	100	—	100	—	100	ns
	Timer input setup time		t_{TICS}	50	—	50	—	50	—	
	Timer clock input setup time		t_{TCKS}	50	—	50	—	50	—	
	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	Asynchro-nous	t_{Scyc}	4	—	4	—	4	—	
		Synchro-nous		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}

	clock input)								
	Receive data hold time (synchronous clock output)		0	—	0	—	0	—	
Ports and TPC	Output data delay time	t_{PWD}	—	100	—	100	—	100	ns
	Input data setup time	t_{PRS}	50	—	50	—	50	—	
	Input data hold time	t_{PRH}	50	—	50	—	50	—	

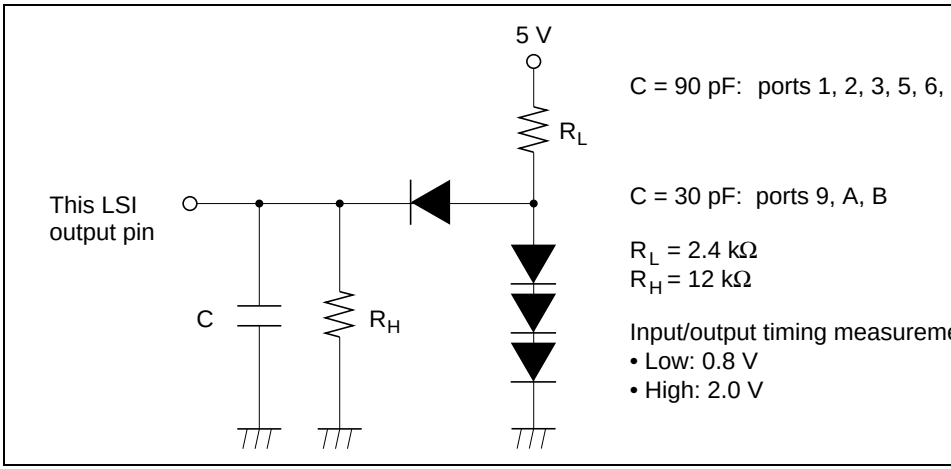


Figure 18.3 Output Load Circuit

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 M
10 MHz, T_a = –20°C to +75°C (regular specifications), T_a = –40°C to +
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
T_a = –20°C to +75°C (regular specifications), T_a = –40°C to +85°C (wi
specifications)

	Condition A			Condition B			Condition C		
	8 MHz			10 MHz			18 MHz		
Item	**Min**	**Typ**	**Max**	**Min**	**Typ**	**Max**	**Min**	**Typ**	**M**
Resolution	10	10	10	10	10	10	10	10	10
Conversion time	—	—	16.8	—	—	13.4	—	—	7.
Analog input capacitance	—	—	20	—	—	20	—	—	20
Permissible signal-source impedance	—	—	10*[1]	—	—	10*[1]	—	—	10
	—	—	5*[2]	—	—	5*[3]	—	—	5*
Nonlinearity error	—	—	±7.5	—	—	±7.5	—	—	±3
Offset error	—	—	±7.5	—	—	±7.5	—	—	±3
Full-scale error	—	—	±7.5	—	—	±7.5	—	—	±3
Quantization error	—	—	±0.5	—	—	±0.5	—	—	±0
Absolute accuracy	—	—	±8.0	—	—	±8.0	—	—	±4

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.

Power supply voltage	V_{CC}	–0.3 to +7.0
Input voltage (except port 7)*[1]	V_{in}	–0.3 to V_{CC} +0.3
Input voltage (port 7)	V_{in}	–0.3 to AV_{CC} +0.3
Analog power supply voltage	AV_{CC}	–0.3 to +7.0
Analog input voltage	V_{AN}	–0.3 to AV_{CC} +0.3
Operating temperature	T_{opr}	Regular specifications: –20 to +75*[2]
		Wide-range specifications: –40 to +85*[2]
Storage temperature	T_{stg}	–55 to +125

Caution: Permanent damage to the chip may result if absolute maximum ratings are e

Notes: 1. 12 V must not be applied to any pin, as this will cause permanent damage to

2. The operating temperature range when programming/erasing flash memory +75°C (regular specifications) or T_a = 0 to +85°C (wide-range specifications)

T_a = 0°C to +85°C (wide-range specifications))

Item		Symbol	Min	Typ	Max	Unit	Test
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} =
			3.5	—	—	V	I_{OH} =
Output low voltage	All output pins	V_{OL}	—	—	0.4	V	I_{OL} =
	Ports 1, 2, 5, B		—	—	1.0	V	I_{OL} =

leakage current (off state)	5, 6, 8, 9, A, B						V_{CC} –0
Input pull-up current	Ports 2, 5	$-I_p$	50	—	300	µA	V_{in} = 0
Input capacitance	NMI, $\overline{RES}$	C_{in}	—	—	50	pF	V_{in} = 0
	All input pins except NMI and $\overline{RES}$		—	—	20		f = 1 M T_a = 2
Current dissipation *2 *4	Normal operation	I_{CC}	—	50	70	mA	f = 18
	Sleep mode		—	35	50		f = 18
	Standby mode*3		—	0.01	5.0	µA	$T_a \leq 5$
			—	—	20.0		50°C
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 w output pins unloaded and the on-chip pull-up transistors in the off state.

3. The values are for $V_{RAM} \leq V_{CC} < 4.5$ 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 +75°C (regular specifications), T_a = 0°C to +85°C (wide-range specifications) (max) higher than the power supply current value in normal operation.

trigger input voltages	$P8_0$ to $P8_1$, PB_0 to PB_3	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}
					0.8		V_{CC} 5.5V
Output high voltage	All output pins	V_{OH}	V_{CC} –0.5	—	—	V	I_{OH} =
			V_{CC} –1.0	—	—	V	I_{OH} =
Output low voltage	All output pins	V_{OL}	—	—	0.4	V	I_{OL} =
	Ports 1, 2, 5, B		—	—	1.0	V	V_{CC} I_{OL} = 4V < I_{OL} =

Three-state leakage current (off state)	Ports 1, 2, 3, 5, 6, 8 to B	$\|I_{TSI}\|$	—	—	1.0	µA	V_{in} = 0 V_{CC} –
Input pull-up current	Ports 2 and 5	$-I_p$	10	—	300	µA	V_{CC} = 5.5 V V_{in} = 0
Input capacitance	NMI, $\overline{RES}$	C_{in}	—	—	50	pF	V_{in} = 0
	All input pins except NMI and $\overline{RES}$		—	—	20	pF	f = 1 T_a = 2
Current dissipation *2 *5	Normal operation	I_{CC} *4	—	15 (3.0 V)	41.5 (5.5 V)	mA	f = 10
	Sleep mode		—	10 (3.0 V)	30.5 (5.5 V)	mA	f = 10
	Standby mode*3		—	0.01	5.0	µA	$T_a \leq 5$
			—	—	20.0		50°C
Analog power supply current	During A/D conversion	AI_{CC}	—	1.3	2.5	mA	AV_{CC}
			—	1.7	2.8		AV_{CC}
	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 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

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 = +75°C (regular specifications), T_a = 0°C to +85°C (wide-range specifications) (max) higher than the current dissipation value in normal operation.

Permissible output low current (total)	Total of 27 pins including ports 1, 2, 5 and B	ΣI_{OL}	—	—	80
	Total of 23 pins, including ports 8, 9, A and B		—	—	75*[2]/ 65*[1]
	Total of all output pins, including the above		—	—	120
Permissible output high current (per pin)	All output pins	I_{OH}	—	—	2.0
Permissible output high current (total)	Total of all output pins	ΣI_{OH}	—	—	40

Notes: To protect chip reliability, do not exceed the output current values in table 18.1

When driving a Darlington pair or LED, always insert a current-limiting resistor 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

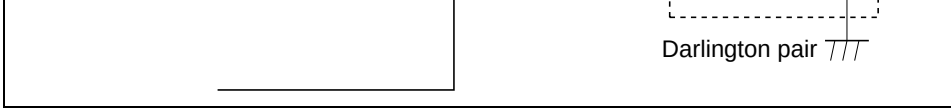


Figure 18.4 Darlington Pair Drive Circuit (Example)

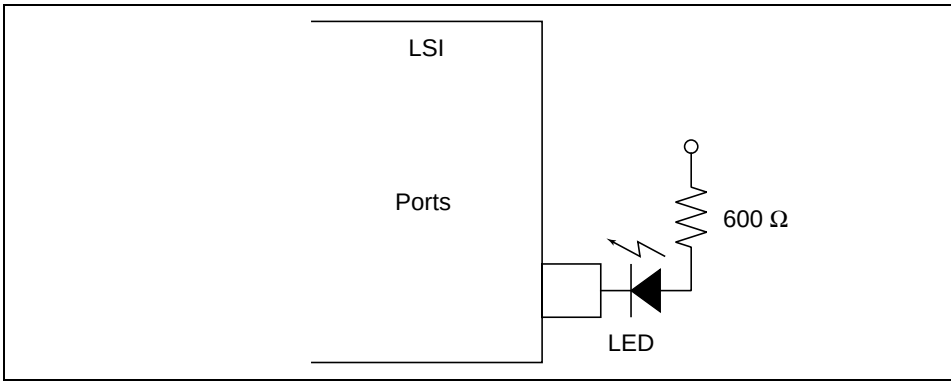


Figure 18.5 LED Drive Circuit (Example)

specifications)

Condition B: V_{CC} = 5.0 V ±10%, AV_{CC} = 5.0 V ±10%, V_{SS} = AV_{SS} = 0 V, ϕ = 2 MHz
T_a = –20°C to +75°C (regular specifications), T_a = –40°C to +85°C (wi
specifications)

		Condition A		Condition B		
		10 MHz		18 MHz		
Item	**Symbol**	**Min**	**Max**	**Min**	**Max**	**Unit**
Clock cycle time	t_{cyc}	100	500	55.5	500	ns
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	—	

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	F
Wait hold time	t_{WTH}	10	—	5	—		

Note: * For condition A, the following times depend on the clock cycle time as shown

$t_{ACC1} = 1.5 \times t_{cyc} - 50$ (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

$t_{ACC1} = 1.5 \times t_{cyc} - 34$ (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)

Item	Symbol	Condition A 10 MHz Min	Condition A 10 MHz Max	Condition B 18 MHz Min	Condition B 18 MHz Max	Unit
$\overline{RES}$ setup time	t_{RESS}	200	—	200	—	ns
$\overline{RES}$ 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
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
Clock oscillator settling time in software standby (crystal)	t_{OSC2}	8	—	7	—	ms

Item			Symbol	Condition A 10 MHz Min	Condition A 10 MHz Max	Condition B 18 MHz Min	Condition B 18 MHz Max	Unit	T C
ITU	Timer output delay time		t_{TOCD}	—	100	—	100	ns	F
	Timer input setup time		t_{TICS}	50	—	50	—		
	Timer clock input setup time		t_{TCKS}	50	—	50	—		F
	Timer clock pulse width	Single edge	t_{TCKWH}	1.5	—	1.5	—	t_{cyc}	
		Both edges	t_{TCKWL}	2.5	—	2.5	—		
SCI	Input clock cycle	Asynchro-nous	t_{Scyc}	4	—	4	—		F
		Synchro-nous		6	—	6	—		
	Input clock rise time		t_{SCKr}	—	1.5	—	1.5		
	Input clock fall time		t_{SCKf}	—	1.5	—	1.5		
	Input clock pulse width		t_{SCKW}	0.4	0.6	0.4	0.6	t_{Scyc}	
	Transmit data delay time		t_{TXD}	—	100	—	100	ns	F
	Receive data setup time (synchronous)		t_{RXS}	100	—	100	—		
	Receive data hold time (synchronous clock input)		t_{RXH}	100	—	100	—		
	Receive data hold time (synchronous clock output)			0	—	0	—		

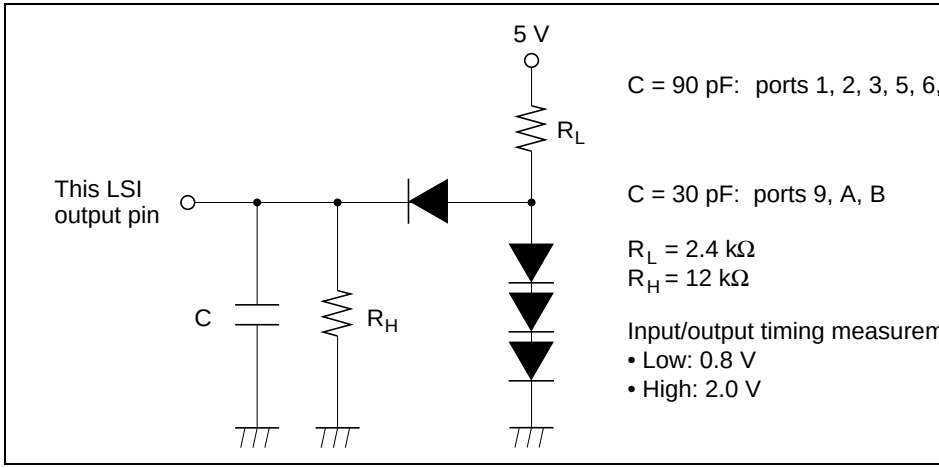


Figure 18.6 Output Load Circuit

Condition B: V_{CC} = 5.0 V ±10%, AV_{CC} = 5.0 V ±10%, V_{SS} = AV_{SS} = 0 V, φ = 2 MHz to
T_a = –20°C to +75°C (regular specifications), T_a = –40°C to +85°C (wid
specifications)

	Condition A			Condition B		
	10 MHz			18 MHz		
Item	**Min**	**Typ**	**Max**	**Min**	**Typ**	**Max**
Resolution	10	10	10	10	10	10
Conversion time	—	—	13.4	—	—	7.5
Analog input capacitance	—	—	20	—	—	20
Permissible signal-source impedance	—	—	5*[1]	—	—	10*[2]
				—	—	5*[3]
Nonlinearity error	—	—	±7.5	—	—	±3.5
Offset error	—	—	±7.5	—	—	±3.5
Full-scale error	—	—	±7.5	—	—	±3.5
Quantization error	—	—	±0.5	—	—	±0.5
Absolute accuracy	—	—	±8.0	—	—	±4.0

Notes: 1. The value is for φ = 10 MHz.
2. The value is for φ ≤ 12 MHz.
3. The value is for φ > 12 MHz.

wide-range specifications)

Item		Symbol	Min	Typ	Max	Unit
Programming time*1 *2 *4		t_P	—	10	200	ms/32 by
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 cor (FLMCR) is set. It does not include the programming verification time.)

3. For the maximum erase time (t_E(max)), the following relationship applies be
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
above formula.

Examples: When z = 5 [ms]: N = 60 times
When z = 10 [ms]: N = 30 times

Programming time*1 *2 *4		t_P	—	10	200	ms/32 by
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 flow erase/erase-verify flowchart.

2. Programming time per 32 bytes (Shows the total period for which the P-bit memory control register (FLMCR) is set. It does not include the programmi verification time.)

3. Block erase time (Shows the total period for which the E-bit in FLMCR is se include the erase verification time.)

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
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 s
  inserted.

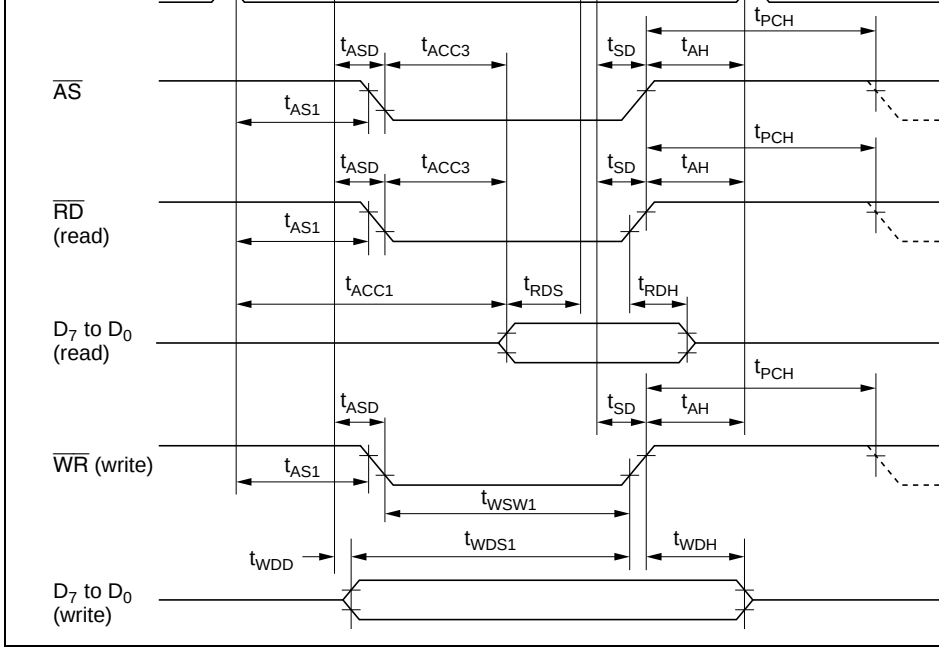


Figure 18.7 Basic Bus Cycle: Two-State Access

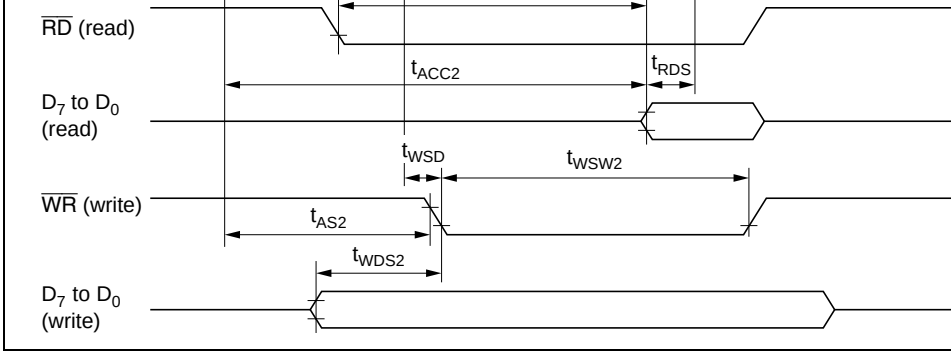


Figure 18.8 Basic Bus Cycle: Three-State Access

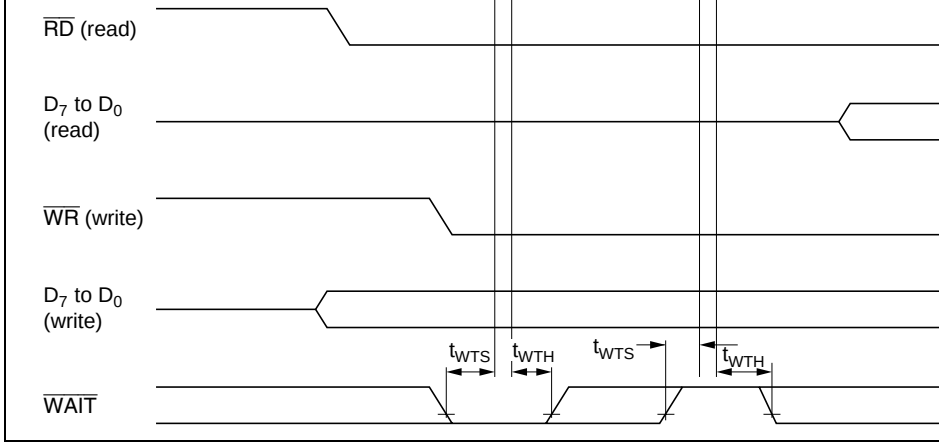


Figure 18.9 Basic Bus Cycle: Three-State Access with One Wait Stat

- Interrupt input timing

  Figure 18.12 shows the interrupt input timing for NMI and $\overline{IRQ_5}$, $\overline{IRQ_4}$, $\overline{IRQ_1}$, and $\overline{I}$

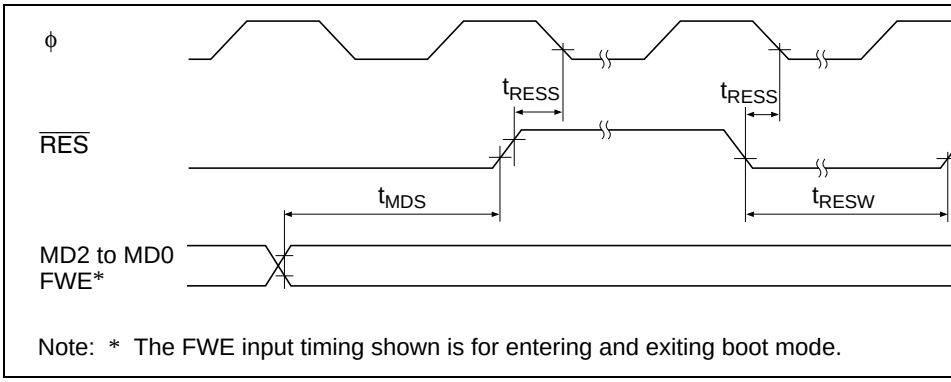


Figure 18.10 Reset Input Timing

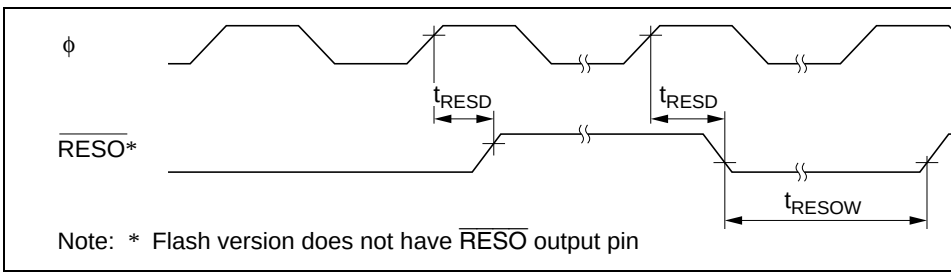


Figure 18.11 Reset Output Timing

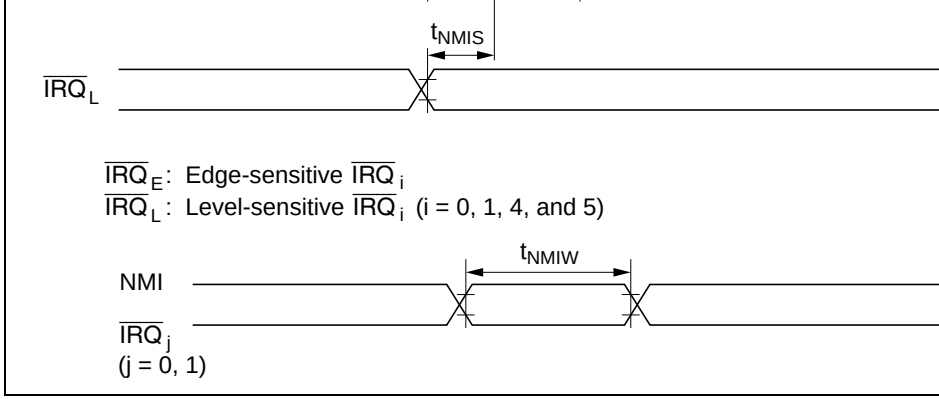


Figure 18.12 Interrupt Input Timing

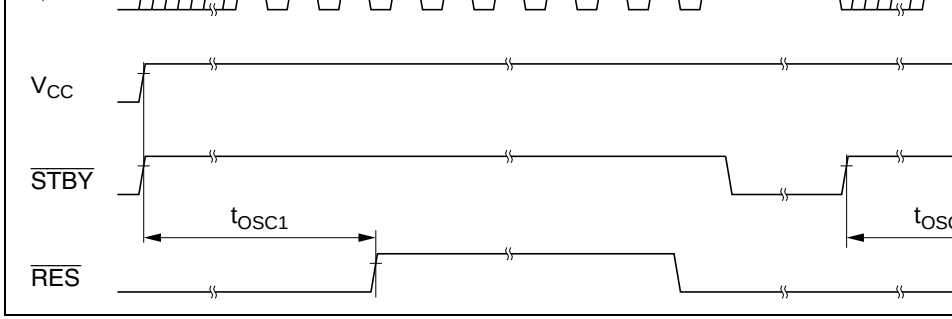


Figure 18.13 Oscillator Settling Timing

18.3.4 TPC and I/O Port Timing

TPC and I/O port timing is shown below.

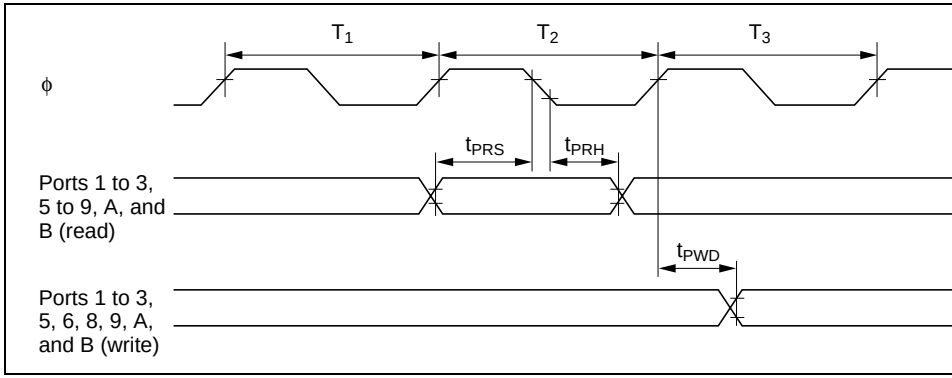
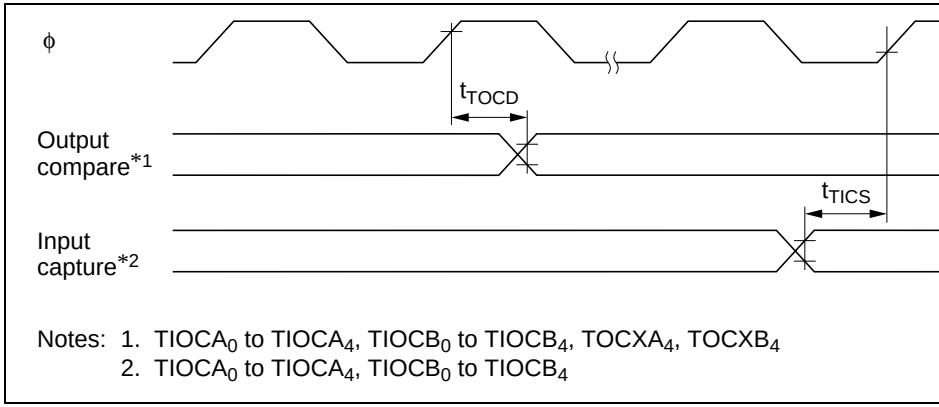


Figure 18.14 TPC and I/O Port Input/Output Timing



Notes: 1. $TIOCA_0$ to $TIOCA_4$, $TIOCB_0$ to $TIOCB_4$, $TOCXA_4$, $TOCXB_4$
2. $TIOCA_0$ to $TIOCA_4$, $TIOCB_0$ to $TIOCB_4$

Figure 18.15 ITU Input/Output Timing

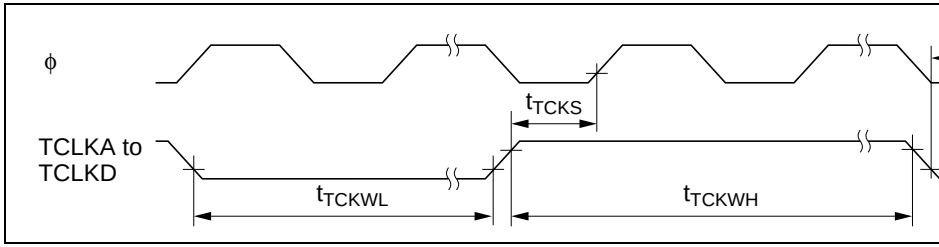


Figure 18.16 ITU External Clock Input Timing

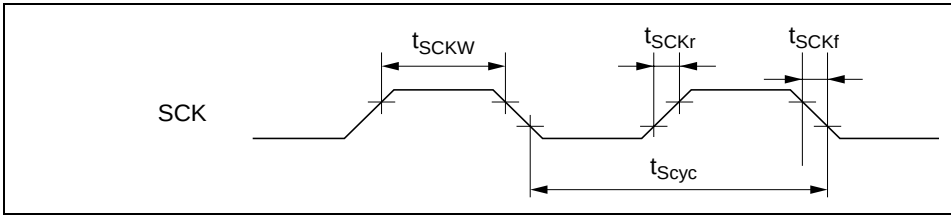


Figure 18.17 SCK Input Clock Timing

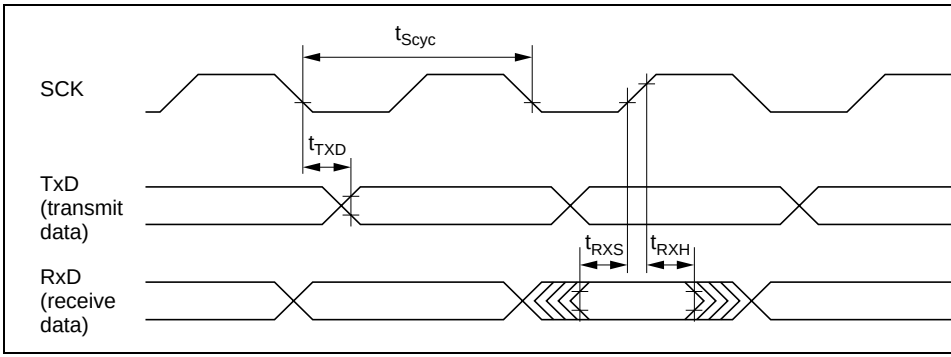


Figure 18.18 SCI Input/Output Timing in Synchronous Mode

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 trans the state on the left to the state on the right
+	Addition of the operands on both sides
–	Subtraction of the operand on the right from the operand on the left
×	Multiplication of the operands on both sides
÷	Division of the operand on the left by the operand on the right
∧	Logical AND of the operands on both sides
•	Logical OR of the operands on both sides
⊕	Exclusive logical OR of the operands on both sides
¬	NOT (logical complement)
( ), < >	Contents of operand

Note: * General registers include 8-bit registers (R0H to R7H and R0L to R7L) and registers (R0 to R7 and E0 to E7).

Δ	Varies depending on conditions, described in notes

Mnemonic	Operand Size	Operation	#xx	Rn	@ERn	@(d, ERn)	@–ERn	@aa	@(d, PC)	@@aa	Implied	Condition Code I	H	N	Z	V	C
MOV.B #xx:8, Rd	B	#xx:8 → Rd8	2									—	—	↕	↕	0	–
MOV.B Rs, Rd	B	Rs8 → Rd8		2								—	—	↕	↕	0	–
MOV.B @ERs, Rd	B	@ERs → Rd8			2							—	—	↕	↕	0	–
MOV.B @(d:16, ERs), Rd	B	@(d:16, ERs) → Rd8				4						—	—	↕	↕	0	–
MOV.B @(d:24, ERs), Rd	B	@(d:24, ERs) → Rd8				8						—	—	↕	↕	0	–
MOV.B @ERs+, Rd	B	@ERs → RD8 ERs32+1 → ERs32					2					—	—	↕	↕	0	–
MOV.B @aa:8, Rd	B	@aa:8 → Rd8						2				—	—	↕	↕	0	–
MOV.B @aa:16, Rd	B	@aa:16 → Rd8						4				—	—	↕	↕	0	–
MOV.B @aa:24, Rd	B	@aa:24 → Rd8						6				—	—	↕	↕	0	–
MOV.B Rs, @ERd	B	Rs8 → @ERd			2							—	—	↕	↕	0	–
MOV.B Rs, @(d:16, ERd)	B	Rd8 → @(d:16, ERd)				4						—	—	↕	↕	0	–
MOV.B Rs, @(d:24, ERd)	B	Rd8 → @(d:24, ERd)				8						—	—	↕	↕	0	–
MOV.B Rs, @–ERd	B	ERd32–1 → ERd32 Rs8 → @ERd					2					—	—	↕	↕	0	–
MOV.B Rs, @aa:8	B	Rs8 → @aa:8						2				—	—	↕	↕	0	–
MOV.B Rs, @aa:16	B	Rs8 → @aa:16						4				—	—	↕	↕	0	–
MOV.B Rs, @aa:24	B	Rs8 → @aa:24						6				—	—	↕	↕	0	–
MOV.W #xx:16, Rd	W	#xx:16 → Rd16	4									—	—	↕	↕	0	–
MOV.W Rs, Rd	W	Rs16 → Rd16		2								—	—	↕	↕	0	–
MOV.W @ERs, Rd	W	@ERs → Rd16			2							—	—	↕	↕	0	–
MOV.W @(d:16, ERs), Rd	W	@(d:16, ERs) → Rd16				4						—	—	↕	↕	0	–
MOV.W @(d:24, ERs), Rd	W	@(d:24, ERs) → Rd16				8						—	—	↕	↕	0	–
MOV.W @ERs+, Rd	W	@ERs → Rd16 ERs32+2 → @ERd32					2					—	—	↕	↕	0	–
MOV.W @aa:16, Rd	W	@aa:16 → Rd16						4				—	—	↕	↕	0	–

MOV.W Rs, @(d:16, ERd)	W	Rs16 → @(d:16, ERd)				4						—	—	↕	↕	0	—
MOV.W Rs, @(d:24, ERd)	W	Rs16 → @(d:24, ERd)				8						—	—	↕	↕	0	—
MOV.W Rs, @–ERd	W	ERd32–2 → ERd32 Rs16 → @ERd					2					—	—	↕	↕	0	—
MOV.W Rs, @aa:16	W	Rs16 → @aa:16						4				—	—	↕	↕	0	—
MOV.W Rs, @aa:24	W	Rs16 → @aa:24						6				—	—	↕	↕	0	—
MOV.L #xx:32, Rd	L	#xx:32 → Rd32	6									—	—	↕	↕	0	—
MOV.L ERs, ERd	L	ERs32 → ERd32		2								—	—	↕	↕	0	—
MOV.L @ERs, ERd	L	@ERs → ERd32			4							—	—	↕	↕	0	—
MOV.L @(d:16, ERs), ERd	L	@(d:16, ERs) → ERd32				6						—	—	↕	↕	0	—
MOV.L @(d:24, ERs), ERd	L	@(d:24, ERs) → ERd32				10						—	—	↕	↕	0	—
MOV.L @ERs+, ERd	L	@ERs → ERd32 ERs32+4 → ERs32					4					—	—	↕	↕	0	—
MOV.L @aa:16, ERd	L	@aa:16 → ERd32						6				—	—	↕	↕	0	—
MOV.L @aa:24, ERd	L	@aa:24 → ERd32						8				—	—	↕	↕	0	—
MOV.L ERs, @ERd	L	ERs32 → @ERd			4							—	—	↕	↕	0	—
MOV.L ERs, @(d:16, ERd)	L	ERs32 → @(d:16, ERd)				6						—	—	↕	↕	0	—
MOV.L ERs, @(d:24, ERd)	L	ERs32 → @(d:24, ERd)				10						—	—	↕	↕	0	—
MOV.L ERs, @–ERd	L	ERd32–4 → ERd32 ERs32 → @ERd					4					—	—	↕	↕	0	—
MOV.L ERs, @aa:16	L	ERs32 → @aa:16						6				—	—	↕	↕	0	—
MOV.L ERs, @aa:24	L	ERs32 → @aa:24						8				—	—	↕	↕	0	—
POP.W Rn	W	@SP → Rn16 SP+2 → SP									2	—	—	↕	↕	0	—
POP.L ERn	L	@SP → ERn32 SP+4 → SP									4	—	—	↕	↕	0	—

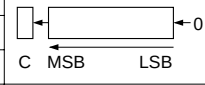
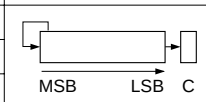
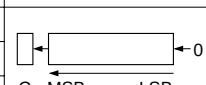
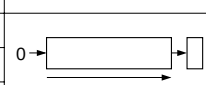
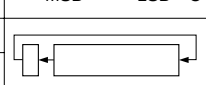
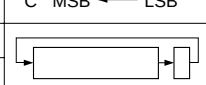
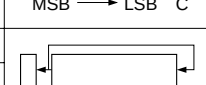
PUSH.L ERn	L	SP–4 → SP ERn32 → @SP									4	—	—	↕	↕	0	–
MOVFPE @aa:16, Rd	B	Cannot be used in the H8/3039 Group						4				Cannot be used in Group					
MOVTPE Rs, @aa:16	B	Cannot be used in the H8/3039 Group						4				Cannot be used in Group					

ADD.B #xx:8, Rd	B	Rd8+#xx:8 → Rd8	2									—	↕	↕	↕	↕	↕
ADD.B Rs, Rd	B	Rd8+Rs8 → Rd8		2								—	↕	↕	↕	↕	↕
ADD.W #xx:16, Rd	W	Rd16+#xx:16 → Rd16	4									—	(1)	↕	↕	↕	↕
ADD.W Rs, Rd	W	Rd16+Rs16 → Rd16		2								—	(1)	↕	↕	↕	↕
ADD.L #xx:32, ERd	L	ERd32+#xx:32 → ERd32	6									—	(2)	↕	↕	↕	↕
ADD.L ERs, ERd	L	ERd32+ERs32 → ERd32		2								—	(2)	↕	↕	↕	↕
ADDX.B #xx:8, Rd	B	Rd8+#xx:8 +C → Rd8	2									—	↕	↕	(3)	↕	↕
ADDX.B Rs, Rd	B	Rd8+Rs8 +C → Rd8		2								—	↕	↕	(3)	↕	↕
ADDS.L #1, ERd	L	ERd32+1 → ERd32		2								—	—	—	—	—	—
ADDS.L #2, ERd	L	ERd32+2 → ERd32		2								—	—	—	—	—	—
ADDS.L #4, ERd	L	ERd32+4 → ERd32		2								—	—	—	—	—	—
INC.B Rd	B	Rd8+1 → Rd8		2								—	—	↕	↕	↕	—
INC.W #1, Rd	W	Rd16+1 → Rd16		2								—	—	↕	↕	↕	—
INC.W #2, Rd	W	Rd16+2 → Rd16		2								—	—	↕	↕	↕	—

DAA Rd	B	Rd8 decimal adjust → Rd8		2								—	*	↕	↕	*	–
SUB.B Rs, Rd	B	Rd8–Rs8 → Rd8		2								—	↕	↕	↕	↕	↕
SUB.W #xx:16, Rd	W	Rd16–#xx:16 → Rd16	4									—	(1)	↕	↕	↕	↕
SUB.W Rs, Rd	W	Rd16–Rs16 → Rd16		2								—	(1)	↕	↕	↕	↕
SUB.L #xx:32, ERd	L	ERd32–#xx:32 → ERd32	6									—	(2)	↕	↕	↕	↕
SUB.L ERs, ERd	L	ERd32–ERs32 → ERd32		2								—	(2)	↕	↕	↕	↕
SUBX.B #xx:8, Rd	B	Rd8–#xx:8–C → Rd8	2									—	↕	↕	(3)	↕	↕
SUBX.B Rs, Rd	B	Rd8–Rs8–C → Rd8		2								—	↕	↕	(3)	↕	↕
SUBS.L #1, ERd	L	ERd32–1 → ERd32		2								—	—	—	—	—	–
SUBS.L #2, ERd	L	ERd32–2 → ERd32		2								—	—	—	—	—	–
SUBS.L #4, ERd	L	ERd32–4 → ERd32		2								—	—	—	—	—	–
DEC.B Rd	B	Rd8–1 → Rd8		2								—	—	↕	↕	↕	–
DEC.W #1, Rd	W	Rd16–1 → Rd16		2								—	—	↕	↕	↕	–
DEC.W #2, Rd	W	Rd16–2 → Rd16		2								—	—	↕	↕	↕	–
DEC.L #1, ERd	L	ERd32–1 → ERd32		2								—	—	↕	↕	↕	–
DEC.L #2, ERd	L	ERd32–2 → ERd32		2								—	—	↕	↕	↕	–
DAS.Rd	B	Rd8 decimal adjust → Rd8		2								—	*	↕	↕	*	–
MULXU. B Rs, Rd	B	Rd8 × Rs8 → Rd16 (unsigned multiplication)		2								—	—	—	—	—	–
MULXU. W Rs, ERd	W	Rd16 × Rs16 → ERd32 (unsigned multiplication)		2								—	—	—	—	—	–
MULXS. B Rs, Rd	B	Rd8 × Rs8 → Rd16 (signed multiplication)		4								—	—	↕	↕	—	–
MULXS. W Rs, ERd	W	Rd16 × Rs16 → ERd32 (signed multiplication)		4								—	—	↕	↕	—	–
DIVXU. B Rs, Rd	B	Rd16 ÷ Rs8 → Rd16 (RdH: remainder, RdL: quotient) (unsigned division)		2								—	—	(6)	(7)	—	–

		Rd: quotient) (unsigned division)															
DIVXS. B Rs, Rd	B	Rd16 ÷ Rs8 → Rd16 (RdH: remainder, RdL: quotient) (signed division)		4								—	—	(8)	(7)	—	—
DIVXS. W Rs, ERd	W	ERd32 ÷ Rs16 → ERd32 (Ed: remainder, Rd: quotient) (signed division)		4								—	—	(8)	(7)	—	—
CMP.B #xx:8, Rd	B	Rd8–#xx:8	2									—	↕	↕	↕	↕	↕
CMP.B Rs, Rd	B	Rd8–Rs8		2								—	↕	↕	↕	↕	↕
CMP.W #xx:16, Rd	W	Rd16–#xx:16	4									—	(1)	↕	↕	↕	↕
CMP.W Rs, Rd	W	Rd16–Rs16		2								—	(1)	↕	↕	↕	↕
CMP.L #xx:32, ERd	L	ERd32–#xx:32	6									—	(2)	↕	↕	↕	↕
CMP.L ERs, ERd	L	ERd32–ERs32		2								—	(2)	↕	↕	↕	↕
NEG.B Rd	B	0–Rd8 → Rd8		2								—	↕	↕	↕	↕	↕
NEG.W Rd	W	0–Rd16 → Rd16		2								—	↕	↕	↕	↕	↕
NEG.L ERd	L	0–ERd32 → ERd32		2								—	↕	↕	↕	↕	↕
EXTU.W Rd	W	0 → (<bits 15 to 8> of Rd16)		2								—	—	0	↕	0	—
EXTU.L ERd	L	0 → (<bits 31 to 16> of Rd32)		2								—	—	0	↕	0	—
EXTS.W Rd	W	(<bit 7> of Rd16) → (<bits 15 to 8> of Rd16)		2								—	—	↕	↕	0	—
EXTS.L ERd	L	(<bit 15> of Rd32) → (<bits 31 to 16> of ERd32)		2								—	—	↕	↕	0	—

AND.B #xx:8, Rd	B	Rd8∧#xx:8 → Rd8	2									—	—	↕	↕	0	–
AND.B Rs, Rd	B	Rd8∧Rs8 → Rd8		2								—	—	↕	↕	0	–
AND.W #xx:16, Rd	W	Rd16∧#xx:16 → Rd16	4									—	—	↕	↕	0	–
AND.W Rs, Rd	W	Rd16∧Rs16 → Rd16		2								—	—	↕	↕	0	–
AND.L #xx:32, ERd	L	ERd32∧#xx:32 → ERd32	6									—	—	↕	↕	0	–
AND.L ERs, ERd	L	ERd32∧ERs32 → ERd32		4								—	—	↕	↕	0	–
OR.B #xx:8, Rd	B	Rd8∨#xx:8 → Rd8	2									—	—	↕	↕	0	–
OR.B Rs, Rd	B	Rd8∨Rs8 → Rd8		2								—	—	↕	↕	0	–
OR.W #xx:16, Rd	W	Rd16∨#xx:16 → Rd16	4									—	—	↕	↕	0	–
OR.W Rs, Rd	W	Rd16∨Rs16 → Rd16		2								—	—	↕	↕	0	–
OR.L #xx:32, ERd	L	ERd32∨#xx:32 → ERd32	6									—	—	↕	↕	0	–
OR.L ERs, ERd	L	ERd32∨ERs32 → ERd32		4								—	—	↕	↕	0	–
XOR.B #xx:8, Rd	B	Rd8⊕#xx:8 → Rd8	2									—	—	↕	↕	0	–
XOR.B Rs, Rd	B	Rd8⊕Rs8 → Rd8		2								—	—	↕	↕	0	–
XOR.W #xx:16, Rd	W	Rd16⊕#xx:16 → Rd16	4									—	—	↕	↕	0	–
XOR.W Rs, Rd	W	Rd16⊕Rs16 → Rd16		2								—	—	↕	↕	0	–
XOR.L #xx:32, ERd	L	ERd32⊕#xx:32 → ERd32	6									—	—	↕	↕	0	–
XOR.L ERs, ERd	L	ERd32⊕ERs32 → ERd32		4								—	—	↕	↕	0	–
NOT.B Rd	B	¬ Rd8 → Rd8		2								—	—	↕	↕	0	–
NOT.W Rd	W	¬ Rd16 → Rd16		2								—	—	↕	↕	0	–
NOT.L ERd	L	¬ Rd32 → Rd32		2								—	—	↕	↕	0	–

SHAL.B Rd	B			2								—	—	↕	↕	↕	↕		
SHAL.W Rd	W			2								—	—	↕	↕	↕	↕		
SHAL.L ERd	L			2								—	—	↕	↕	↕	↕		
SHAR.B Rd	B			2								—	—	↕	↕	0	↕		
SHAR.W Rd	W			2								—	—	↕	↕	0	↕		
SHAR.L ERd	L			2								—	—	↕	↕	0	↕		
SHLL.B Rd	B			2								—	—	↕	↕	0	↕		
SHLL.W Rd	W			2								—	—	↕	↕	0	↕		
SHLL.L ERd	L			2								—	—	↕	↕	0	↕		
SHLR.B Rd	B			2								—	—	↕	↕	0	↕		
SHLR.W Rd	W			2								—	—	↕	↕	0	↕		
SHLR.L ERd	L			2								—	—	↕	↕	0	↕		
ROTXL.B Rd	B			2								—	—	↕	↕	0	↕		
ROTXL.W Rd	W			2								—	—	↕	↕	0	↕		
ROTXL.L ERd	L			2								—	—	↕	↕	0	↕		
ROTXR.B Rd	B			2								—	—	↕	↕	0	↕		
ROTXR.W Rd	W			2								—	—	↕	↕	0	↕		
ROTXR.L ERd	L			2								—	—	↕	↕	0	↕		
ROTL.B Rd	B			2								—	—	↕	↕	0	↕		
ROTL.W Rd	W			2								—	—	↕	↕	0	↕		
ROTL.L ERd	L			2								—	—	↕	↕	0	↕		
ROTR.B Rd	B			2								—	—	↕	↕	0	↕		
ROTR.W Rd	W			2								—	—	↕	↕	0	↕		
ROTR.L ERd	L			2								—	—	↕	↕	0	↕		

BSET #xx:3, Rd	B	(#xx:3 of Rd8) ← 1		2								—	—	—	—	—	–
BSET #xx:3, @ERd	B	(#xx:3 of @ERd) ← 1			4							—	—	—	—	—	–
BSET #xx:3, @aa:8	B	(#xx:3 of @aa:8) ← 1						4				—	—	—	—	—	–
BSET Rn, Rd	B	(Rn8 of Rd8) ← 1		2								—	—	—	—	—	–
BSET Rn, @ERd	B	(Rn8 of @ERd) ← 1			4							—	—	—	—	—	–
BSET Rn, @aa:8	B	(Rn8 of @aa:8) ← 1						4				—	—	—	—	—	–
BCLR #xx:3, Rd	B	(#xx:3 of Rd8) ← 0		2								—	—	—	—	—	–
BCLR #xx:3, @ERd	B	(#xx:3 of @ERd) ← 0			4							—	—	—	—	—	–
BCLR #xx:3, @aa:8	B	(#xx:3 of @aa:8) ← 0						4				—	—	—	—	—	–
BCLR Rn, Rd	B	(Rn8 of Rd8) ← 0		2								—	—	—	—	—	–
BCLR Rn, @ERd	B	(Rn8 of @ERd) ← 0			4							—	—	—	—	—	–
BCLR Rn, @aa:8	B	(Rn8 of @aa:8) ← 0						4				—	—	—	—	—	–
BNOT #xx:3, Rd	B	(#xx:3 of Rd8) ← ¬ (#xx:3 of Rd8)		2								—	—	—	—	—	–
BNOT #xx:3, @ERd	B	(#xx:3 of @ERd) ← ¬ (#xx:3 of @ERd)			4							—	—	—	—	—	–
BNOT #xx:3, @aa:8	B	(#xx:3 of @aa:8) ← ¬ (#xx:3 of @aa:8)						4				—	—	—	—	—	–
BNOT Rn, Rd	B	(Rn8 of Rd8) ← ¬ (Rn8 of Rd8)		2								—	—	—	—	—	–
BNOT Rn, @ERd	B	(Rn8 of @ERd) ← ¬ (Rn8 of @ERd)			4							—	—	—	—	—	–
BNOT Rn, @aa:8	B	(Rn8 of @aa:8) ← ¬ (Rn8 of @aa:8)						4				—	—	—	—	—	–
BTST #xx:3, Rd	B	¬ (#xx:3 of Rd8) → Z		2								—	—	—	↕	—	–
BTST #xx:3, @ERd	B	¬ (#xx:3 of @ERd) → Z			4							—	—	—	↕	—	–
BTST #xx:3, @aa:8	B	¬ (#xx:3 of @aa:8) → Z						4				—	—	—	↕	—	–
BTST Rn, Rd	B	¬ (Rn8 of @Rd8) → Z		2								—	—	—	↕	—	–
BTST Rn, @ERd	B	¬ (Rn8 of @ERd) → Z			4							—	—	—	↕	—	–
BTST Rn, @aa:8	B	¬ (Rn8 of @aa:8) → Z						4				—	—	—	↕	—	–
BLD #xx:3, Rd	B	(#xx:3 of Rd8) → C		2								—	—	—	—	—	↕

BILD #xx:3, Rd	B	¬ (#xx:3 of Rd8) → C		2								—	—	—	—	—	↕
BILD #xx:3, @ERd	B	¬ (#xx:3 of @ERd) → C			4							—	—	—	—	—	↕
BILD #xx:3, @aa:8	B	¬ (#xx:3 of @aa:8) → C						4				—	—	—	—	—	↕
BST #xx:3, Rd	B	C → (#xx:3 of Rd8)		2								—	—	—	—	—	—
BST #xx:3, @ERd	B	C → (#xx:3 of @ERd24)			4							—	—	—	—	—	—
BST #xx:3, @aa:8	B	C → (#xx:3 of @aa:8)						4				—	—	—	—	—	—
BIST #xx:3, Rd	B	¬ C → (#xx:3 of Rd8)		2								—	—	—	—	—	—
BIST #xx:3, @ERd	B	¬ C → (#xx:3 of @ERd24)			4							—	—	—	—	—	—
BIST #xx:3, @aa:8	B	¬ C → (#xx:3 of @aa:8)						4				—	—	—	—	—	—
BAND #xx:3, Rd	B	C∧(#xx:3 of Rd8) → C		2								—	—	—	—	—	↕
BAND #xx:3, @ERd	B	C∧(#xx:3 of @ERd24) → C			4							—	—	—	—	—	↕
BAND #xx:3, @aa:8	B	C∧(#xx:3 of @aa:8) → C						4				—	—	—	—	—	↕
BIAND #xx:3, Rd	B	C∧ ¬ (#xx:3 of Rd8) → C		2								—	—	—	—	—	↕
BIAND #xx:3, @ERd	B	C∧ ¬ (#xx:3 of @ERd24) → C			4							—	—	—	—	—	↕
BIAND #xx:3, @aa:8	B	C∧ ¬ (#xx:3 of @aa:8) → C						4				—	—	—	—	—	↕
BOR #xx:3, Rd	B	C∨(#xx:3 of Rd8) → C		2								—	—	—	—	—	↕
BOR #xx:3, @ERd	B	C∨(#xx:3 of @ERd24) → C			4							—	—	—	—	—	↕
BOR #xx:3, @aa:8	B	C∨(#xx:3 of @aa:8) → C						4				—	—	—	—	—	↕
BIOR #xx:3, Rd	B	C∨ ¬ (#xx:3 of Rd8) → C		2								—	—	—	—	—	↕
BIOR #xx:3, @ERd	B	C∨ ¬ (#xx:3 of @ERd24) → C			4							—	—	—	—	—	↕
BIOR #xx:3, @aa:8	B	C∨ ¬ (#xx:3 of @aa:8) → C						4				—	—	—	—	—	↕
BXOR #xx:3, Rd	B	C ⊕ (#xx:3 of Rd8) → C		2								—	—	—	—	—	↕
BXOR #xx:3, @ERd	B	C ⊕(#xx:3 of @ERd24)→ C			4							—	—	—	—	—	↕
BXOR #xx:3, @aa:8	B	C ⊕(#xx:3 of @aa:8)→ C						4				—	—	—	—	—	↕
BIXOR #xx:3, Rd	B	C ⊕ ¬ (#xx:3 of Rd8) → C		2								—	—	—	—	—	↕
BIXOR #xx:3, @ERd	B	C ⊕ ¬ (#xx:3 of @ERd24) → C			4							—	—	—	—	—	↕
BIXOR #xx:3, @aa:8	B	C ⊕ ¬ (#xx:3 of @aa:8)→ C						4				—	—	—	—	—	↕

BRA d:8 (BT d:8)	—	If condition is true then PC ← PC+d else next;	Always							2			—	—	—	—	—	-
BRA d:16 (BT d:16)	—									4			—	—	—	—	—	-
BRN d:8 (BF d:8)	—		Never							2			—	—	—	—	—	-
BRN d:16 (BF d:16)	—									4			—	—	—	—	—	-
BHI d:8	—		C ∨ Z = 0							2			—	—	—	—	—	-
BHI d:16	—									4			—	—	—	—	—	-
BLS d:8	—		C ∨ Z = 1							2			—	—	—	—	—	-
BLS d:16	—									4			—	—	—	—	—	-
BCC d:8 (BHS d:8)	—		C = 0							2			—	—	—	—	—	-
BCC d:16 (BHS d:16)	—									4			—	—	—	—	—	-
BCS d:8 (BLO d:8)	—		C = 1							2			—	—	—	—	—	-
BCS d:16 (BLO d:16)	—									4			—	—	—	—	—	-
BNE d:8	—		Z = 0							2			—	—	—	—	—	-
BNE d:16	—									4			—	—	—	—	—	-
BEQ d:8	—		Z = 1							2			—	—	—	—	—	-
BEQ d:16	—									4			—	—	—	—	—	-
BVC d:8	—		V = 0							2			—	—	—	—	—	-
BVC d:16	—									4			—	—	—	—	—	-
BVS d:8	—		V = 1							2			—	—	—	—	—	-
BVS d:16	—									4			—	—	—	—	—	-
BPL d:8	—		N = 0							2			—	—	—	—	—	-
BPL d:16	—									4			—	—	—	—	—	-
BMI d:8	—		N = 1							2			—	—	—	—	—	-
BMI d:16	—									4			—	—	—	—	—	-
BGE d:8	—		N ⊕ V = 0							2			—	—	—	—	—	-
BGE d:16	—									4			—	—	—	—	—	-
BLT d:8	—		N ⊕ V = 1							2			—	—	—	—	—	-
BLT d:16	—									4			—	—	—	—	—	-
BGT d:8	—		Z ∨ (N ⊕ V) = 0							2			—	—	—	—	—	-
BGT d:16	—									4			—	—	—	—	—	-

		PC ← PC+d else next;																
JMP @ERn	—	PC ← ERn			2						—	—	—	—	—	—		
JMP @aa:24	—	PC ← aa:24						4			—	—	—	—	—	—		
JMP @@aa:8	—	PC ← @aa:8							2		—	—	—	—	—	—		
BSR d:8	—	PC → @–SP PC ← PC+d:8						2			—	—	—	—	—	—		
BSR d:16	—	PC → @–SP PC ← PC+d:16						4			—	—	—	—	—	—		
JSR @ERn	—	PC → @–SP PC ← @ERn			2						—	—	—	—	—	—		
JSR @aa:24	—	PC → @–SP PC ← @aa:24						4			—	—	—	—	—	—		
JSR @@aa:8	—	PC → @–SP PC ← @aa:8							2		—	—	—	—	—	—		
RTS	—	PC ← @SP+								2	—	—	—	—	—	—		

TRAPA #x:2	—	PC → @–SP CCR → @–SP <vector> → PC									2	1	—	—	—	—	
RTE	—	CCR ← @SP+ PC ← @SP+										↕	↕	↕	↕	↕	
SLEEP	—	Transition to power-down state										—	—	—	—	—	
LDC #xx:8, CCR	B	#xx:8 → CCR	2									↕	↕	↕	↕	↕	
LDC Rs, CCR	B	Rs8 → CCR		2								↕	↕	↕	↕	↕	
LDC @ERs, CCR	W	@ERs → CCR			4							↕	↕	↕	↕	↕	
LDC @(d:16, ERs), CCR	W	@(d:16, ERs) → CCR				6						↕	↕	↕	↕	↕	
LDC @(d:24, ERs), CCR	W	@(d:24, ERs) → CCR				10						↕	↕	↕	↕	↕	
LDC @ERs+, CCR	W	@ERs → CCR ERs32+2 → ERs32					4					↕	↕	↕	↕	↕	
LDC @aa:16, CCR	W	@aa:16 → CCR						6				↕	↕	↕	↕	↕	
LDC @aa:24, CCR	W	@aa:24 → CCR						8				↕	↕	↕	↕	↕	
STC CCR, Rd	B	CCR → Rd8		2								—	—	—	—	—	
STC CCR, @ERd	W	CCR → @ERd			4							—	—	—	—	—	
STC CCR, @(d:16, ERd)	W	CCR → @(d:16, ERd)				6						—	—	—	—	—	
STC CCR, @(d:24, ERd)	W	CCR → @(d:24, ERd)				10						—	—	—	—	—	
STC CCR, @–ERd	W	ERd32–2 → ERd32 CCR → @ERd					4					—	—	—	—	—	
STC CCR, @aa:16	W	CCR → @aa:16						6				—	—	—	—	—	
STC CCR, @aa:24	W	CCR → @aa:24						8				—	—	—	—	—	
ANDC #xx:8, CCR	B	CCR∧#xx:8 → CCR	2									↕	↕	↕	↕	↕	
ORC #xx:8, CCR	B	CCR∨#xx:8 → CCR	2									↕	↕	↕	↕	↕	
XORC #xx:8, CCR	B	CCR⊕#xx:8 → CCR	2									↕	↕	↕	↕	↕	
NOP	—	PC ← PC+2									2	—	—	—	—	—	

EEPMOV. B	—	if R4L ≠ 0 then repeat @R5 → @R6 R5+1 → R5 R6+1 → R6 R4L–1 → R4L until R4L=0 else next									4	—	—	—	—	—	—
EEPMOV. W	—	if R4 ≠ 0 then repeat @R5 → @R6 R5+1 → R5 R6+1 → R6 R4–1 → R4 until R4=0 else next									4	—	—	—	—	—	—

Notes: 1. The number of states is the number of states required for execution when th
instruction and its operands are located in on-chip memory. For other cases
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 previou

(5) The number of states required for execution of an instruction that transfers d
synchronization with the E clock is variable.

(6) Set to 1 when the divisor is negative; otherwise cleared to 0.

(7) Set to 1 when the divisor is zero; otherwise cleared to 0.

(8) Set to 1 when the quotient is negative; otherwise cleared to 0.

Instruction code:

1st byte		2nd byte	
AH	AL	BH	BL

Instruction when most significant bit of BH is 0.

Instruction when most significant bit of BH is 1.

AH \ AL	0	1	2	3	4	5	6	7	8	9	A	B	C	D
0	NOP	Table A.2 (2)	STC	LDC	ORC	XORC	ANDC	LDC	ADD		Table A.2 (2)	Table A.2 (2)	MOV	
1	Table A.2 (2)	Table A.2 (2)	Table A.2 (2)	Table A.2 (2)	OR.B	XOR.B	AND.B	Table A.2 (2)	SUB		Table A.2 (2)	Table A.2 (2)	CMP	
2	MOV.B													
3														
4	BRA	BRN	BHI	BLS	BCC	BCS	BNE	BEQ	BVC	BVS	BPL	BMI	BGE	BLT
5	MULXU	DIVXU	MULXU	DIVXU	RTS	BSR	RTE	TRAPA	Table A.2 (2)	JMP			BSR	
6	BSET	BNOT	BCLR	BTST	OR	XOR	AND	BST / BIST	MOV					
7					BOR / BIOR	BXOR / BIXOR	BAND / BIAND	BLD / BILD	MOV	Table A.2 (2)	Table A.2 (2)	EEPMOV	Tabl (	
8	ADD													
9	ADDX													
A	CMP													
B	SUBX													
C	OR													
D	XOR													
E	AND													
F	MOV													

Instruction code:

1st byte		2nd byte	
AH	AL	BH	BL

AH AL \ BH	0	1	2	3	4	5	6	7	8	9	A	B	C
01	MOV				LDC/STC				SLEEP				Table A.2 (3)
0A	INC								ADD				
0B	ADDS					INC		INC	ADDS				
0F	DAA								MOV				
10	SHLL			SHLL					SHAL			SHAL	
11	SHLR			SHLR					SHAR			SHAR	
12	ROTXL			ROTXL					ROTL			ROTL	
13	ROTXR			ROTXR					ROTR			ROTR	
17	NOT			NOT		EXTU		EXTU	NEG			NEG	
1A	DEC								SUB				
1B	SUBS					DEC		DEC	SUB				
1F	DAS								CMP				
58	BRA	BRN	BHI	BLS	BCC	BCS	BNE	BEQ	BVC	BVS	BPL	BMI	BGE
79	MOV	ADD	CMP	SUB	OR	XOR	AND						
7A	MOV	ADD	CMP	SUB	OR	XOR	AND						

Instruction code:

1st byte		2nd byte		3rd byte		4th byte	
AH	AL	BH	BL	CH	CL	DH	DL

Instruction when most signi

Instruction when most signi

CL / AH / ALBH / BLCH	0	1	2	3	4	5	6	7	8	9	A	B	C
01406										LDC / STC		LDC / STC	
01C05	MULXS		MULXS										
01D05		DIVXS		DIVXS									
01F06					OR	XOR	AND						
7Cr06*1				BTST									
7Cr07*1				BTST	BOR / BIOR	BXOR / BIXOR	BAND / BIAND	BLD / BILD					
7Dr06*1	BSET	BNOT	BCLR					BST / BIST					
7Dr07*1	BSET	BNOT	BCLR										
7Eaa6*2				BTST									
7Eaa7*2				BTST	BOR / BIOR	BXOR / BIXOR	BAND / BIAND	BLD / BILD					
7Faa6*2	BSET	BNOT	BCLR					BST / BIST					
7Faa7*2	BSET	BNOT	BCLR										

Notes: 1. r is the register designation field.
2. aa is the absolute address field.

RENESAS

Examples of Calculation of Number of States Required for Execution

Examples: Advanced mode, stack located in external address space, on-chip supportin accessed with 8-bit bus width, external devices accessed in three states with one wait st 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 = $2 \times 4 + 2 \times 4 + 2 \times 4 = 24$

Instruction fetch	S_I	2	3	3	4	3 + 2m	2
Branch address read	S_J						
Stack operation	S_K						
Byte data access	S_L		3		2	3 + m	
Word data access	S_M		6		4	6 + 2m	
Internal operation	S_N	1	1	1	1	1	1

Legend:

m: Number of wait states inserted in external device access

	ADD.L #xx:32, ERd	3		
	ADD.L ERs, ERd	1		
ADDS	ADDS #1/2/4, ERd	1		
ADDX	ADDX #xx:8, Rd	1		
	ADDX Rs, Rd	1		
AND	AND.B #xx:8, Rd	1		
	AND.B Rs, Rd	1		
	AND.W #xx:16, Rd	2		
	AND.W Rs, Rd	1		
	AND.L #xx:32, ERd	3		
	AND.L ERs, ERd	2		
ANDC	ANDC #xx:8, CCR	1		
BAND	BAND #xx:3, Rd	1		
	BAND #xx:3, @ERd	2		1
	BAND #xx:3, @aa:8	2		1
Bcc	BRA d:8 (BT d:8)	2		
	BRN d:8 (BF d:8)	2		
	BHI d:8	2		
	BLS d:8	2		
	BCC d:8 (BHS d:8)	2		
	BCS d:8 (BLO d:8)	2		
	BNE d:8	2		
	BEQ d:8	2		
	BVC d:8	2		
	BVS d:8	2		
	BPL d:8	2		
	BMI d:8	2		
	BGE d:8	2		
	BLT d:8	2		
	BGT d:8	2		
	BLE d:8	2		

	BNE d:16	2		
	BEQ d:16	2		
	BVC d:16	2		
	BVS d:16	2		
	BPL d:16	2		
	BMI d:16	2		
	BGE d:16	2		
	BLT d:16	2		
	BGT d:16	2		
	BLE d:16	2		
BCLR	BCLR #xx:3, Rd	1		
	BCLR #xx:3, @ERd	2		2
	BCLR #xx:3, @aa:8	2		2
	BCLR Rn, Rd	1		
	BCLR Rn, @ERd	2		2
	BCLR Rn, @aa:8	2		2
BIAND	BIAND #xx:3, Rd	1		
	BIAND #xx:3, @ERd	2		1
	BIAND #xx:3, @aa:8	2		1
BILD	BILD #xx:3, Rd	1		
	BILD #xx:3, @ERd	2		1
	BILD #xx:3, @aa:8	2		1
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

	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	
		Advanced	2	2	
BST	BST #xx:3, Rd		1		
	BST #xx:3, @ERd		2		2
	BST #xx:3, @aa:8		2		2
BTST	BTST #xx:3, Rd		1		
	BTST #xx:3, @ERd		2		1
	BTST #xx:3, @aa:8		2		1
	BTST Rn, Rd		1		
	BTST Rn, @ERd		2		1
	BTST Rn, @aa:8		2		1
BXOR	BXOR #xx:3, Rd		1		
	BXOR #xx:3, @ERd		2		1
	BXOR #xx:3, @aa:8		2		1

DAA	DAA Rd		1			
DAS	DAS Rd		1			
DEC	DEC.B Rd		1			
	DEC.W #1/2, Rd		1			
	DEC.L #1/2, ERd		1			
DIVXS	DIVXS.B Rs, Rd		2			
	DIVXS.W Rs, ERd		2			
DIVXU	DIVXU.B Rs, Rd		1			
	DIVXU.W Rs, ERd		1			
EEPMOV	EEPMOV.B		2			2n +2*1
	EEPMOV.W		2			2n +2*1
EXTS	EXTS.W Rd		1			
	EXTS.L ERd		1			
EXTU	EXTU.W Rd		1			
	EXTU.L ERd		1			
INC	INC.B Rd		1			
	INC.W #1/2, Rd		1			
	INC.L #1/2, ERd		1			
JMP	JMP @ERn		2			
	JMP @aa:24		2			
	JMP @@aa:8	Normal	2	1		
		Advanced	2	2		
JSR	JSR @ERn	Normal	2		1	
		Advanced	2		2	
	JSR @aa:24	Normal	2		1	
		Advanced	2		2	
	JSR @@aa:8	Normal	2	1	1	
		Advanced	2	2	2	

	LDC @aa:16, CCR	3			1
	LDC @aa:24, CCR	4			1
MOV	MOV.B #xx:8, Rd	1			
	MOV.B Rs, Rd	1			
	MOV.B @ERs, Rd	1		1	
	MOV.B @(d:16, ERs), Rd	2		1	
	MOV.B @(d:24, ERs), Rd	4		1	
	MOV.B @ERs+, Rd	1		1	
	MOV.B @aa:8, Rd	1		1	
	MOV.B @aa:16, Rd	2		1	
	MOV.B @aa:24, Rd	3		1	
	MOV.B Rs, @ERd	1		1	
	MOV.B Rs, @(d:16, ERd)	2		1	
	MOV.B Rs, @(d:24, ERd)	4		1	
	MOV.B Rs, @–ERd	1		1	
	MOV.B Rs, @aa:8	1		1	
	MOV.B Rs, @aa:16	2		1	
	MOV.B Rs, @aa:24	3		1	
	MOV.W #xx:16, Rd	2			
	MOV.W Rs, Rd	1			
	MOV.W @ERs, Rd	1			1
	MOV.W @(d:16, ERs), Rd	2			1
	MOV.W @(d:24, ERs), Rd	4			1
	MOV.W @ERs+, Rd	1			1
	MOV.W @aa:16, Rd	2			1
	MOV.W @aa:24, Rd	3			1
	MOV.W Rs, @ERd	1			1
	MOV.W Rs, @(d:16, ERd)	2			1
	MOV.W Rs, @(d:24, ERd)	4			1
	MOV.W Rs, @–ERd	1			1
	MOV.W Rs, @aa:16	2			1

	MOV.L @ERs+, ERd	2		2
	MOV.L @aa:16, ERd	3		2
	MOV.L @aa:24, ERd	4		2
	MOV.L ERs, @ERd	2		2
	MOV.L ERs, @(d:16, ERd)	3		2
	MOV.L ERs, @(d:24, ERd)	5		2
	MOV.L ERs, @–ERd	2		2
	MOV.L ERs, @aa:16	3		2
	MOV.L ERs, @aa:24	4		2
MOVFPE	MOVFPE @aa:16, Rd*[2]	2	1	
MOVTPE	MOVTPE Rs, @aa:16*[2]	2	1	
MULXS	MULXS.B Rs, Rd	2		
	MULXS.W Rs, ERd	2		
MULXU	MULXU.B Rs, Rd	1		
	MULXU.W Rs, ERd	1		
NEG	NEG.B Rd	1		
	NEG.W Rd	1		
	NEG.L ERd	1		
NOP	NOP	1		
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		

	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
RTS	RTS	Normal	2	1
		Advanced	2	2
SHAL	SHAL.B Rd		1	
	SHAL.W Rd		1	
	SHAL.L ERd		1	
SHAR	SHAR.B Rd		1	
	SHAR.W Rd		1	
	SHAR.L ERd		1	
SHLL	SHLL.B Rd		1	
	SHLL.W Rd		1	
	SHLL.L ERd		1	
SHLR	SHLR.B Rd		1	
	SHLR.W Rd		1	
	SHLR.L ERd		1	
SLEEP	SLEEP		1	

	STC CCR, @aa:24		4			1
SUB	SUB.B Rs, Rd		1			
	SUB.W #xx:16, Rd		2			
	SUB.W Rs, Rd		1			
	SUB.L #xx:32, ERd		3			
	SUB.L ERs, ERd		1			
SUBS	SUBS #1/2/4, ERd		1			
SUBX	SUBX #xx:8, Rd		1			
	SUBX Rs, Rd		1			
TRAPA	TRAPA #x:2	Normal	2	1	2	
		Advanced	2	2	2	
XOR	XOR.B #xx:8, Rd		1			
	XOR.B Rs, Rd		1			
	XOR.W #xx:16, Rd		2			
	XOR.W Rs, Rd		1			
	XOR.L #xx:32, ERd		3			
	XOR.L ERs, ERd		2			
XORC	XORC #xx:8, CCR		1			

Notes: 1. n is the value set in register R4L or R4. The source and destination are acc
times each.

2. Not used with this LSI.

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	—	—	—	—	—	—	—	—	—

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
H'5E	MSTCR	8	PSTOP	—	MSTOP5	MSTOP4	MSTOP3	—	—	MSTOP0
H'5F	—		—	—	—	—	—	—	—	—

H'67	TSR0	8	—	—	—	—	—	OVF	IMFB	IMFA
H'68	TCNT0H	16								
H'69	TCNT0L									
H'6A	GRA0H	16								
H'6B	GRA0L									
H'6C	GRB0H	16								
H'6D	GRB0L									
H'6E	TCR1	8	—	CCLR1	CCLR0	CKEG1	CKEG0	TPSC2	TPSC1	TPSC0
H'6F	TIOR1	8	—	IOB2	IOB1	IOB0	—	IOA2	IOA1	IOA0
H'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
H'79	TIOR2	8	—	IOB2	IOB1	IOB0	—	IOA2	IOA1	IOA0
H'7A	TIER2	8	—	—	—	—	—	OVIE	IMIEB	IMIEA
H'7B	TSR2	8	—	—	—	—	—	OVF	IMFB	IMFA
H'7C	TCNT2H	16								
H'7D	TCNT2L									
H'7E	GRA2H	16								
H'7F	GRA2L									
H'80	GRB2H	16								
H'81	GRB2L									

H'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
H'91	TOCR	8	—	—	—	XTGD	—	—	OLS4	OLS3
H'92	TCR4	8	—	CCLR1	CCLR0	CKEG1	CKEG0	TPSC2	TPSC1	TPSC0
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									

		8	NDR7	NDR6	NDR5	NDR4	—	—	—	—
H'A6	NDRB*[1]	8	—	—	—	—	—	—	—	—
		8	—	—	—	—	NDR11	NDR10	NDR9	NDR8
H'A7	NDRA*[1]	8	—	—	—	—	—	—	—	—
		8	—	—	—	—	NDR3	NDR2	NDR1	NDR0
H'A8	TCSR*[2]	8	OVF	WT/$\overline{IT}$	TME	—	—	CKS2	CKS1	CKS0
H'A9	TCNT*[2]	8								
H'AA	—		—	—	—	—	—	—	—	—
H'AB	RSTCSR*[2]	8	WRST	RSTOE	—	—	—	—	—	—
H'AC	—		—	—	—	—	—	—	—	—
H'AD	—		—	—	—	—	—	—	—	—
H'AE	—		—	—	—	—	—	—	—	—
H'AF	—		—	—	—	—	—	—	—	—
H'B0	SMR	8	C/$\overline{A}$	CHR	PE	O/$\overline{E}$	STOP	MP	CKS1	CKS0
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
H'B7										
H'B8	SMR	8	C/$\overline{A}$	CHR	PE	O/$\overline{E}$	STOP	MP	CKS1	CKS0
H'B9	BRR	8	—	—	—	—	—	—	—	—
H'BA	SCR	8	TIE	RIE	TE	RE	MPIE	TEIE	CKE1	CKE0
H'BB	TDR	8	—	—	—	—	—	—	—	—
H'BC	SSR	8	TDRE	RDRF	ORER	FER	PER	TEND	MPB	MPBT
H'BD	RDR	8	—	—	—	—	—	—	—	—
H'BE	—		—	—	—	—	—	—	—	—
H'BF	—		—	—	—	—	—	—	—	—

H'C7	—		—	—	—	—	—	—	—	—
H'C8	P5DDR	8	—	—	—	—	$P5_3$DDR	$P5_2$DDR	$P5_1$DDR	$P5_0$DDR
H'C9	P6DDR	8	—	—	$P6_5$DDR	$P6_4$DDR	$P6_3$DDR	—	—	$P6_0$DDR
H'CA	P5DR	8	—	—	—	—	$P5_3$	$P5_2$	$P5_1$	$P5_0$
H'CB	P6DR	8	—	—	$P6_5$	$P6_4$	$P6_3$	—	—	$P6_0$
H'CC	—		—	—	—	—	—	—	—	—
H'CD	P8DDR	8	—	—	—	—	—	—	$P8_1$DDR	$P8_0$DDR
H'CE	P7DR	8	$P7_7$	$P7_6$	$P7_5$	$P7_4$	$P7_3$	$P7_2$	$P7_1$	$P7_0$
H'CF	P8DR	8	—	—	—	—	—	—	$P8_1$	$P8_0$
H'D0	P9DDR	8	—	—	$P9_5$DDR	$P9_4$DDR	$P9_3$DDR	$P9_2$DDR	$P9_1$DDR	$P9_0$DDR
H'D1	PADDR	8	PA_7DDR	PA_6DDR	PA_5DDR	PA_4DDR	PA_3DDR	PA_2DDR	PA_1DDR	PA_0DDR
H'D2	P9DR	8	—	—	$P9_5$	$P9_4$	$P9_3$	$P9_2$	$P9_1$	$P9_0$
H'D3	PADR	8	PA_7	PA_6	PA_5	PA_4	PA_3	PA_2	PA_1	PA_0
H'D4	PBDDR	8	PB_7DDR	—	PB_5DDR	PB_4DDR	PB_3DDR	PB_2DDR	PB_1DDR	PB_0DDR
H'D5	—		—	—	—	—	—	—	—	—
H'D6	PBDR	8	PB_7	—	PB_5	PB_4	PB_3	PB_2	PB_1	PB_0
H'D7	—		—	—	—	—	—	—	—	—
H'D8	P2PCR	8	$P2_7$PCR	$P2_6$PCR	$P2_5$PCR	$P2_4$PCR	$P2_3$PCR	$P2_2$PCR	$P2_1$PCR	$P2_0$PCR
H'D9	—		—	—	—	—	—	—	—	—
H'DA	—		—	—	—	—	—	—	—	—
H'DB	P5PCR	8	—	—	—	—	$P5_3$PCR	$P5_2$PCR	$P5_1$PCR	$P5_0$PCR
H'DC	—		—	—	—	—	—	—	—	—
H'DD	—		—	—	—	—	—	—	—	—
H'DE	—		—	—	—	—	—	—	—	—
H'DF	—		—	—	—	—	—	—	—	—

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	—		—	—	—	—	—	—	—	—
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
H'F2	SYSCR	8	SSBY	STS2	STS1	STS0	UE	NMIEG	—	RAME
H'F3	ADRCR	8	A_{23}E	A_{22}E	A_{21}E	—	—	—	—	—
H'F4	ISCR	8	—	—	IRQ5SC	IRQ4SC	—	—	IRQ1SC	IRQ0SC
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

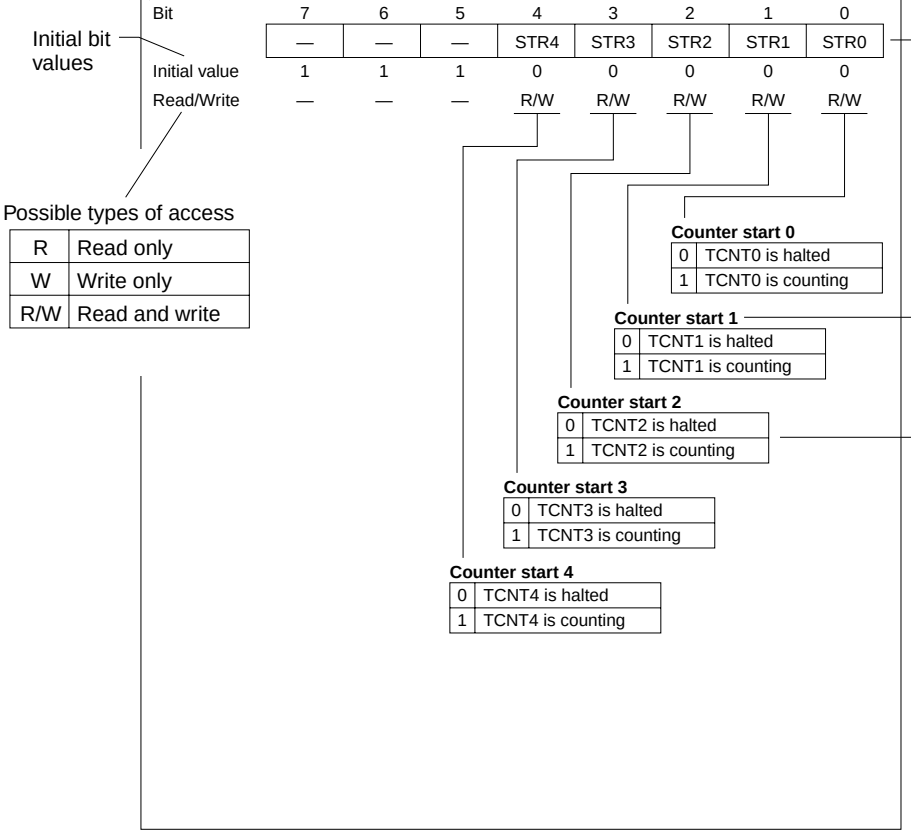
WDT: Watchdog timer

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.


Bit
7
6
5
4
3
2
1
0
—
—
—
STR4
STR3
STR2
STR1
STR0
Initial bit values
Initial value
1
1
1
0
0
0
0
0
Read/Write
—
—
—
R/W
R/W
R/W
R/W
R/W
Possible types of access
R Read only
W Write only
R/W Read and write
Counter start 0
0 TCNT0 is halted
1 TCNT0 is counting
Counter start 1
0 TCNT1 is halted
1 TCNT1 is counting
Counter start 2
0 TCNT2 is halted
1 TCNT2 is counting
Counter start 3
0 TCNT3 is halted
1 TCNT3 is counting
Counter start 4
0 TCNT4 is halted
1 TCNT4 is counting

Program mode

0	Program mode cleared (Initial value
1	Transition to program mode [Setting condition] When FWE = 1, SWE = 1, and PSU

Erase mode

0	Erase mode cleared (Initial value)
1	Transition to erase mode [Setting condition] When FWE = 1, SWE = 1, and ESU = 1

Program-verify mode

0	Program-verify mode cleared (Initial value)
1	Transition to program-verify mode [Setting condition] When FWE = 1 and SWE = 1

Erase-verify mode

0	Erase-verify mode cleared (Initial value)
1	Transition to erase-verify mode [Setting condition] When FWE = 1 and SWE = 1

Program setup

0	Program setup cleared (Initial value)
1	Program setup [Setting condition] When FWE = 1 and SWE = 1

Erase setup

0	Erase setup cleared (Initial value)
1	Erase setup [Setting condition] When FWE = 1 and SWE = 1

Software write enable bit

0	Program/erase disabled (Initial value)
1	Program/erase enabled [Setting condition] When FWE = 1

Flash write enable bit

0	When a low level is input to the FWE pin (hardware protection state)
1	When a high level is input to the FWE pin

Note: 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 dis
Fix the FWE pin low in mode 6.

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 ad
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.

5 to 7	Read/Write	—	—	—	—	R/W*	R/W*	R/W

Reserved bits

RAM select, RAM2, RAM1

Bit 3	Bit 2	Bit 1	RAM Area	RAM Emul
RAMS	RAM2	RAM1		
0	0/1	0/1	H'FFF000 to H'FFF3FF	No emulati
1	0	0	H'000000 to H'0003FF	Mapping R
		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 addre 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 supporte these bits can be modified, but must not be set to 1.

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 Sy

Bit	7	6	5	7	3	2	1
	—	—	—	—	—	—	DIV1
Initial value	1	1	1	1	1	1	0
Read/Write	—	—	—	—	—	—	R/W

Divide bits 1 and 0

Bit 1	Bit 0	Freq
DIV1	DIV0	Divis
0	0	1/1in
	1	1/2
1	0	1/4
	1	1/8

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

Counter start 0

0	TCNT0 is ha
1	TCNT0 is co

Counter start 1

0	TCNT1 is halted
1	TCNT1 is counting

Counter start 2

0	TCNT2 is halted
1	TCNT2 is counting

Counter start 3

0	TCNT3 is halted
1	TCNT3 is counting

Counter start 4

0	TCNT4 is halted
1	TCNT4 is counting

Timer sync 0

0	TCNT0 operates indepe
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

PWM mode 0

0	Channel 0 operates normall
1	Channel 0 operates in PWM

PWM mode 1

0	Channel 1 operates normally
1	Channel 1 operates in PWM mode

PWM mode 2

0	Channel 2 operates normally
1	Channel 2 operates in PWM mode

PWM mode 3

0	Channel 3 operates normally
1	Channel 3 operates in PWM mode

PWM mode 4

0	Channel 4 operates normally
1	Channel 4 operates in PWM mode

Flag direction

0	OVF is set to 1 in TSR2 when TCNT2 overflows or underflows
1	OVF is set to 1 in TSR2 when TCNT2 overflows

Phase counting mode flag

0	Channel 2 operates normally
1	Channel 2 operates in phase counting mode

Buffer mode A3

0	GRA3 operates nor
1	GRA3 is buffered by

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 m
	1	Channels 3 and 4 operate together in reset-synchronized PWM

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 inpu
1	0	TCNT is cleared by GRB compare match or inpu
	1	Synchronous clear: TCNT is cleared in synchron with other synchronized time

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
1	0	0	GRA is an input capture register	GRA captures rising edge of inp
		1		GRA captures falling edge of inp
	1	0		GRA captures both edges of inp
		1		

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 matc
1	0	0	GRB is an input capture register	GRB captures rising edge of input
		1		GRB captures falling edge of input
	1	0		GRB captures both edges of input
		1		

Input capture/compare match interrupt

0	IMIA interrupt requested by IMFA is di
1	IMIA interrupt requested by IMFA is en

Input capture/compare match interrupt enable

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

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 capt signal, when GRA functions as an input capture re

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.

GRA0 H/L—General Register A0 H/L **H'6A, H'6B**

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

Output compare or input capture register

GRB0 H/L—General Register B0 H/L **H'6C, H'6D**

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

Output compare or input capture register

TCR1—Timer Control Register 1 **H'6E**

Bit	7	6	5	4	3	2	1
	—	CCLR1	CCLR0	CKEG1	CKEG0	TPSC2	TPSC1
Initial value	1	0	0	0	0	0	0
Read/Write	—	R/W	R/W	R/W	R/W	R/W	R/W

Note: Bit functions are the same as for ITU0.

TIER1—Timer Interrupt Enable Register 1 **H'70**

Bit	7	6	5	4	3	2	1
	—	—	—	—	—	OVIE	IMIEB
Initial value	1	1	1	1	1	0	0
Read/Write	—	—	—	—	—	R/W	R/W

Note: Bit functions are the same as for ITU0.

TSR1—Timer Status Register 1 **H'71**

Bit	7	6	5	4	3	2	1
	—	—	—	—	—	OVF	IMFB
Initial value	1	1	1	1	1	0	0
Read/Write	—	—	—	—	—	R/(W)*	R/(W)*

Notes: Bit functions are the same as for ITU0.
* Only 0 can be written to clear the flag.

TCNT1 H/L—Timer Counter 1 H/L **H'72, H'73**

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

Note: Bit functions are the same as for ITU0.

GRB1 H/L—General Register B1 H/L **H'76, H'77**

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

Note: Bit functions are the same as for ITU0.

TCR2—Timer Control Register 2 **H'78**

Bit	7	6	5	4	3	2	1
	—	CCLR1	CCLR0	CKEG1	CKEG0	TPSC2	TPSC1
Initial value	1	0	0	0	0	0	0
Read/Write	—	R/W	R/W	R/W	R/W	R/W	R/W

Notes: 1. Bit functions are the same as for ITU0.
2. When channel 2 is used in phase counting mode, the counter clock sourc
bits CKEG1, CKEG0 and TPSC2 to TPSC0 is ignored.

2. Channel 2 does not have a compare match toggle output function. If this s
used, 1 output will be selected automatically.

TIER2—Timer Interrupt Enable Register 2 **H'7A**

Bit	7	6	5	4	3	2	1
	—	—	—	—	—	OVIE	IMIEB
Initial value	1	1	1	1	1	0	0
Read/Write	—	—	—	—	—	R/W	R/W

Note: Bit functions are the same as for ITU0.

Bit function
same as f

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 underflo 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

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

Phase counting mode: up/down-counter
Other modes: up-counter

RENESAS

GRB2 H/L—General Register B2 H/L **H'80, H'81**

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

Note: Bit functions are the same as for ITU0.

TCR3—Timer Control Register 3 **H'82**

Bit	7	6	5	4	3	2	1
	—	CCLR1	CCLR0	CKEG1	CLEG0	TPSC2	TPSC1
Initial value	1	0	0	0	0	0	0
Read/Write	—	R/W	R/W	R/W	R/W	R/W	R/W

Note: Bit functions are the same as for ITU0.

TIOR3—Timer I/O Control Register 3 **H'83**

Bit	7	6	5	4	3	2	1
	—	IOB2	IOB1	IOB0	—	IOA2	IOA1
Initial value	1	0	0	0	1	0	0
Read/Write	—	R/W	R/W	R/W	—	R/W	R/W

Note: Bit functions are the same as for ITU0.

TSR3—Timer Status Register 3 **H'85**

Bit	7	6	5	4	3	2	1
	—	—	—	—	—	OVF	IMFB
Initial value	1	1	1	1	1	0	0
Read/Write	—	—	—	—	—	R/(W)*	R/(W)*

Bit functio
same as f

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 underflo H'0000 to H'FFFF

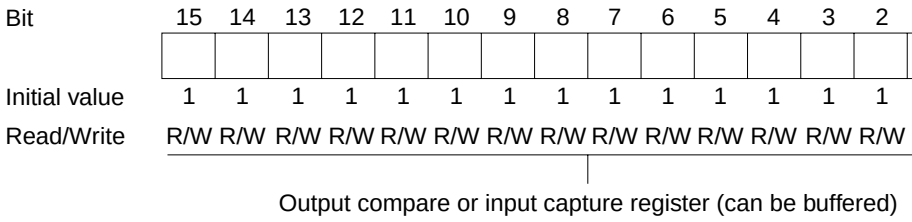
Note: * Only 0 can be written to clear the flag.

TCNT3 H/L—Timer Counter 3 H/L **H'86, H'87**

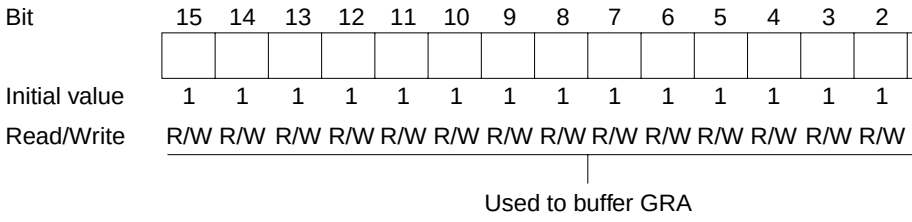
Bit	15	14	13	12	11	10	9	8	7	6	5	4	3	2
Initial value	0	0	0	0	0	0	0	0	0	0	0	0	0	0
Read/Write	R/W	R/W	R/W	R/W	R/W	R/W	R/W	R/W	R/W	R/W	R/W	R/W	R/W	R/W

Complementary PWM mode: up/down counter
Other modes: up-counter

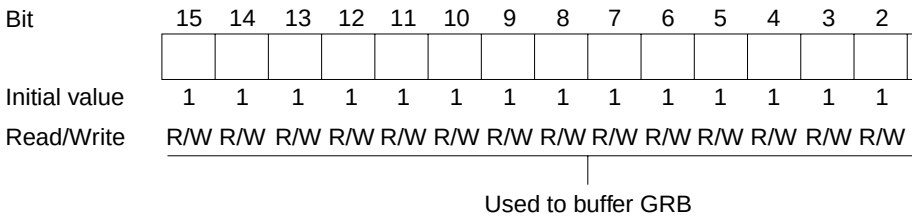
GRB3 H/L—General Register B3 H/L **H'8A, H'8B**



BRA3 H/L—Buffer Register A3 H/L **H'8C, H'8D**



BRB3 H/L—Buffer Register B3 H/L **H'8E, H'8F**



Master enable $TIOCA_3$

0	$TIOCA_3$ output is disabled regardless of TIOR3, TMDR, and TFCR
1	$TIOCA_3$ is enabled for output according to TIOR3, TMDR, and TFC

Master enable $TIOCA_4$

0	$TIOCA_4$ output is disabled regardless of TIOR4, TMDR, and TFCR se
1	$TIOCA_4$ is enabled for output according to TIOR4, TMDR, and TFCR

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

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 inve

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 cleare disabling ITU output.

TCR4—Timer Control Register 4 H'92

Bit	7	6	5	4	3	2	1
	—	CCLR1	CCLR0	CKEG1	CKEG0	TPSC2	TPSC1
Initial value	1	0	0	0	0	0	0
Read/Write	—	R/W	R/W	R/W	R/W	R/W	R/W

Note: Bit functions are the same as for ITU0.

TIER4—Timer Interrupt Enable Register 4 **H'94**

Bit	7	6	5	4	3	2	1
	—	—	—	—	—	OVIE	IMIEB
Initial value	1	1	1	1	1	0	0
Read/Write	—	—	—	—	—	R/W	R/W

Note: Bit functions are the same as for ITU0.

TSR4—Timer Status Register 4 **H'95**

Bit	7	6	5	4	3	2	1
	—	—	—	—	—	OVF	IMFB
Initial value	1	1	1	1	1	0	0
Read/Write	—	—	—	—	—	R/(W)*	R/(W)*

Notes: Bit functions are the same as for ITU0.
* Only 0 can be written to clear the flag.

TCNT4 H/L—Timer Counter 4 H/L **H'96, H'97**

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

Note: Bit functions are the same as for ITU3.

GRB4 H/L—General Register B4 H/L **H'9A, H'9B**

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

Note: Bit functions are the same as for ITU3.

BRA4 H/L—Buffer Register A4 H/L **H'9C, H'9D**

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

Note: Bit functions are the same as for ITU3.

BRB4 H/L—Buffer Register B4 H/L **H'9E, H'9F**

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

Note: Bit functions are the same as for ITU3.

Group 0 non-overlap

0	Normal TPC output in group 0 Output values change at compare match A in the selected ITU
1	Non-overlapping TPC output in group 0, controlled by compare 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 cha
1	Non-overlapping TPC output in group 1, controlled by compare ma 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 channe
1	Non-overlapping TPC output in group 2, controlled by compare match A and B in the selected ITU channel

Group 3 non-overlap

0	Normal TPC output in group 3 Output values change at compare match A in the selected ITU channel
1	Non-overlapping TPC output in group 3, controlled by compare match A and B in the selected ITU channel

Group 0 compare match select 1 and 0

Bit 1	Bit 0	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
	1	TPC output group 0 (TP_3 to TP_0) is triggered by compare match in ITU
1	0	TPC output group 0 (TP_3 to TP_0) is triggered by compare match in ITU
	1	TPC output group 0 (TP_3 to TP_0) is triggered by compare match in ITU

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 c
	1	TPC output group 1 (TP_7 to TP_4) is triggered by compare match in ITU c
1	0	TPC output group 1 (TP_7 to TP_4) is triggered by compare match in ITU c
	1	TPC output group 1 (TP_7 to TP_4) is triggered by compare match in ITU c

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 c
	1	TPC output group 2 (TP_{11} to TP_8) is triggered by compare match in ITU c
1	0	TPC output group 2 (TP_{11} to TP_8) is triggered by compare match in ITU c
	1	TPC output group 2 (TP_{11} to TP_8) is triggered by compare match in ITU c

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 c
	1	TPC output group 3 (TP_{15} to TP_{12})* is triggered by compare match in ITU c
1	0	TPC output group 3 (TP_{15} to TP_{12})* is triggered by compare match in ITU c
	1	TPC output group 3 (TP_{15} to TP_{12})* is triggered by compare match in ITU c

Note: * Since this LSI does not have a TP_{14} pin, the TP_{14} signal cannot be output off-chip.

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
1	TPC outputs TP_{15} to TP_8* are enabled (NDR15 to NDR8 are transferred to PB_7 to PB

Note: * Since this LSI does not have a TP_{14} pin, the TP_{14} signal ca output off-chip.

NDERA—Next Data Enable Register A H'A3

Bit	7	6	5	4	3	2	1
	NDER7	NDER6	NDER5	NDER4	NDER3	NDER2	NDER1
Initial value	0	0	0	0	0	0	0
Read/Write	R/W	R/W	R/W	R/W	R/W	R/W	R/W

Next data enable 7 to 0

Bits 7 to 0	Description
NDER7 to NDER0	
0	TPC outputs TP_7 to TP_0 are disabled (NDR7 to NDR0 are not transferred to PA_7 to
1	TPC outputs TP_7 to TP_0 are enabled (NDR7 to NDR0 are transferred to PA_7 to PA_0

Next output data for
TPC output group 3*

Next output data fo
TPC output group 2

Address H'FFA6

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

- Different output triggers for TPC output groups 2 and 3

  Address H'FFA4

Bit	7	6	5	4	3	2	1
	NDR15	NDR14	NDR13	NDR12	—	—	—
Initial value	0	0	0	0	1	1	1
Read/Write	R/W	R/W	R/W	R/W	—	—	—

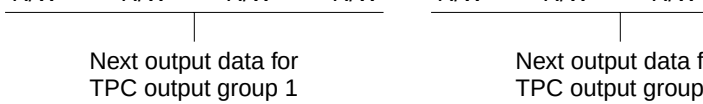
Next output data for
TPC output group 3*

Address H'FFA6

Bit	7	6	5	4	3	2	1
	—	—	—	—	NDR11	NDR10	NDR9
Initial value	1	1	1	1	0	0	0
Read/Write	—	—	—	—	R/W	R/W	R/W

Next output data fo
TPC output group

Note: * Since this LSI does not have a TP_{14} pin, the TP_{14} signal cannot be output off-



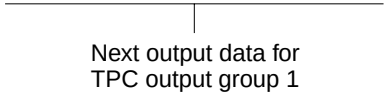
Address H'FFA7

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

- Different output triggers for TPC output groups 0 and 1

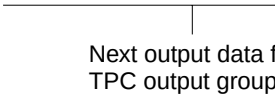
Address H'FFA5

Bit	7	6	5	4	3	2	1
	NDR7	NDR6	NDR5	NDR4	—	—	—
Initial value	0	0	0	0	1	1	1
Read/Write	R/W	R/W	R/W	R/W	—	—	—



Address H'FFA7

Bit	7	6	5	4	3	2	1
	—	—	—	—	NDR3	NDR2	NDR1
Initial value	1	1	1	1	0	0	0
Read/Write	—	—	—	—	R/W	R/W	R/W



Clock select 2 to 0

CKS2	CKS1	CKS0	D
0	0	0	ϕ
		1	ϕ
	1	0	ϕ
		1	ϕ
1	0	0	ϕ
		1	ϕ
	1	0	ϕ
		1	ϕ

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.

Count value

RSTCSR—Reset Control/Status Register **H'AB (read), H'AA (write)**

Bit	7	6	5	4	3	2	1
	WRST	RSTOE	—	—	—	—	—
Initial value	0	0	1	1	1	1	1
Read/Write	R/(W)*	R/W	—	—	—	—	—

Reset output enable

0	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.

Clock select 1 and

Bit 1	Bit 0	Clock
CKS1	CKS0	
0	0	φ clo
	1	φ/4 c
1	0	φ/16
	1	φ/64

Multiprocessor mode

0	Multiprocessor function
1	Multiprocessor format se

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

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 generi
		Synchronous mode	Internal clock, SCK pin used for serial
	1	Asynchronous mode	Internal clock, SCK pin used for clock
		Synchronous mode	Internal clock, SCK pin used for serial
1	0	Asynchronous mode	External clock, SCK pin used for clock
		Synchronous mode	External clock, SCK pin used for seria
	1	Asynchronous mode	External clock, SCK pin used for clock
		Synchronous mode	External clock, SCK pin used for seria

Transmit-end interrupt enable

0	Transmit-end interrupt requests (TEI) are disabled
1	Transmit-end interrupt requests (TEI) are enabled

Multiprocessor interrupt enable

0	Multiprocessor interrupts are disabled (normal receive operation)
1	Multiprocessor interrupts are enabled

Receive enable

0	Receiving is disabled
1	Receiving is enabled

Transmit enable

0	Transmitting is disabled
1	Transmitting is enabled

Receive interrupt enable

0	Receive-data-full (RXI) and receive-error (ERI) interrupt requests are disabled
1	Receive-data-full (RXI) and receive-error (ERI) interrupt requests are enabled

Transmit interrupt enable

0	Transmit-data-empty interrupt request (TXI) is disabled
1	Transmit-data-empty interrupt request (TXI) is enabled

Multiprocessor bit

0	Multiprocessor bit value in receive data is 0
1	Multiprocessor bit value in receive data is 1

Multiprocessor bit tra

0	Multiprocessor bit transmit data is 0
1	Multiprocessor bit 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 writ PER.
1	[Setting condition] Parity error: (parity of receive data d 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 ORER.
1	[Setting condition] Overrun error (reception of next ser 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.

SCMR—Smart Card Mode Register **H'B6**

Bit	7	6	5	4	3	2	1
	—	—	—	—	SDIR	SINV	—
Initial value	1	1	1	1	0	0	1
Read/Write	—	—	—	—	R/W	R/W	—

Smart card interface mode select

0	Smart card interface function is di
1	Smart card interface function is en

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

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

BRR—Bit Rate Register **H'B9**

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

Note: Bit functions are the same as for SCI0.

SCR—Serial Control Register **H'BA**

Bit	7	6	5	4	3	2	1
	TIE	RIE	TE	RE	MPIE	TEIE	CKE1
Initial value	0	0	0	0	0	0	0
Read/Write	R/W	R/W	R/W	R/W	R/W	R/W	R/W

Note: Bit functions are the same as for SCI0.

TDR—Transmit Data Register **H'BB**

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

Note: Bit functions are the same as for SCI0.

* Only 0 can be written to clear the flag.

RDR—Receive Data Register **H'BD**

Bit	7	6	5	4	3	2	1
Initial value	0	0	0	0	0	0	0
Read/Write	R	R	R	R	R	R	R

Note: Bit functions are the same as for SCI0.

P1DDR—Port 1 Data Direction Register **H'C0**

Bit		7	6	5	4	3	2	1
		$P1_7$DDR	$P1_6$DDR	$P1_5$DDR	$P1_4$DDR	$P1_3$DDR	$P1_2$DDR	$P1_1$D
Modes 1 and 3	Initial value	1	1	1	1	1	1	1
	Read/Write	—	—	—	—	—	—	—
Modes 5 to 7	Initial value	0	0	0	0	0	0	0
	Read/Write	W	W	W	W	W	W	W

Port 1 input/output select

0	Generic input pin
1	Generic output pin

Port 2 input/output select

0	Generic input pin
1	Generic output pin

P1DR—Port 1 Data Register H'C2

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

Data for port 1 pins

P2DR—Port 2 Data Register H'C3

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

Data for port 2 pins

0	Generic input pin
1	Generic output pin

P3DR—Port 3 Data Register **H'C6**

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

Data for port 3 pins

P5DDR—Port 5 Data Direction Register **H'C8**

	Bit	7	6	5	4	3	2	1
		—	—	—	—	$P5_3DDR$	$P5_2DDR$	$P5_1D$
Modes 1 and 3	Initial value	1	1	1	1	1	1	1
	Read/Write	—	—	—	—	—	—	—
Modes 5 to 7	Initial value	1	1	1	1	0	0	0
	Read/Write	—	—	—	—	W	W	W

Port 5 input/outpu

0	Generic input
1	Generic output

Port 6 input/output select

0	Generic input
1	Generic output

P5DR—Port 5 Data Register H'CA

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

Data for port 5 pins

P6DR—Port 6 Data Register H'CB

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

Data for port 6 pins

Port 8 input/output select

0	Generic input
1	Generic output

P7DR—Port 7 Data Register — H'CE

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

Data for port 7 pins

Note: * Determined by pins $P7_7$ to $P7_0$.

P8DR—Port 8 Data Register — H'CF

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

Data for port 8 pins

Port 9 input/output select

0	Generic input
1	Generic output

PADDR—Port A Data Direction Register H'D1

	Bit	7	6	5	4	3	2	1
		PA_7DDR	PA_6DDR	PA_5DDR	PA_4DDR	PA_3DDR	PA_2DDR	PA_1D
Mode 3	Initial value	1	0	0	0	0	0	0
	Read/Write	—	W	W	W	W	W	W
Modes 1 and 5 to 7	Initial value	0	0	0	0	0	0	0
	Read/Write	W	W	W	W	W	W	W

Port A input/output select

0	Generic input
1	Generic output

P9DR—Port 9 Data Register H'D2

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

Data for port 9 pins

PBDDR—Port B Data Direction Register **H'D4**

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

Port B input/output select

0	Generic input
1	Generic output

PBDR—Port B Data Register **H'D6**

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

Data for port B pins

0	Input pull-up transistor is off
1	Input pull-up transistor is on

Note: Valid when the corresponding P2DDR bit is cleared to 0 (designating generic i

P5PCR—Port 5 Input Pull-Up Control Register H'DB

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

Port 5 input pull-up cont

0	Input pull-up transistor
1	Input pull-up transistor

Note: Valid when the corresponding P5DDR bit is cleared to 0 (designating generic i

ADDRA H/L—A/D Data Register A H/L H'E0, H'E1

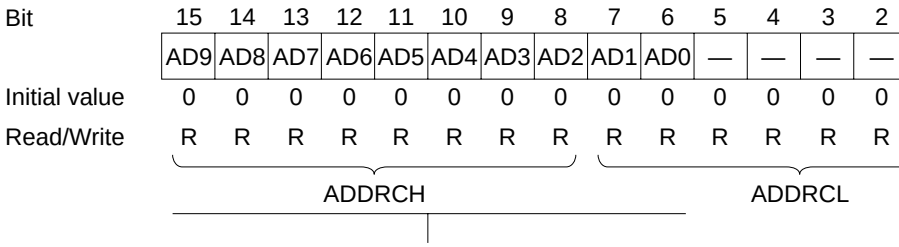
Bit	15	14	13	12	11	10	9	8	7	6	5	4	3	2
	AD9	AD8	AD7	AD6	AD5	AD4	AD3	AD2	AD1	AD0	—	—	—	—
Initial value	0	0	0	0	0	0	0	0	0	0	0	0	0	0
Read/Write	R	R	R	R	R	R	R	R	R	R	R	R	R	R

ADDRAH ADDRAL

A/D conversion data
10-bit data giving an
A/D conversion result

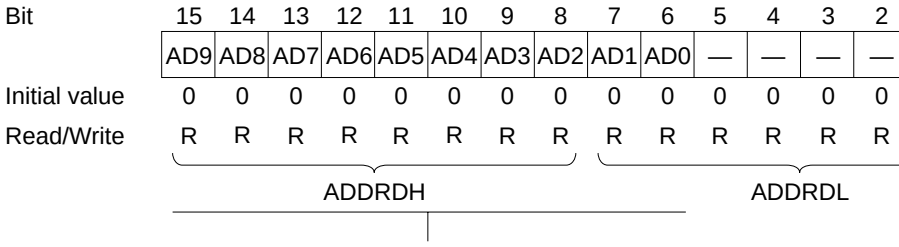
A/D conversion data
10-bit data giving an
A/D conversion result

ADDRC H/L—A/D Data Register C H/L H'E4, H'E5



A/D conversion data
10-bit data giving an
A/D conversion result

ADDRD H/L—A/D Data Register D H/L H'E6, H'E7



A/D conversion data
10-bit data giving an
A/D conversion result

0	A/D conversion cannot be externally triggered
1	A/D conversion starts at the fall of the external trigger signal ($\overline{AD}$

Clock select

0	Conversion time = 266 states (maximum)
1	Conversion time = 134 states (maximum)

Channel select 2 to 0

Group Selection	Channel Selection		Description	
CH2	CH1	CH0	Single Mode	Scan
0	0	0	AN_0	AN_0
		1	AN_1	AN_0,
	1	0	AN_2	AN_0
		1	AN_3	AN_0
1	0	0	AN_4	AN_4
		1	AN_5	AN_4,
	1	0	AN_6	AN_4
		1	AN_7	AN_4

Scan mode

0	Single mode
1	Scan mode

A/D start

0	A/D conversion is stopped
1	Single mode: A/D conversion starts; ADST is automatically cleared to 0 whe conversion ends Scan mode: A/D conversion starts and continues, cycling among the selec channels, until ADST is cleared to 0 by software, by a reset, o transition to standby mode

A/D interrupt enable

0	A/D end interrupt request is disabled
1	A/D end interrupt request is enabled

A/D end flag

0	[Clearing condition] Read ADF while ADF = 1, then write 0 in ADF
1	[Setting conditions] Single mode: A/D conversion ends Scan mode: A/D conversion ends in all selected channels

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

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

Bit	7	6	5	4	3	2	1
	—	—	—	—	WMS1	WMS0	WC1
Initial value	1	1	1	1	0	0	1
Read/Write	—	—	—	—	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
0	0	No wait states ins wait-state controll
	1	1 state inserted
1	0	2 states inserted
	1	3 states inserted

0	Wait-state control is disabled (pin wait mode 0
1	Wait-state control is enabled

MDCR—Mode Control Register **H'F1** **Sy**

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

Mode select 2 to 0

Bit 2	Bit 1	Bit 0	Operating mo
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).

RAM enable

0	On-chip RAM is
1	On-chip RAM is

NMI edge select

0	An interrupt is requested at the falling edg
1	An interrupt is requested at the rising edg

User bit enable

0	CCR bit 6 (UI) is used as an interrupt mask b
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

Read/Write	R/W	R/W	R/W	—	—	—	—

Address 23 to 21 enable

0	Address output
1	I/O pins other than the above

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 $\overline{I}$ inputs are low
1	Interrupts are requested by falling-edge input at $\overline{IRQ}_5$, $\overline{I}$ $\overline{IRQ}_1$ and $\overline{IRQ}_0$

IER—IRQ Enable Register H'F5 Interrup

Bit	7	6	5	4	3	2	1
	—	—	IRQ5E	IRQ4E	—	—	IRQ1E
Initial value	0	0	0	0	0	0	0
Read/Write	R/W	R/W	R/W	R/W	R/W	R/W	R/W

IRQ_5, IRQ_4, IRQ_1, IRQ_0 enable

0	IRQ_5, IRQ_4, IRQ_1 and IRQ_0 interrupts are di
1	IRQ_5, IRQ_4, IRQ_1 and IRQ_0 interrupts are er

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 I • IRQnSC = 0, $\overline{IRQn}$ input is high, and interrupt e handling is carried out. • IRQnSC = 1 and IRQn interrupt exception hand carried out.
1	[Setting conditions] • IRQnSC = 0 and $\overline{IRQn}$ input is low. • IRQnSC = 1 and $\overline{IRQn}$ input changes from high

Note: n = 5, 4, 1 and 0

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

0	Priority level 0 (low priority)
1	Priority level 1 (high priority)

- Interrupt sources controlled by each bit

	Bit 7 IPRA7	Bit 6 IPRA6	Bit 5 —	Bit 4 IPRA4	Bit 3 IPRA3	Bit 2 IPRA2	Bit 1 IPRA1
Interrupt source	IRQ_0	IRQ_1	—	IRQ_4, IRQ_5	WDT	ITU channel 0	ITU channel 1

IPRB—Interrupt Priority Register B **H'F9** **Interrup**

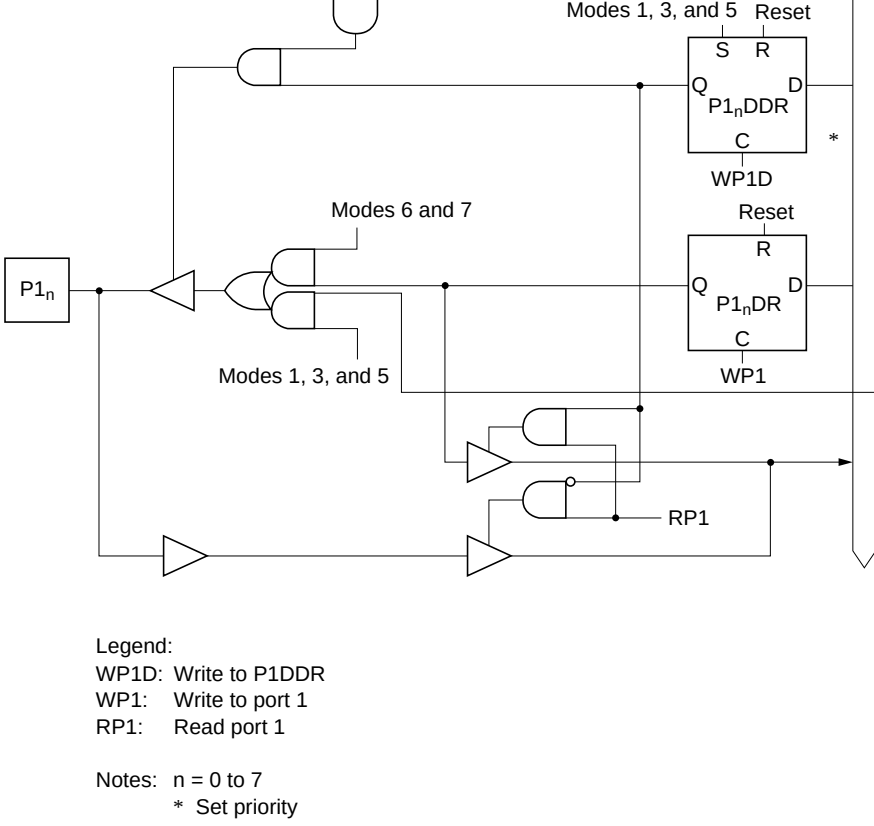
Bit	7	6	5	4	3	2	1
	IPRB7	IPRB6	—	—	IPRB3	IPRB2	IPRB1
Initial value	0	0	0	0	0	0	0
Read/Write	R/W	R/W	R/W	R/W	R/W	R/W	R/W

Priority level B7, B6, B3 to B1

0	Priority level 0 (low priority)
1	Priority level 1 (high priority)

- Interrupt sources controlled by each bit

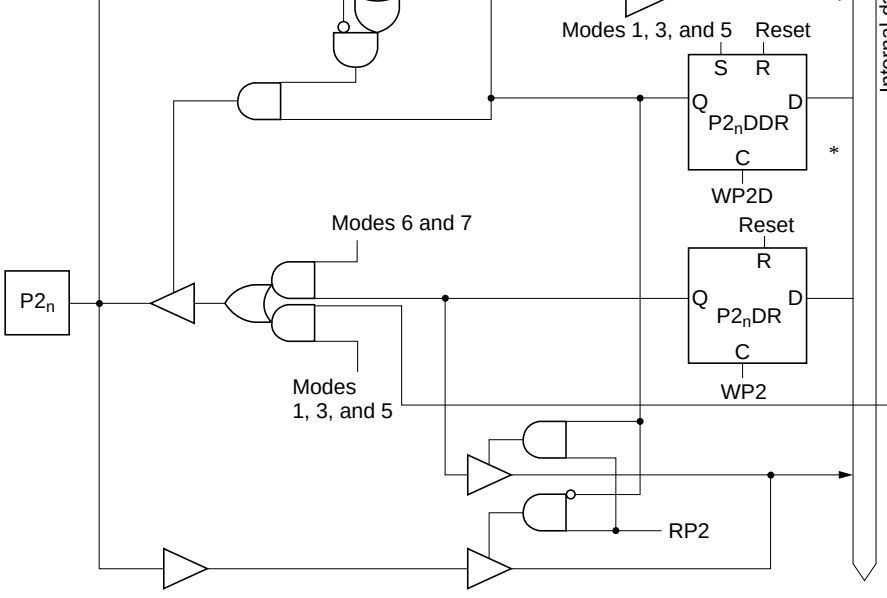
	Bit 7 IPRB7	Bit 6 IPRB6	Bit 5 —	Bit 4 —	Bit 3 IPRB3	Bit 2 IPRB2	Bit 1 IPRB1
Interrupt source	ITU channel 3	ITU channel 4	—	—	SCI channel 0	SCI channel 1	A/D converte



Legend:
WP1D: Write to P1DDR
WP1: Write to port 1
RP1: Read port 1

Notes: n = 0 to 7
* Set priority

Figure C.1 Port 1 Block Diagram

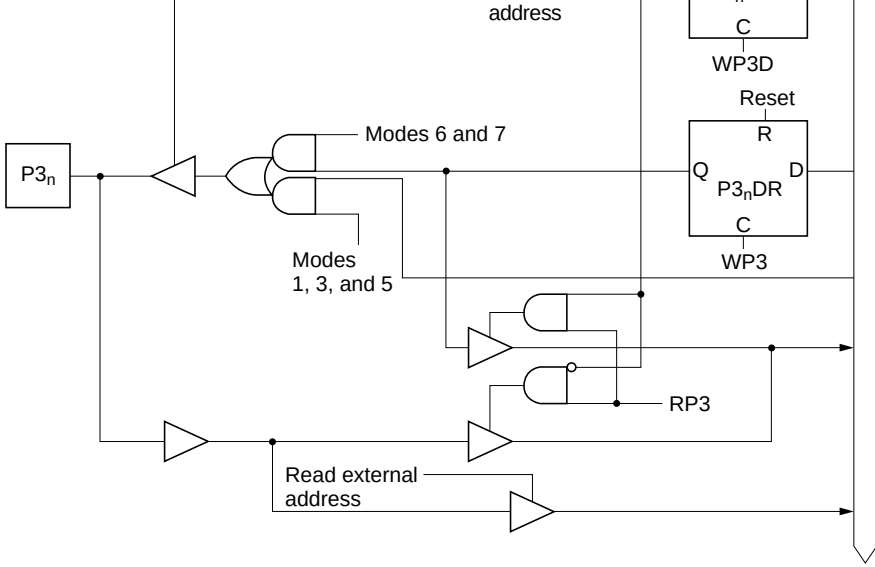


Legend:

WP2P: Write to P2PCR
RP2P: Read P2PCR
WP2D: Write to P2DDR
WP2: Write to port 2
RP2: Read port 2

Notes: n = 0 to 7
* Set priority

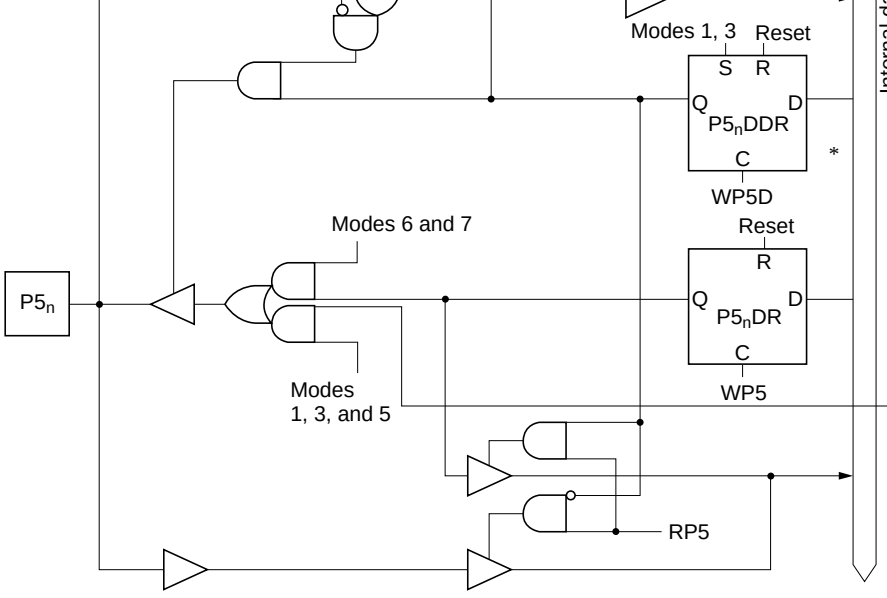
Figure C.2 Port 2 Block Diagram



Legend:
WP3D: Write to P3DDR
WP3: Write to port 3
RP3: Read port 3

Note: n = 0 to 7

Figure C.3 Port 3 Block Diagram



Legend:
WP5P: Write to P5PCR
RP5P: Read P5PCR
WP5D: Write to P5DDR
WP5: Write to port 5
RP5: Read port 5

Notes: n = 0 to 3
* Set priority

Figure C.4 Port 5 Block Diagram

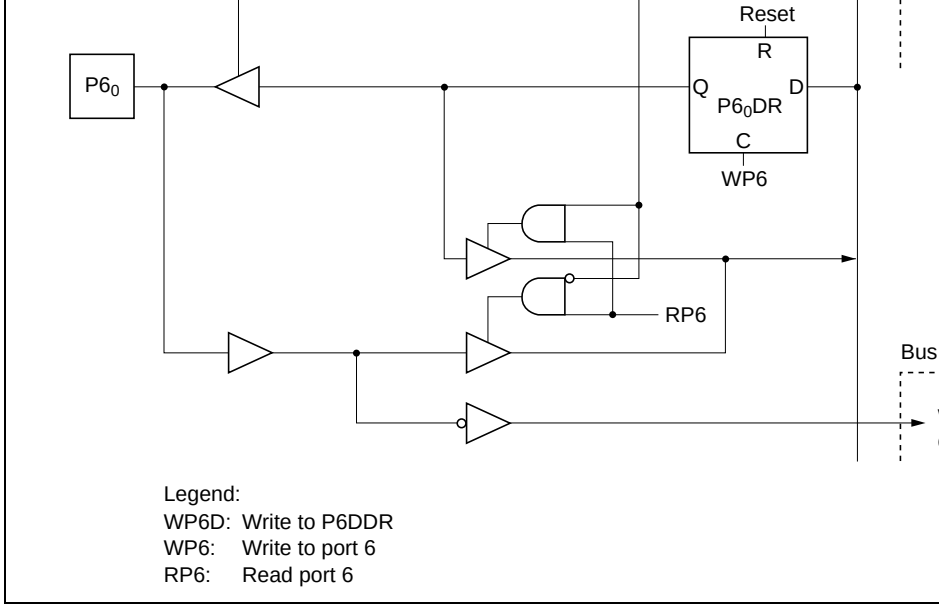
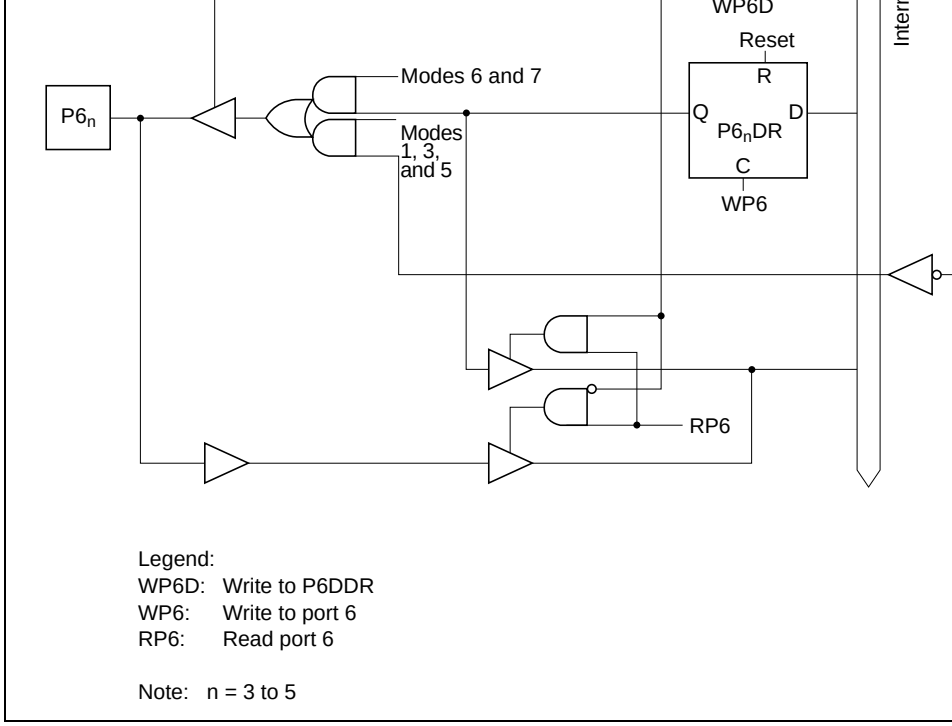


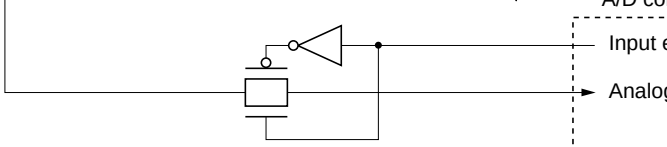
Figure C.5 (a) Port 6 Block Diagram (Pin P6$_0$)



Legend:
WP6D: Write to P6DDR
WP6: Write to port 6
RP6: Read port 6

Note: n = 3 to 5

Figure C.5 (b) Port 6 Block Diagram (Pins $P6_3$ to $P6_5$)



Legend:
RP7: Read port 7

Note: n = 0 to 7

Figure C.6 Port 7 Block Diagram

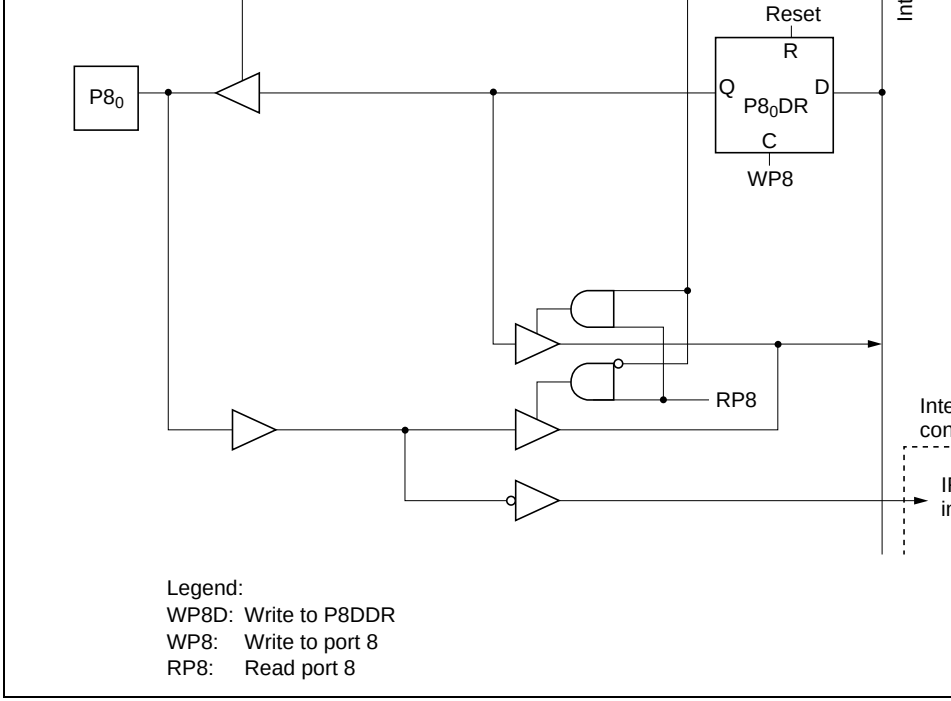


Figure C.7(a) Port 8 Block Diagram (Pin P8$_0$)

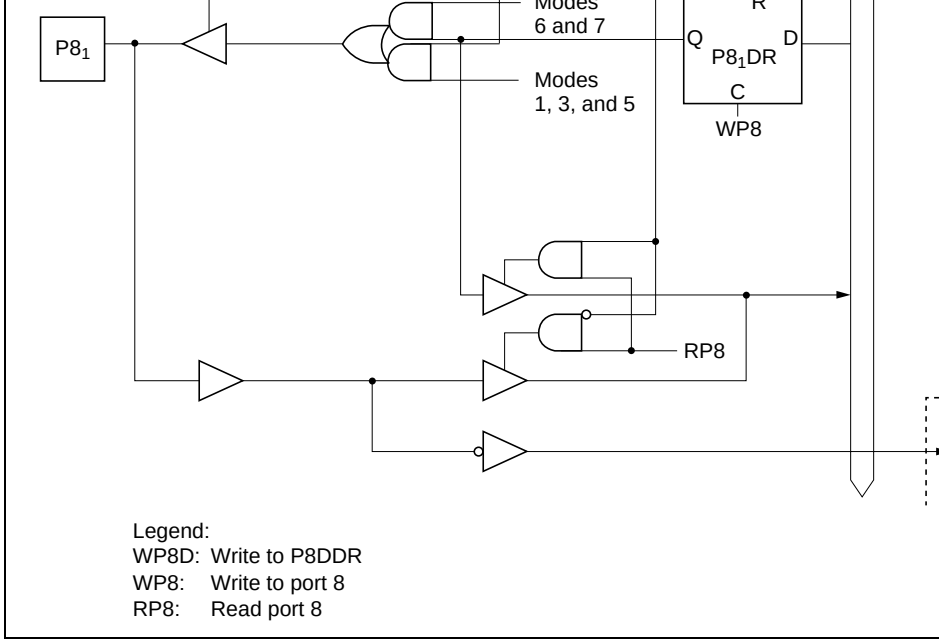
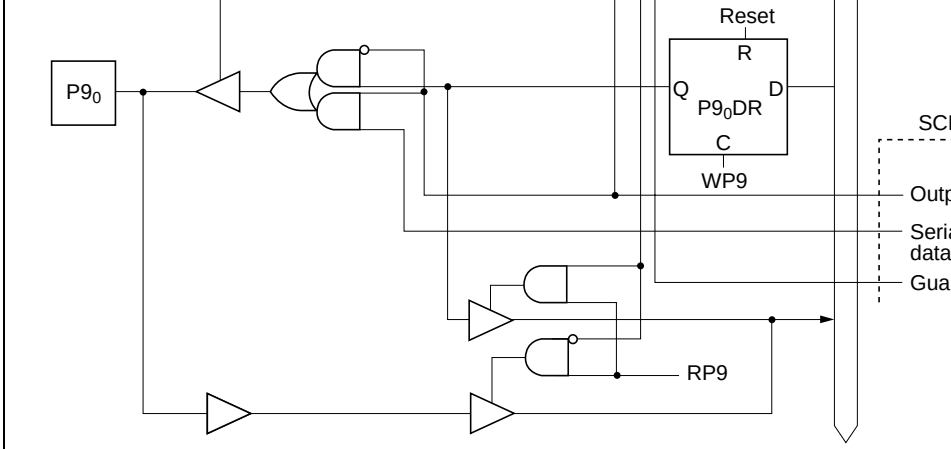
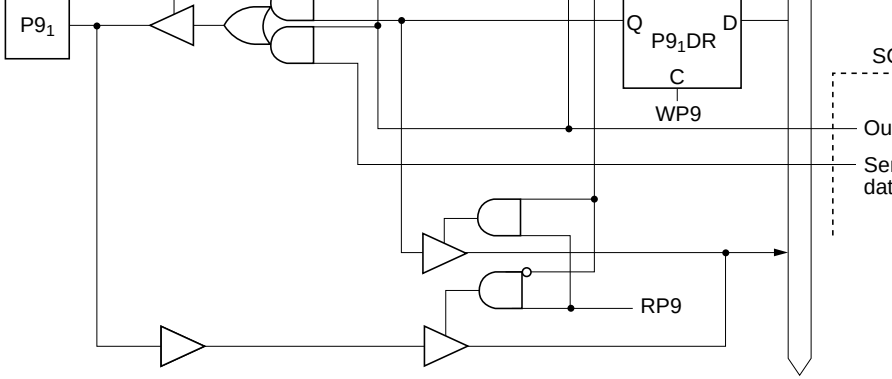


Figure C.7 (b) Port 8 Block Diagram (Pin P8$_1$)



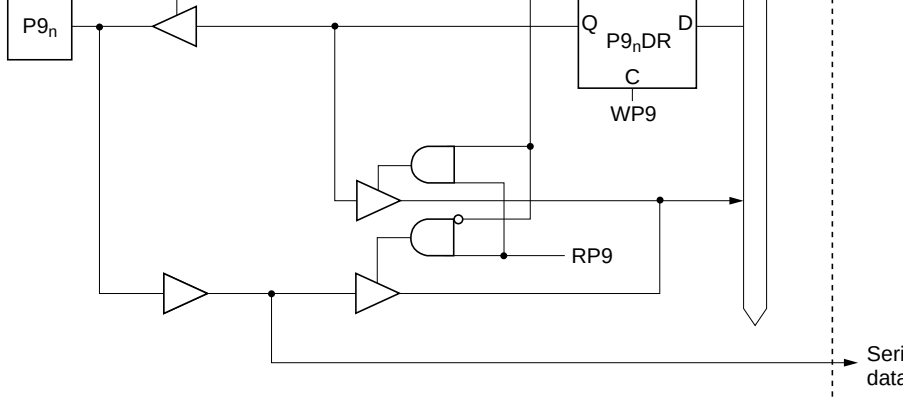
Legend:
WP9D: Write to P9DDR
WP9: Write to port 9
RP9: Read port 9

Figure C.8 (a) Port 9 Block Diagram (Pin P9$_0$)



Legend:
WP9D: Write to P9DDR
WP9: Write to port 9
RP9: Read port 9

Figure C.8 (b) Port 9 Block Diagram (Pin P9$_1$)



Legend:
WP9D: Write to P9DDR
WP9: Write to port 9
RP9: Read port 9

Note: n = 2, 3

Figure C.8 (c) Port 9 Block Diagram (Pin $P9_2$, $P9_3$)

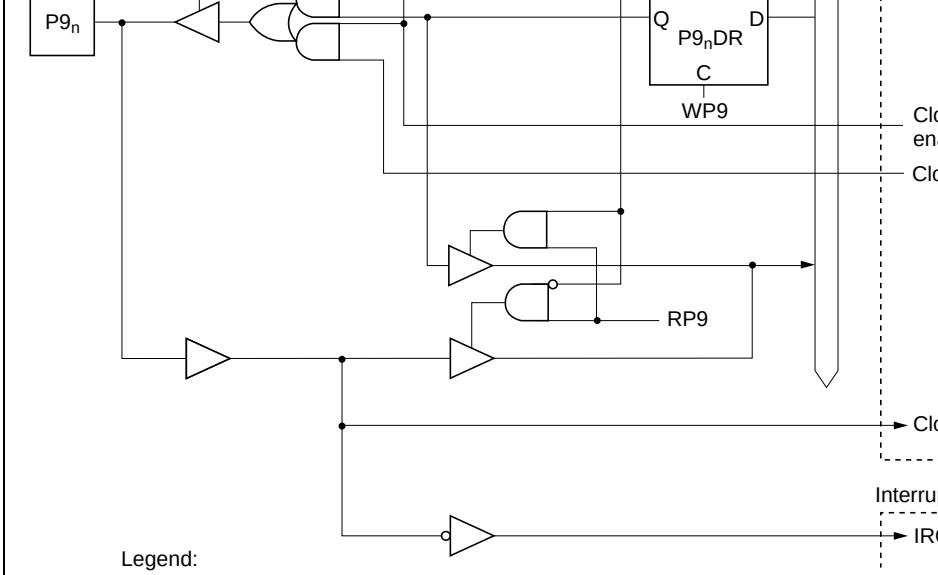
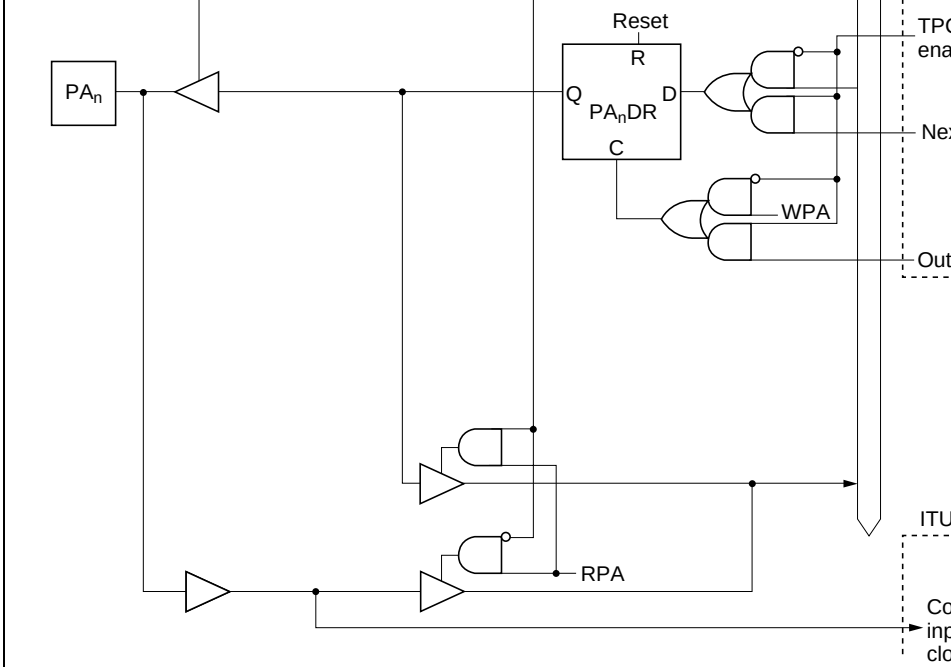


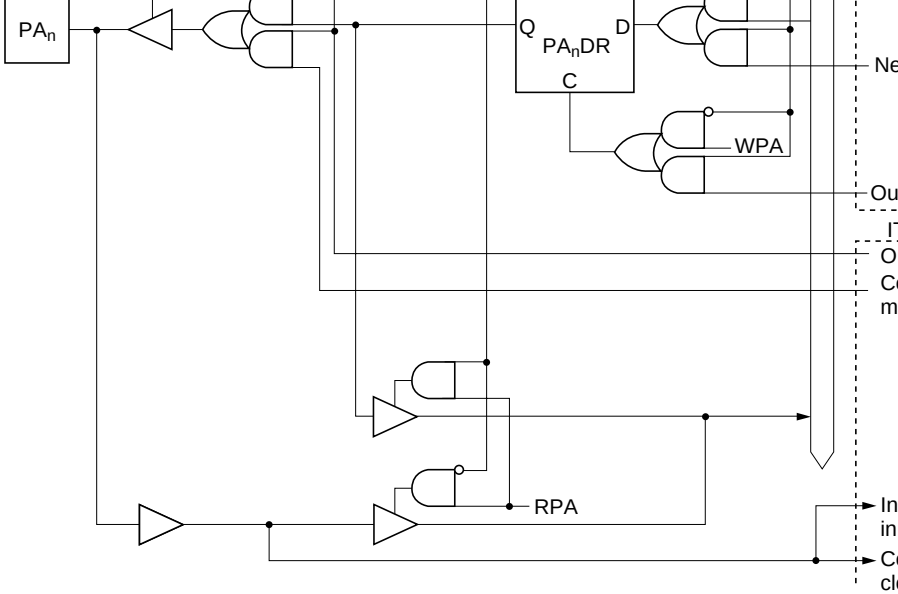
Figure C.8 (d) Port 9 Block Diagram (Pin $P9_4$, $P9_5$)



Legend:
WPAD: Write to PADDR
WPA: Write to port A
RPA: Read port A

Note: n = 0 or 1

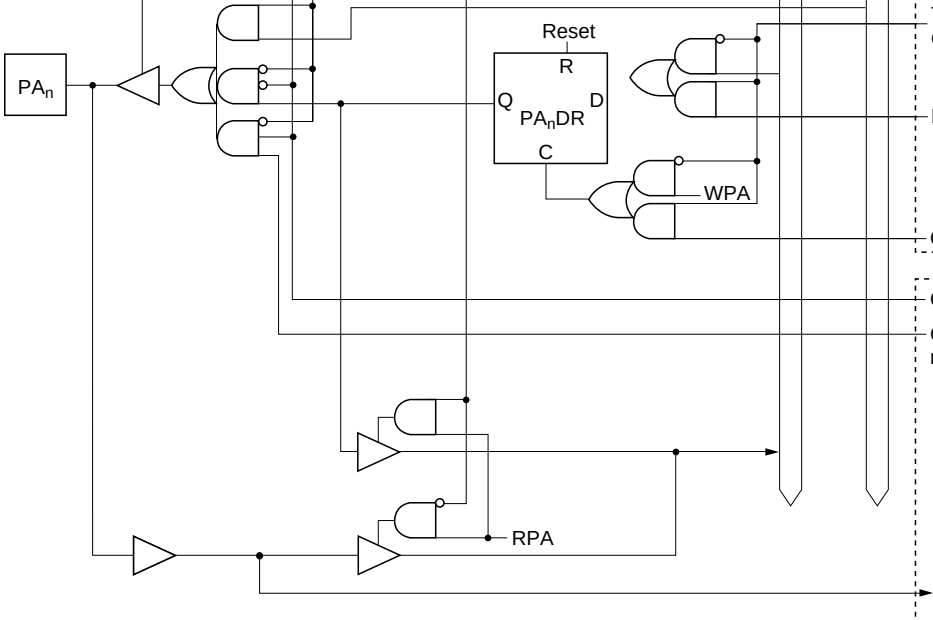
Figure C.9 (a) Port A Block Diagram (Pins PA_0, PA_1)



Legend:
WPAD: Write to PADDR
WPA: Write to port A
RPA: Read port A

Note: n = 2 or 3

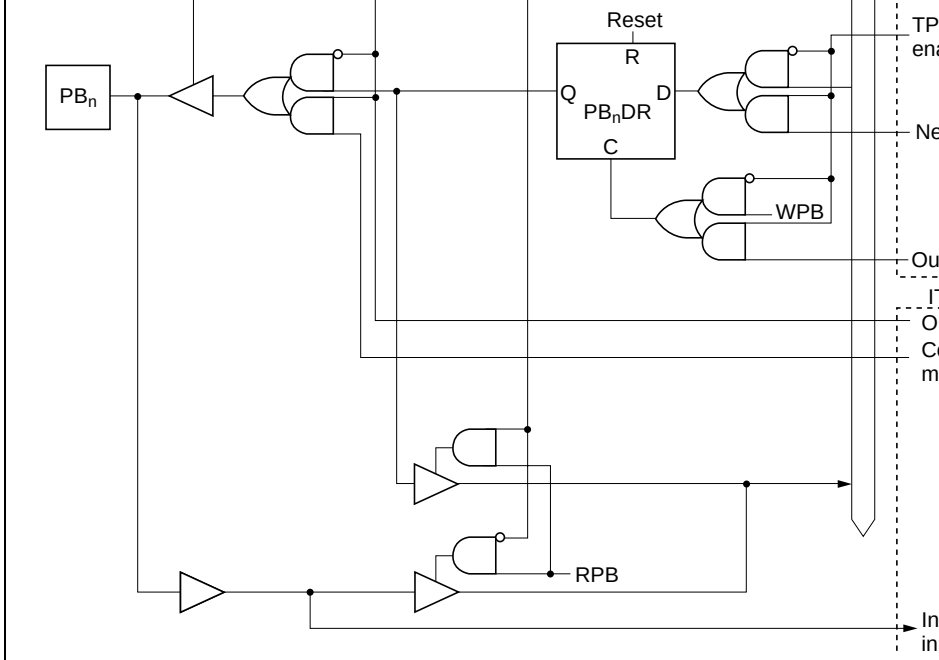
Figure C.9 (b) Port A Block Diagram (Pins PA$_2$, PA$_3$)



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.

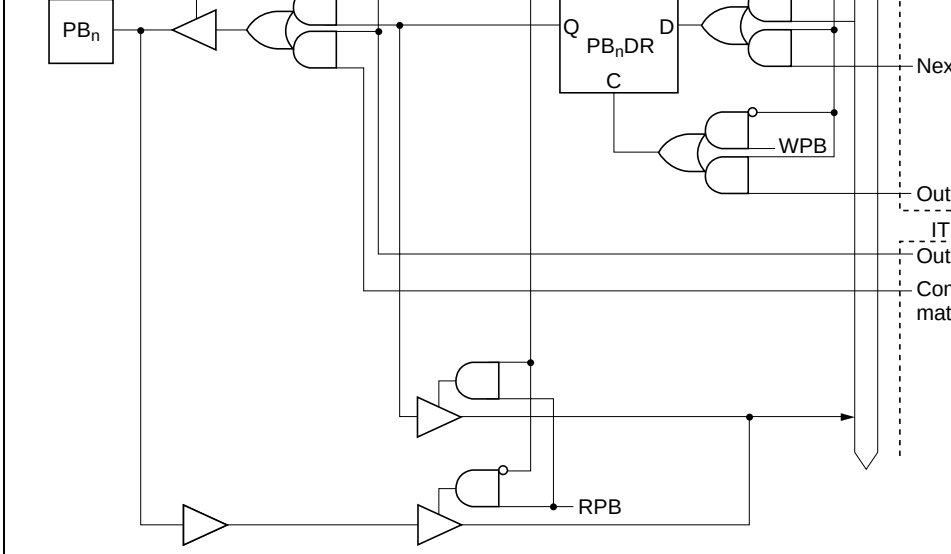
Figure C.9 (c) Port A Block Diagram (Pins PA$_4$ to PA$_7$)



Legend:
WPBD: Write to PBDDR
WPB: Write to port B
RPB: Read port B

Note: n = 0 to 3

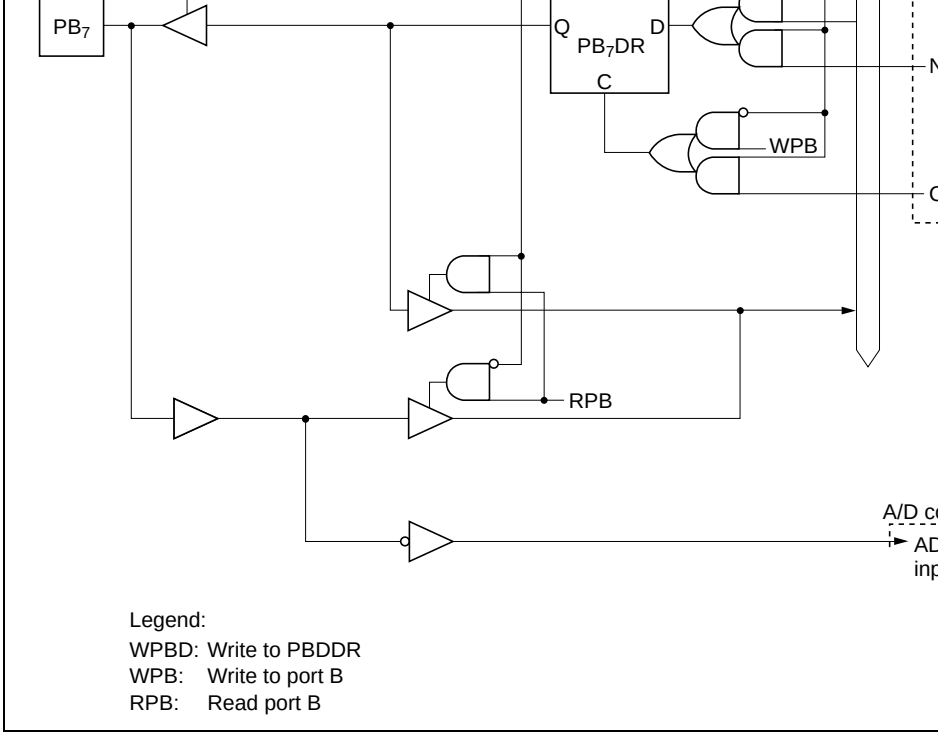
Figure C.10 (a) Port B Block Diagram (Pins PB_0 to PB_3)



Legend:
WPBD: Write to PBDDR
WPB: Write to port B
RPB: Read port B

Note: n = 4 or 5

Figure C.10 (b) Port B Block Diagram (Pins PB_4, PB_5)



Legend:
WPBD: Write to PBDDR
WPB: Write to port B
RPB: Read port B

Figure C.10 (c) Port B Block Diagram (Pin PB_7)

ϕ	—	Clock output	T	H	Clock output
$\overline{RESO}$*[1]	—	T*[2]	T	T	$\overline{RESO}$
$P1_7$ to $P1_0$	1, 3	L	T	T	A_7 to A_0
	5	T	T	keep	Input port (D
				T	A_7 to A_0 (DD
	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 (D
				T	A_{15} to A_8 (DD
	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 (D
				T	A_{19} to A_{16} (D
	6, 7	T	T	keep	I/O port
$P6_0$	1, 3, 5	T	T	keep	I/O port, $\overline{WA}$
	6, 7	T	T	keep	I/O port
$P6_5$ to $P6_3$	1, 3, 5	H	T	T	$\overline{WR}$, $\overline{RD}$, $\overline{AS}$
	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

PA_6 to PA_4	3	T	T	[ADRCR = 0] T	(ADRCR = A_{21} to A_{23}
				[ADRCR = 1] keep	(ADRCR = 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 previou

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.

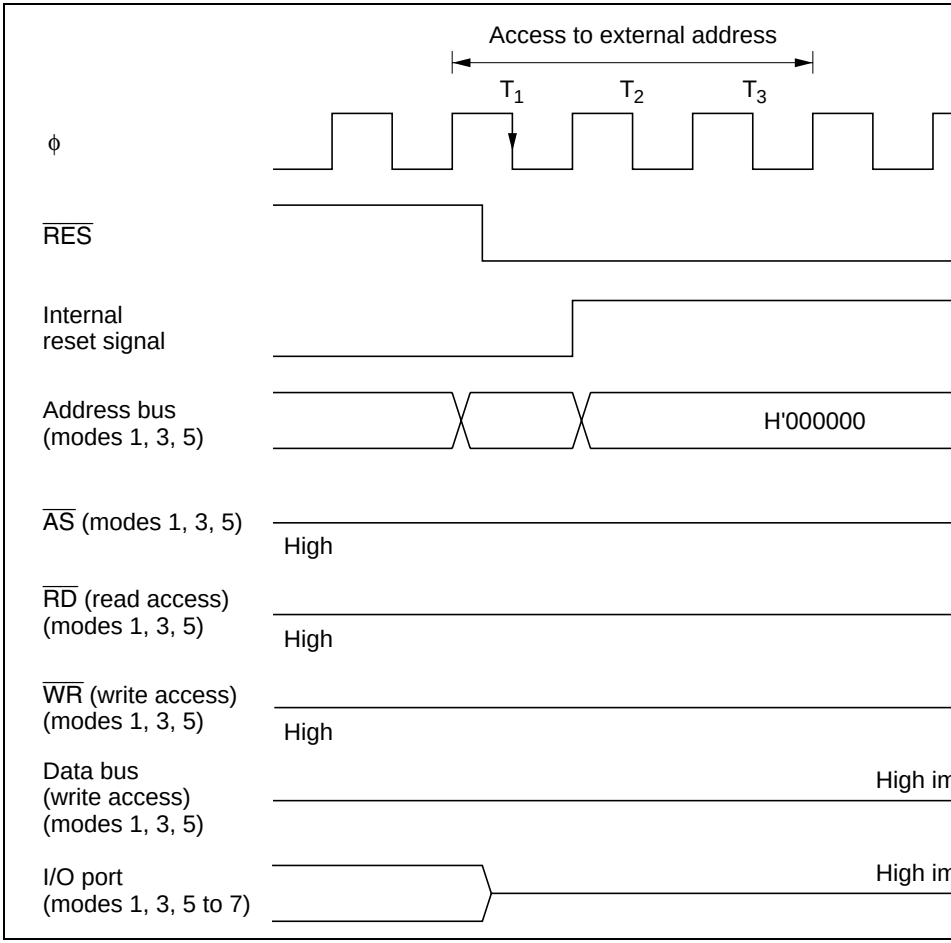


Figure D.1 Reset during Memory Access (Reset during T1 State)

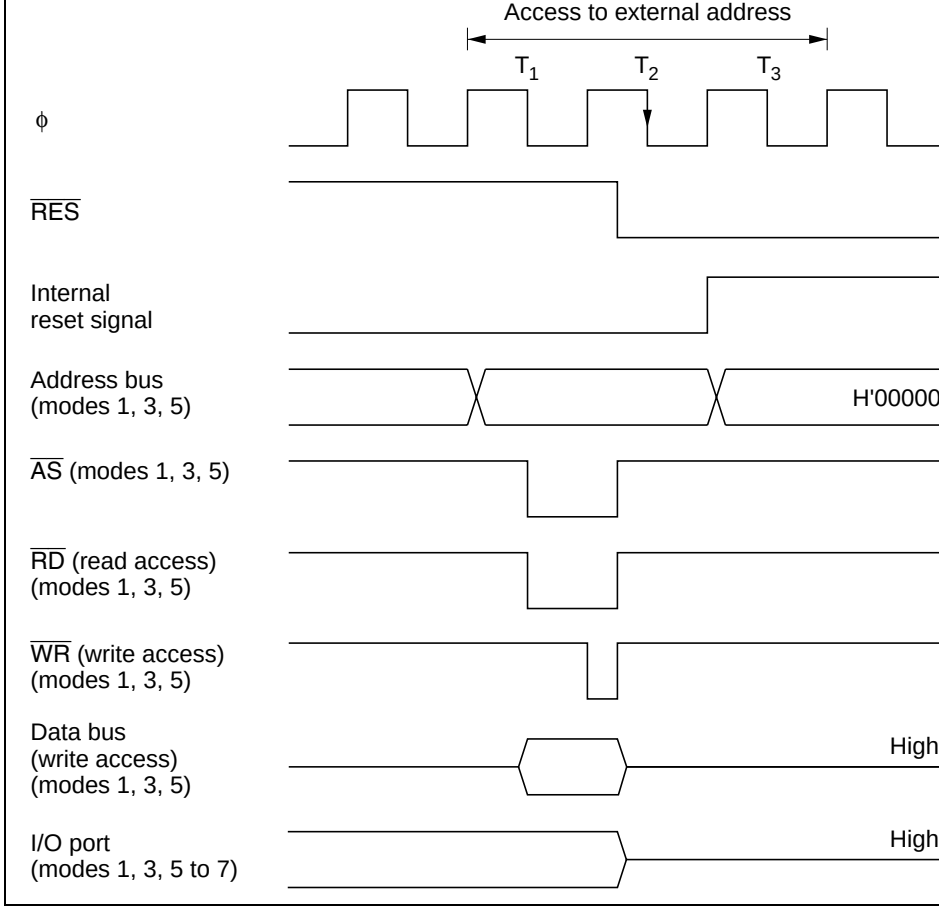


Figure D.2 Reset during Memory Access (Reset during T2 State)

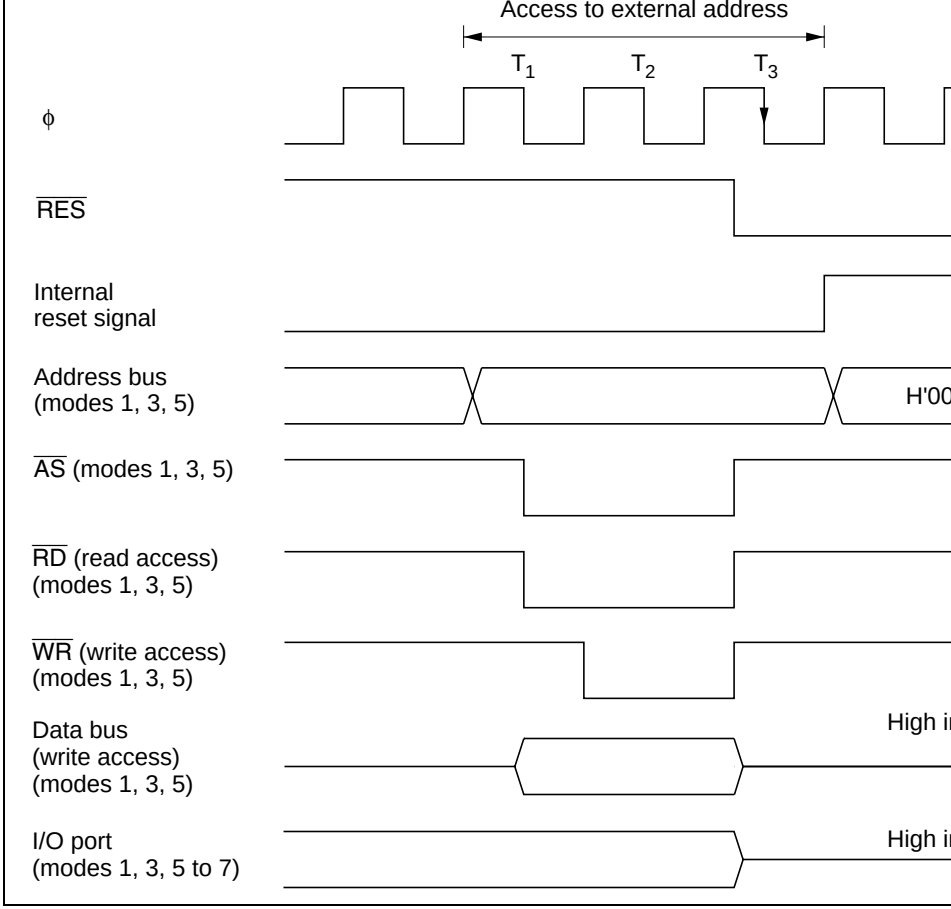
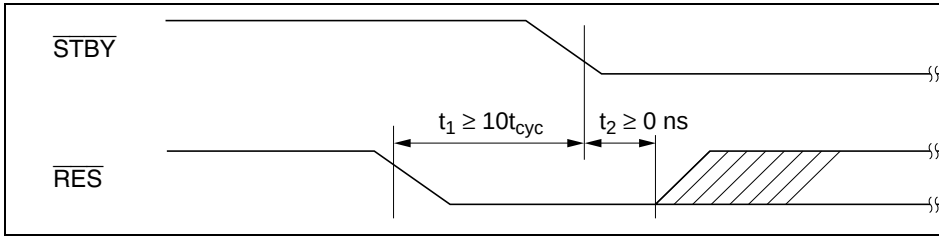


Figure D.3 Reset during Memory Access (Reset during T3 State)

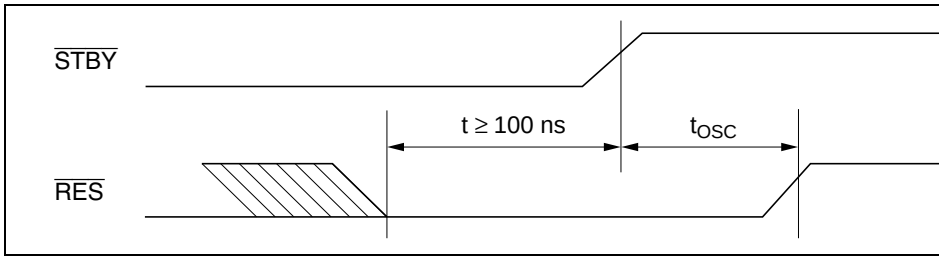
until STBY goes low (minimum delay from STBY low to RES high: 0 ns).



(2) To retain RAM contents with the RAME bit cleared to 0 in SYSCR, RES does not driven low as in (1).

Timing of Recovery from Hardware Standby Mode

Drive the RES signal low approximately 100 ns before STBY goes high.



		3 V version	HD64F3039VF	HD64F3039VF	80-pin QFP
			HD64F3039VTE	HD64F3039VTE	80-pin TQFI
	Mask ROM version	5 V version	HD6433039F	HD6433039(***)F	80-pin QFP
			HD6433039TE	HD6433039(***)TE	80-pin TQFI
		3 V version	HD6433039VF	HD6433039(***)VF	80-pin QFP
			HD6433039VTE	HD6433039(***)VTE	80-pin TQFI
H8/3038	Mask ROM version	5 V version	HD6433038F	HD6433038(***)F	80-pin QFP
			HD6433038TE	HD6433038(***)TE	80-pin TQFI
		3 V version	HD6433038VF	HD6433038(***)VF	80-pin QFP
			HD6433038VTE	HD6433038(***)VTE	80-pin TQFI
H8/3037	Mask ROM version	5 V version	HD6433037F	HD6433037(***)F	80-pin QFP
			HD6433037TE	HD6433037(***)TE	80-pin TQFI
		3 V version	HD6433037VF	HD6433037(***)VF	80-pin QFP
			HD6433037VTE	HD6433037(***)VTE	80-pin TQFI
H8/3036	Mask ROM version	5 V version	HD6433036F	HD6433036(***)F	80-pin QFP
			HD6433036TE	HD6433036(***)TE	80-pin TQFI
		3 V version	HD6433036VF	HD6433036(***)VF	80-pin QFP
			HD6433036VTE	HD6433036(***)VTE	80-pin TQFI

Note: (***) in mask ROM versions is the ROM code.

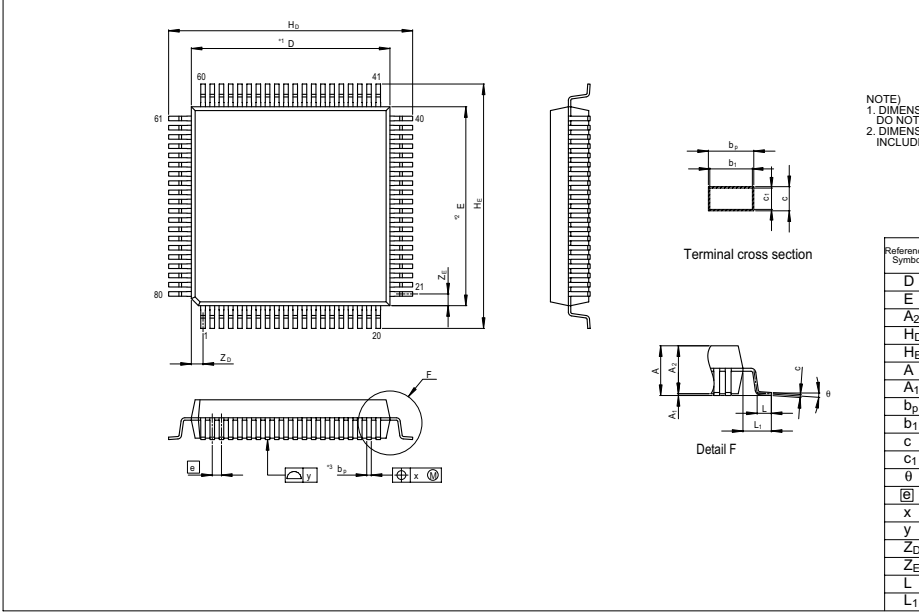
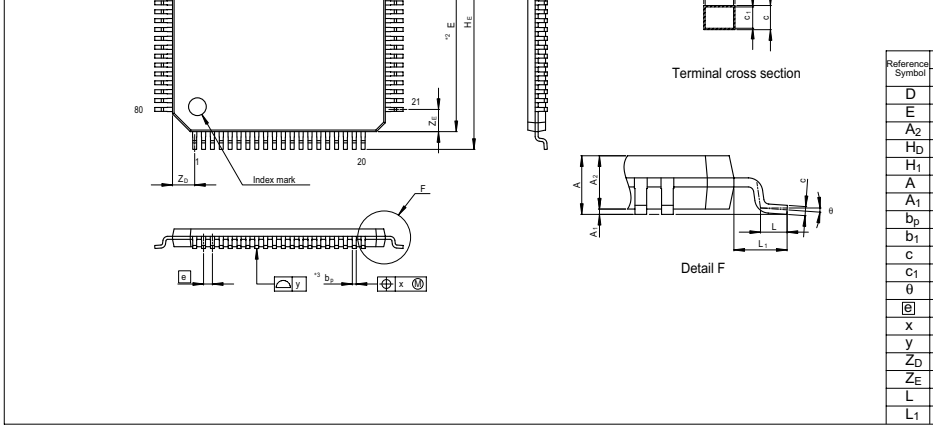


Figure G.1 Package Dimensions (FP-80A)



Reference Symbol
D
E
A_2
H_D
H_1
A
A_1
b_p
b_1
c
c_1
θ
e
x
y
Z_D
Z_E
L
L_1

Figure G.2 Package Dimensions (TFP-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,

RENESAS SALES OFFICES

http://www.re

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 Ehsar
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

X-ON Electronics

Largest Supplier of Electrical and Electronic Components

Click to view similar products for 8-bit Microcontrollers - MCU ***category:***

Click to view products by Renesas ***manufacturer:***

Other Similar products are found below :

CY8C20524-12PVXIT CY8C28433-24PVXIT MB95F012KPFT-G-SNE2 MB95F013KPMC-G-SNE2 MB95F263KPF-G-SNE2 MB95F264KPFT-G-SNE2 MB95F398KPMC-G-SNE2 MB95F478KPMC2-G-SNE2 MB95F562KPF-G-SNE2 MB95F564KPF-G-SNE2 MB95F634KPMC-G-SNE2 MB95F636KWQN-G-SNE1 MB95F696KPMC-G-SNE2 MB95F698KPMC1-G-SNE2 MB95F698KPMC2-G-SNE2 MB95F698KPMC-G-SNE2 MB95F818KPMC1-G-SNE2 MC908JK1ECDWER MC9S08PA32AVLD MC9S08PT60AVLD R5F1076CMSPV0 R5F5631ECDFBV0 C8051F389-B-GQ C8051F392-A-GMR ISD-ES1600_USB_PROG 901015X SC705C8AE0VFBE STM8TL53G4U6 PIC16F877-04/P-B R5F10Y17ASP#30 CY8C3MFIDOCK-125 403708R MB95F354EPF-G-SNE2 MB95F564KPFT-G-SNE2 MB95F564KWQN-G-SNE1 MB95F636KP-G-SH-SNE2 MB95F636KPMC-G-SNE2 MB95F694KPMC-G-SNE2 MB95F778JPMC1-G-SNE2 MB95F818KPMC-G-SNE2 MC908QY8CDWER MC9S08PT16AVLD MC9S08PT32AVLH MC9S08PT60AVLC MC9S08PT60AVLH C8051F500-IQR LC87F0G08AUJA-AH CP8361BT STM8S207C6T3 CG8421AF