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.

Accordingly, although Hitachi, Hitachi, Ltd., Hitachi Semiconductors, and other Hitachi b
names are mentioned in the document, these names have in fact all been changed to Renes
Technology Corp. Thank you for your understanding. Except for our corporate trademark,
corporate statement, no changes whatsoever have been made to the contents of the docume
these changes do not constitute any alteration to the contents of the document itself.

Renesas Technology Home Page: http://www.renesas.com

Renesas Techno
Customer Suppo
April 1, 2003

RENESAS
Renesas Technology Corp.

Notes regarding these materials

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

User's Manual

ADE-602-081

H8/3867 Series
H8/3827 Series

Hardware Manual

H8/3867 HD6473867, HD6433867
H8/3866 HD6433866
H8/3865 HD6433865
H8/3864 HD6433864
H8/3863 HD6433863
H8/3862 HD6433862
H8/3827 HD6473827, HD6433827
H8/3826 HD6433826
H8/3825 HD6433825
H8/3824 HD6433824
H8/3823 HD6433823
H8/3822 HD6433822

Renesas Electronics
www.renesas.com

Rev.3 1999.02

have received the latest product standards or specifications before final design, pur
use.

3. Hitachi makes every attempt to ensure that its products are of high quality and relia
However, contact Hitachi's sales office before using the product in an application t
demands especially high quality and reliability or where its failure or malfunction
threaten human life or cause risk of bodily injury, such as aerospace, aeronautics, n
power, combustion control, transportation, traffic, safety equipment or medical equ
life support.
4. Design your application so that the product is used within the ranges guaranteed by
particularly for maximum rating, operating supply voltage range, heat radiation cha
installation conditions and other characteristics. Hitachi bears no responsibility for
damage when used beyond the guaranteed ranges. Even within the guaranteed ran
consider normally foreseeable failure rates or failure modes in semiconductor devi
employ systemic measures such as fail-safes, so that the equipment incorporating H
product does not cause bodily injury, fire or other consequential damage due to op
the Hitachi product.
5. This product is not designed to be radiation resistant.
6. No one is permitted to reproduce or duplicate, in any form, the whole or part of thi
without written approval from Hitachi.
7. Contact Hitachi's sales office for any questions regarding this document or Hitachi
semiconductor products.

	supply	
5	Figure 1.1 Block Diagram	Modification
8	Table 1.2 Pin Functions / Power source pin	Modification of stabilization cap
13	2.1.1 Features	Change in High-speed operatio
88	Figure 4.2 Typical Connection to Crystal Oscillator	Addition of recommended value
89	Figure 4.4 Typical Connection to Ceramic Oscillator	Addition of recommended value
91	Figure 4.7 Typical Connection to 32.768-kHz/38.4 kHz Crystal Oscillator (Subclock)	Addition of description
92	Figure 4.9 Pin Connection when not Using Subclock	Modification
95	Table 5.1 Operating Modes	Modification of subsleep mode/ descriptions
97	Table 5.2 Internal State in Each Operating Mode	Modification of Note 4
99	5.1.1 System Control Registers 1. System control register 1 (SYSCR1)	Addition of Notes to Bits 6 to 4
100	2. System control register 2 (SYSCR2)	Modification of Bit 4 contents
102	5.2 Sleep Mode	Addition of description
104	5.3.3 Oscillator Settling Timer after Stanby Mode is Cleared	Addition of description
107	5.5.2 Clearing Subsleep Mode • Clearing by interrupt	Addition of description
109	5.7.1 Transition to Active (Medium-Speed) Mode	Addition of description
147	Table 8.10 Port 3 Pin States	Modification
226	Table 9.13 Timer G Operation Modes	Addition of description in Notes
248	9.7.5 Application Notes	Addition and modification of des
251	Figure 10.1 SCI3 Block Diagram	Modification

		and 5
303	10.5 Application Notes	Addition of 9 and 10.
357	14.2 When Using the Internal Power Supply Step-Down Circuit	Modification of description
360 to 362	15.2.1 Power Supply Voltage and Operating Range	Modification of 1 to 3
363 to 368	Table 15.2 DC Characteristics	Addition and modification
369 to 370	Table 15.3 Control Signal Timing	Addition and modification
371	Table 15.4 Serial Interface (SCI3-1, SCI3-2) Timing	Addition of Notes
372	Table 15.5 A/D Converter Characteristics	Modification
373	Table 15.7 AC Characteristics for External Segment Expansion	Addition
376	Figure 15.6 SCI-3 Synchronous Mode Input/Output Timing	Modification of Notes
377	Figure 15.7 Segment Expansion Signal Timing	Addition

serial communication interface, and an A/D converter. This allows H8/3867 Series de
used as embedded microcomputers in systems requiring LCD display.

The H8/3867 Series incorporates an LCD drive power supply and step-up constant po

(5 V), enabling a fixed 5 V voltage to be obtained independently of V_{CC}.

This manual describes the hardware of the H8/3867 Series and H8/3827 Series. For d
H8/3864 Series instruction set, refer to the H8/300L Series Programming Manual.

microcomputers equipped with a controller/driver. Other on-chip peripheral functions timers, a 14-bit pulse width modulator (PWM), two serial communication interface ch an A/D converter. Together, these functions make the H8/3864 Series ideally suited f applications in systems requiring low power consumption and LCD display. Models i H8/3867 and H8/3827 Series are the H8/3862 and H8/3822, with on-chip 16-kbyte RO kbyte RAM, the H8/3863 and H8/3823, with 24-kbyte ROM and 1-kbyte RAM, the H H8/3824, with 32-kbyte ROM and 2-kbyte RAM, the H8/3865 and H8/3825, with 40- and 2-kbyte RAM, the H8/3866 and H8/3826, with 48-kbyte ROM and 2-kbyte RAM, H8/3867 and H8/3827, with 60-kbyte ROM and 2-kbyte RAM.

The H8/3867 and H8/3827 are also available in a ZTAT™* version with on-chip PRO can be programmed as required by the user.

Table 1.1 summarizes the features of the H8/3867 Series and H8/3827 Series.

Note: * ZTAT (Zero Turn Around Time) is a trademark of Hitachi, Ltd.

	— Max. operating speed: 6 MHz — Add/subtract: 0.67 µs (operating at 3 MHz) — Multiply/divide: 4.67 µs (operating at 3 MHz) — Can run on 32.768 kHz or 38.4 kHz subclock • Instruction set compatible with H8/300 CPU — Instruction length of 2 bytes or 4 bytes — Basic arithmetic operations between registers — MOV instruction for data transfer between memory and • Typical instructions — Multiply (8 bits × 8 bits) — Divide (16 bits ÷ 8 bits) — Bit accumulator — Register-indirect designation of bit position
Interrupts	36 interrupt sources • 13 external interrupt sources (IRQ_4 to IRQ_0, WKP_7 to WKP • 23 internal interrupt sources
Clock pulse generators	Two on-chip clock pulse generators • System clock pulse generator: 0.4 to 6 MHz • Subclock pulse generator: 32.768 kHz, 38.4 kHz
Power-down modes	Seven power-down modes • Sleep (high-speed) mode • Sleep (medium-speed) mode • Standby mode • Watch mode • Subsleep mode • Subactive mode • Active (medium-speed) mode

	• H8/3866, H8/3826: 48-kbyte ROM, 2-kbyte RAM • H8/3867, H8/3827: 60-kbyte ROM, 2-kbyte RAM
I/O ports	64 pins • 55 I/O pins • 9 input pins
Timers	Six on-chip timers • Timer A: 8-bit timer Count-up timer with selection of eight internal clock signal from the system clock (ø)* and four clock signals divided watch clock ($ø_w$)* • Asynchronous event counter: 16-bit timer — Count-up timer able to count asynchronous external e independently of the MCU's internal clocks • Timer C: 8-bit timer — Count-up/down timer with selection of seven internal c or event input from external pin — Auto-reloading • Timer F: 16-bit timer — Can be used as two independent 8-bit timers — Count-up timer with selection of four internal clock sig input from external pin — Provision for toggle output by means of compare-matc • Timer G: 8-bit timer — Count-up timer with selection of four internal clock sig — Incorporates input capture function (built-in noise canc • Watchdog timer — Reset signal generated by overflow of 8-bit counter

Note: * See section 4, Clock Pulse Generator, for the definition of ø and $ø_w$.

14-bit PWM	Pulse-division PWM output for reduced ripple • Can be used as a 14-bit D/A converter by connecting to an low-pass filter.
A/D converter	Successive approximations using a resistance ladder • 8-channel analog input pins • Conversion time: 31/ø or 62/ø per channel
LCD controller/driver	LCD controller/driver equipped with a maximum of 32 segment four common pins • Choice of four duty cycles (static, 1/2, 1/3, or 1/4) • Segment pins can be switched to general-purpose port fun bit units
LCD drive power supply	Step-up constant-voltage power supply allows LCD display (H8/3867 Series only)

HD6433822F			
HD6433862W, HD6433822W	—	80-pin TQFP (TFP-80C)	
HD6433863H, HD6433823H	—	80-pin QFP (FP-80A)	ROM RAM
HD6433863F, HD6433823F	—	80-pin QFP (FP-80B)	
HD6433863W, HD6433823W	—	80-pin TQFP (TFP-80C)	
HD6433864H, HD6433824H	—	80-pin QFP (FP-80A)	ROM RAM
HD6433864F, HD6433824F	—	80-pin QFP (FP-80B)	
HD6433864W, HD6433824W	—	80-pin TQFP (TFP-80C)	
HD6433865H, HD6433825H	—	80-pin QFP (FP-80A)	ROM RAM
HD6433865F, HD6433825F	—	80-pin QFP (FP-80B)	
HD6433865W, HD6433825W	—	80-pin TQFP (TFP-80C)	
HD6433866H, HD6433826H	—	80-pin QFP (FP-80A)	ROM RAM
HD6433866F, HD6433826F	—	80-pin QFP (FP-80B)	
HD6433866W, HD6433826W	—	80-pin TQFP (TFP-80C)	
HD6433867H, HD6433827H	HD6473867H, HD6473827H	80-pin QFP (FP-80A)	ROM RAM
HD6433867F, HD6433827F	HD6473867F, HD6473827F	80-pin QFP (FP-80B)	
HD6433867W, HD6433827W	HD6473867W, HD6473827W	80-pin TQFP (TFP-80C)	

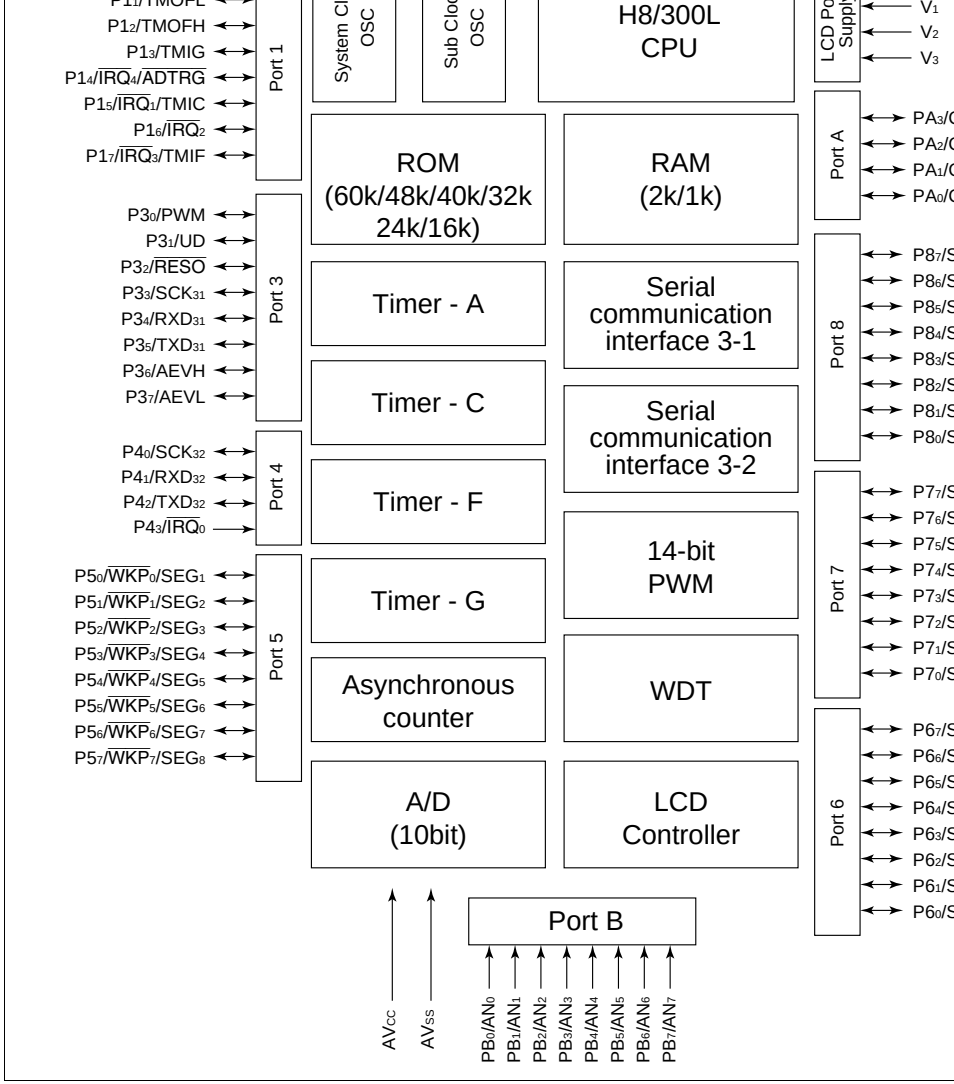


Figure 1.1 Block Diagram

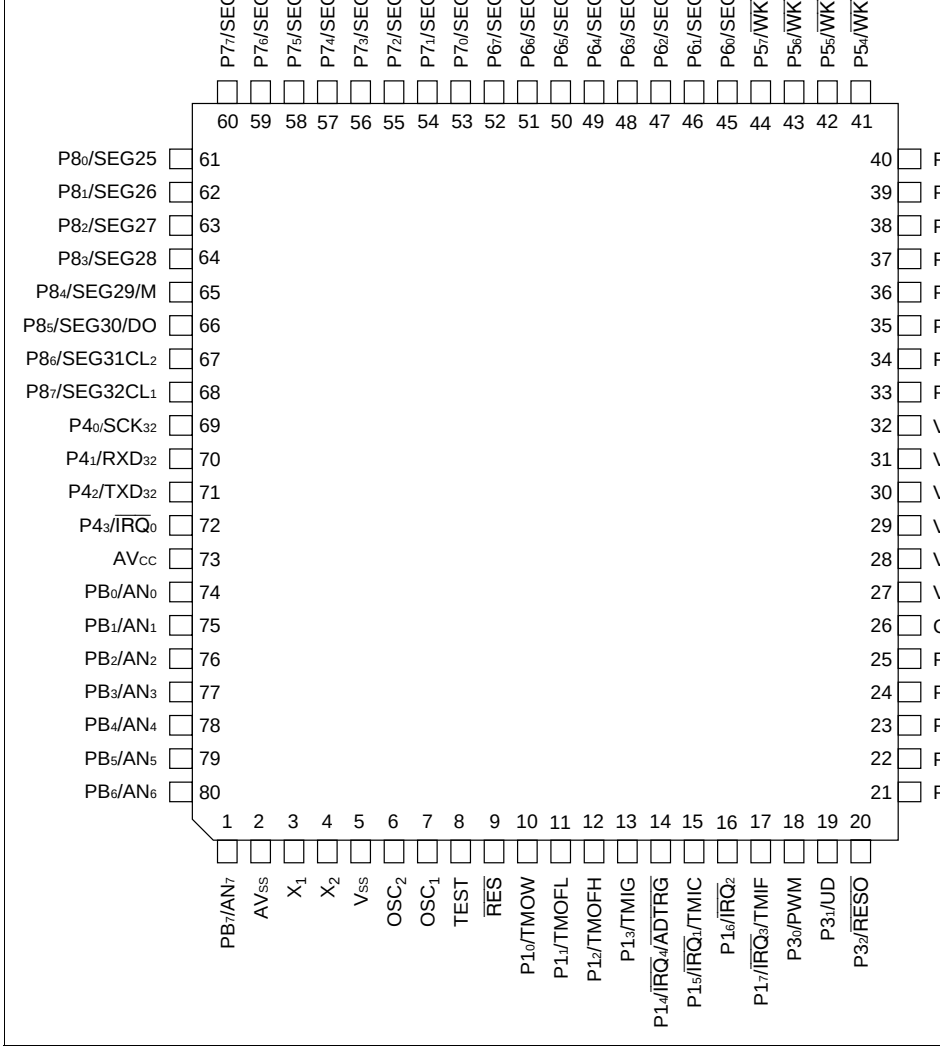


Figure 1.2 Pin Arrangement (FP-80A, TFP-80C: Top View)

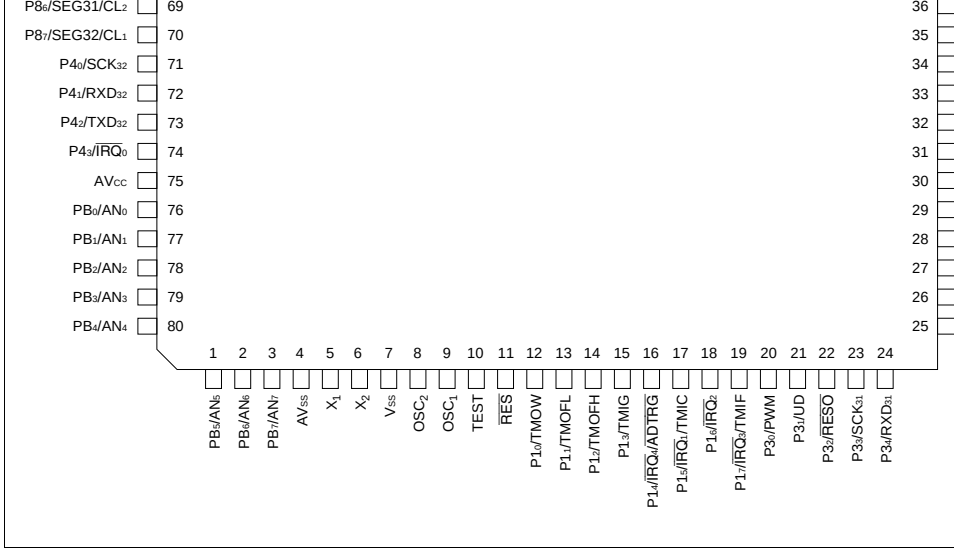


Figure 1.3 Pin Arrangement (FP-80B: Top View)

Power source pins	V_{CC} CV_{CC}	32 26	34 28	Input	**Power supply:** All V_{CC} pins connected to the system po See section 14, Power Supp
	V_{SS}	5 27	7 29	Input	**Ground:** All V_{SS} pins shoul connected to the system po (0 V).
	AV_{CC}	73	75	Input	**Analog power supply:** This power supply pin for the A/D When the A/D converter is n connect this pin to the syste supply.
	AV_{SS}	2	4	Input	**Analog ground:** This is the converter ground pin. It sho connected to the system po (0V).
	V_0	31	33	Output	**LCD power supply:** These
	V_1 V_2 V_3	30 29 28	32 31 30	Input	power supply pins for the LC controller/driver. They incor power supply split-resistanc normally used with V_0 and V
Clock pins	OSC_1	7	9	Input	These pins connect to a cry
	OSC_2	6	8	Output	ceramic oscillator, or can be input an external clock. See Clock Pulse Generators, fo connection diagram.
	X_1	3	5	Input	These pins connect to a 32.
	X_2	4	6	Output	38.4-kHz crystal oscillator. See section 4, Clock Pulse Generators, for a typical cor diagram.
System control	$\overline{RES}$	9	11	Input	**Reset:** When this pin is driv chip is reset
	$\overline{RESO}$	20	22	Output	**Reset output:** Outputs the reset signal.

pins	$\overline{IRQ}_1$	15	17		are input pins for edge-sensit
	$\overline{IRQ}_2$	16	18		external interrupts, with a sel
	$\overline{IRQ}_3$	17	19		rising or falling edge
	$\overline{IRQ}_4$	14	16		
	$\overline{WKP}_7$ to $\overline{WKP}_0$	44 to 37	46 to 39	Input	**Wakeup interrupt request 0** These are input pins for rising edge-sensitive external interr
Timer pins	TMOW	10	12	Output	**Clock output:** This is an outp waveforms generated by the output circuit.
	AEVL	25	27	Input	**Asynchronous event count**
	AEVH	24	26		**input:** This is an event input input to the asynchronous ev counter.
	TMIC	15	17	Input	**Timer C event input:** This is input pin for input to the timer
	UD	19	21	Input	**Timer C up/down select:** Th selects up- or down-counting timer C counter. The counter as an up-counter when this p and as a down-counter when
	TMIF	17	19	Input	**Timer F event input:** This is input pin for input to the timer
	TMOFL	11	13	Output	**Timer FL output:** This is an for waveforms generated by t FL output compare function.
	TMOFH	12	14	Output	**Timer FH output:** This is an for waveforms generated by t FH output compare function.
	TMIG	13	15	Input	**Timer G capture input:** This pin for timer G input capture.

		80 to 76		
$P4_3$	72	74	Input	**Port 4 (bit 3):** This is a 1-bi
$P4_2$ to $P4_0$	71 to 69	73 to 71	I/O	**Port 4 (bits 2 to 0):** This is port. Input or output can be for each bit by means of por register 4 (PCR4).
PA_3 to PA_0	33 to 36	35 to 38	I/O	**Port A:** This is a 4-bit I/O po output can be designated fo means of port control regist
$P1_7$ to $P1_0$	17 to 10	19 to 12	I/O	**Port 1:** This is an 8-bit I/O p output can be designated fo means of port control regist
$P3_7$ to $P3_0$	25 to 18	27 to 20	I/O	**Port 3:** This is an 8-bit I/O p output can be designated fo means of port control regist
$P5_7$ to $P5_0$	44 to 37	46 to 39	I/O	**Port 5:** This is an 8-bit I/O p output can be designated fo means of port control regist
$P6_7$ to $P6_0$	52 to 45	54 to 47	I/O	**Port 6:** This is an 8-bit I/O p output can be designated fo means of port control regist
$P7_7$ to $P7_0$	60 to 53	62 to 55	I/O	**Port 7:** This is an 8-bit I/O p output can be designated fo means of port control regist
$P8_7$ to $P8_0$	68 to 61	70 to 63	I/O	**Port 8:** This is an 8-bit I/O p output can be designated fo means of port control regist

(SCI)	SCK_{31}	21	23	I/O	**SCI3-1 clock I/O:** This is the SCI31 clock I/O pi
	RXD_{32}	70	72	Input	**SCI3-2 receive data input:** This is the SCI32 data input p
	TXD_{32}	71	73	Output	**SCI3-2 transmit data output** This is the SCI32 data output
	SCK_{32}	69	71	I/O	**SCI3-2 clock I/O:** This is the SCI32 clock I/O pi
A/D converter	AN7 to An0	1 80 to 74	3 to 1 80 to 76	Input	**Analog input channels 7 to** These are analog data input the A/D converter
	$\overline{ADTRG}$	14	16	Input	**A/D converter trigger input** This is the external trigger inp the A/D converter
LCD controller/ driver	COM_4 to COM_1	33 to 36	35 to 38	Output	**LCD common output:** Thes LCD common output pins.
	SEG_{32} to SEG_1	68 to 37	70 to 39	Output	**LCD segment output:** Thes LCD segment output pins.
	CL1	68	70	Output	**LCD latch clock:** This is the for the segment external expa display data latch clock.
	CL2	67	69	Output	**LCD shift clock:** This is the for the segment external expa display data shift clock.
	DO	66	68	Output	**LCD serial data output:** This output pin for segment extern expansion serial display data
	M	65	67	Output	**LCD alternation signal:** This output pin for the segment ex expansion LCD alternation si

Features of the H8/300L CPU are listed below.

- General-register architecture
  Sixteen 8-bit general registers, also usable as eight 16-bit general registers
- Instruction set with 55 basic instructions, including:
  — Multiply and divide instructions
  — Powerful bit-manipulation instructions
- Eight addressing modes
  — Register direct
  — Register indirect
  — Register indirect with displacement
  — Register indirect with post-increment or pre-decrement
  — Absolute address
  — Immediate
  — Program-counter relative
  — Memory indirect
- 64-kbyte address space
- High-speed operation
  — All frequently used instructions are executed in two to four states
  — High-speed arithmetic and logic operations
  — 8- or 16-bit register-register add or subtract: 0.67 µs*
  — 8 × 8-bit multiply: 4.67 µs*
  — 16 ÷ 8-bit divide: 4.67 µs*

Note: * These values are at ø = 3 MHz.

- Low-power operation modes
  SLEEP instruction for transfer to low-power operation

Figure 2.1 shows the register structure of the H8/300L CPU. There are two groups of re
general registers and control registers.

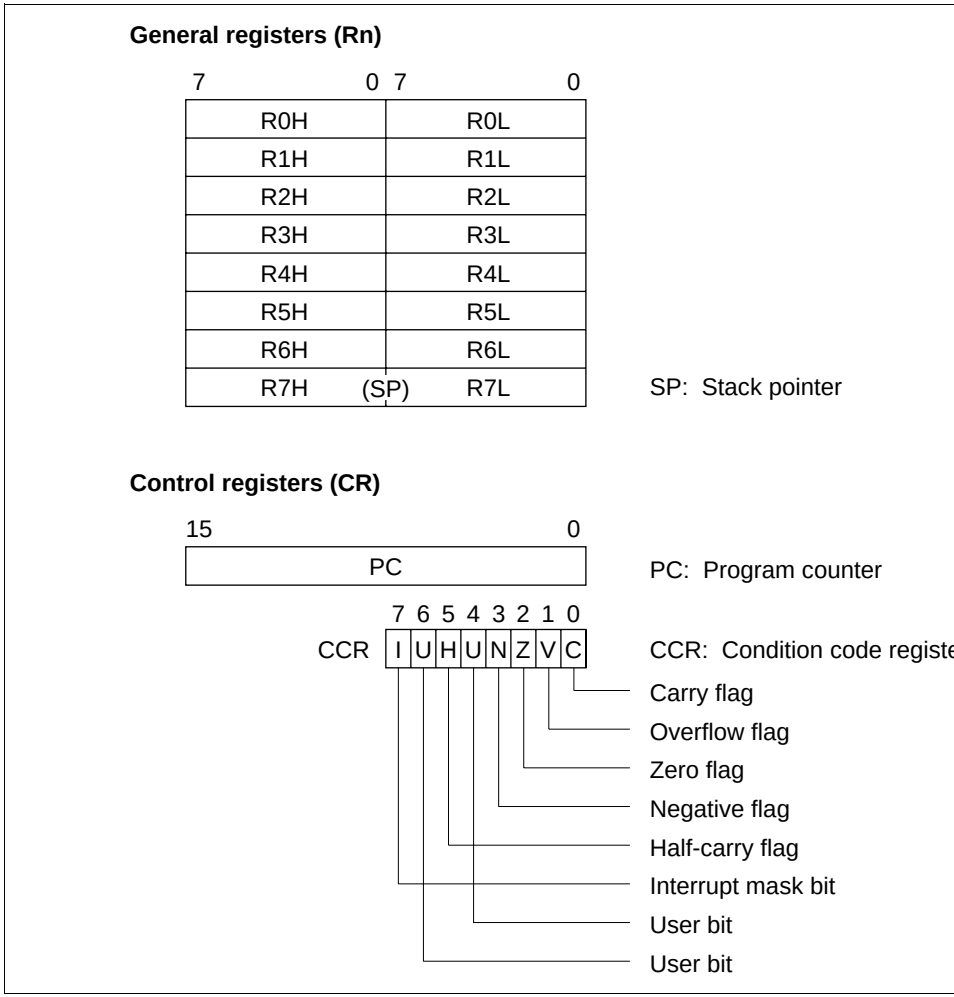


Figure 2.1 CPU Registers

When used as address registers, the general registers are accessed as 16-bit registers (E

R7 also functions as the stack pointer (SP), used implicitly by hardware in exception p
and subroutine calls. When it functions as the stack pointer, as indicated in figure 2.2,
points to the top of the stack.

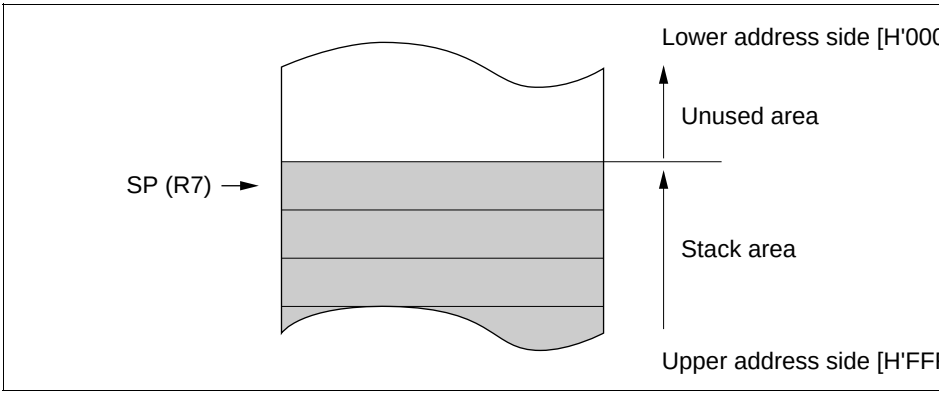


Figure 2.2 Stack Pointer

2.2.2 Control Registers

The CPU control registers include a 16-bit program counter (PC) and an 8-bit conditio
register (CCR).

Program Counter (PC): This 16-bit register indicates the address of the next instruct
will execute. All instructions are fetched 16 bits (1 word) at a time, so the least signifi
the PC is ignored (always regarded as 0).

Condition Code Register (CCR): This 8-bit register contains internal status informat
including the interrupt mask bit (I) and half-carry (H), negative (N), zero (Z), overflow
carry (C) flags. These bits can be read and written by software (using the LDC, STC,
ORC, and XORC instructions). The N, Z, V, and C flags are used as branching condit
conditional branching (Bcc) instructions.

The H flag is used implicitly by the DAA and DAS instructions.

When the ADD.W, SUB.W, or CMP.W instruction is executed, the H flag is set to 1 if carry or borrow at bit 11, and is cleared to 0 otherwise.

Bit 4—User Bit (U): Can be used freely by the user.

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

Bit 2—Zero Flag (Z): Set to 1 to indicate a zero result, and cleared to 0 to indicate a n result.

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

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

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

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

Some instructions leave some or all of the flag bits unchanged.

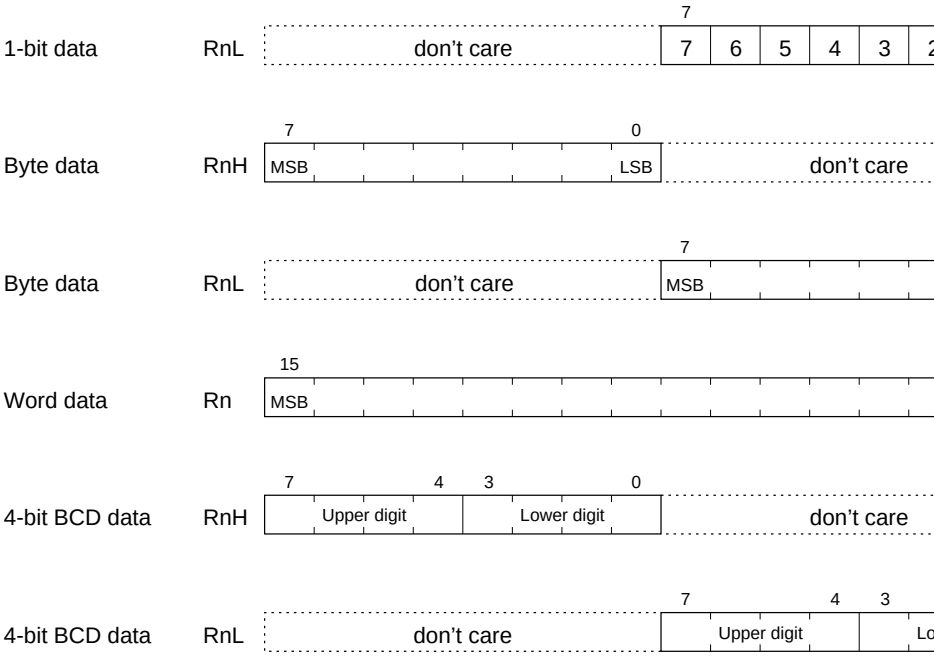
Refer to the H8/300L Series Programming Manual for the action of each instruction on bits.

2.2.3 Initial Register Values

When the CPU is reset, the program counter (PC) is initialized to the value stored at ad H'0000 in the vector table, and the I bit in the CCR is set to 1. The other CCR bits and t registers are not initialized. In particular, the stack pointer (R7) is not initialized. The st should be initialized by software, by the first instruction executed after a reset.

DIVXU (16 bits ÷ 8 bits) instructions operate on word data.

- The DAA and DAS instructions perform decimal arithmetic adjustments on byte d packed BCD form. Each nibble of the byte is treated as a decimal digit.



Notation:
RnH: Upper byte of general register
RnL: Lower byte of general register
MSB: Most significant bit
LSB: Least significant bit

Figure 2.3 Register Data Formats

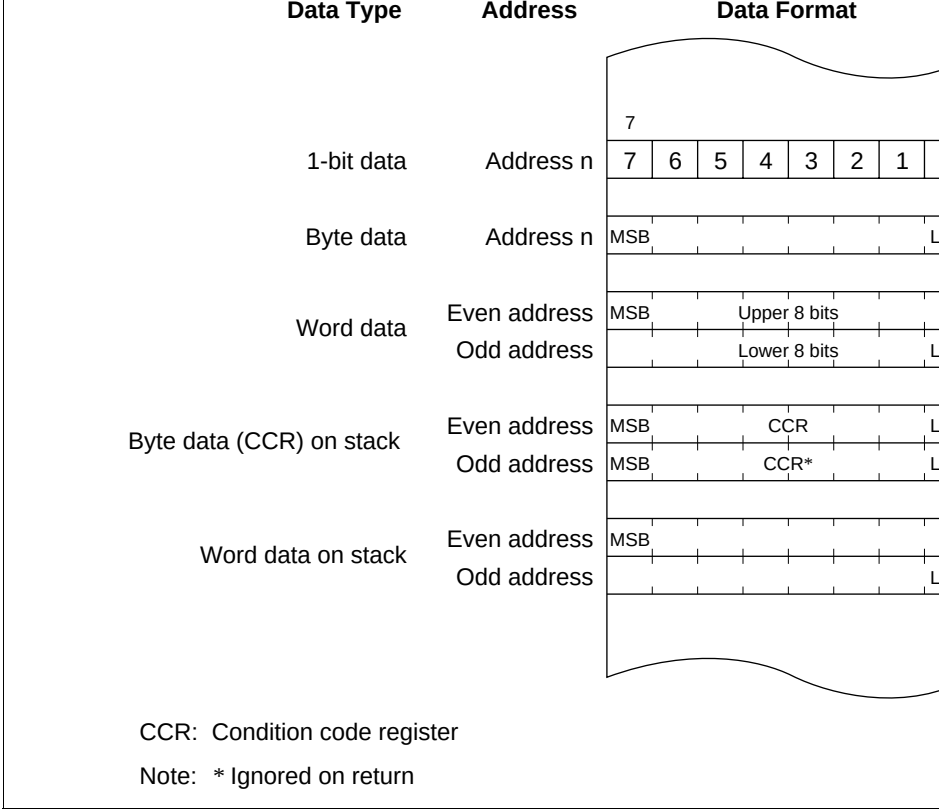


Figure 2.4 Memory Data Formats

When the stack is accessed using R7 as an address register, word access should alway performed. When the CCR is pushed on the stack, two identical copies of the CCR are make a complete word. When they are restored, the lower byte is ignored.

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

1. **Register Direct—Rn:** The register field of the instruction specifies an 8- or 16-bit g register containing the operand.

   Only the MOV.W, ADD.W, SUB.W, CMP.W, ADDS, SUBS, MULXU (8 bits × 8 DIVXU (16 bits ÷ 8 bits) instructions have 16-bit operands.
2. **Register Indirect—@Rn:** The register field of the instruction specifies a 16-bit ger register containing the address of the operand in memory.
3. **Register Indirect with Displacement—@(d:16, Rn):** The instruction has a second (bytes 3 and 4) containing a displacement which is added to the contents of the spec general register to obtain the operand address in memory.

   This mode is used only in MOV instructions. For the MOV.W instruction, the result must be even.

The @–Rn mode is used with MOV instructions that store register contents to

The register field of the instruction specifies a 16-bit general register which is
by 1 or 2 to obtain the address of the operand in memory. The register retains t
decremented value. The size of the decrement is 1 for MOV.B or 2 for MOV.W
MOV.W, the original contents of the register must be even.

5. **Absolute Address—@aa:8 or @aa:16:** The instruction specifies the absolute add
operand in memory.

   The absolute address may be 8 bits long (@aa:8) or 16 bits long (@aa:16). The MO
manipulation instructions can use 8-bit absolute addresses. The MOV.B, MOV.W,
JSR instructions can use 16-bit absolute addresses.

   For an 8-bit absolute address, the upper 8 bits are assumed to be 1 (H'FF). The add
H'FF00 to H'FFFF (65280 to 65535).

6. **Immediate—#xx:8 or #xx:16:** The instruction contains an 8-bit operand (#xx:8) i
byte, or a 16-bit operand (#xx:16) in its third and fourth bytes. Only MOV.W instr
contain 16-bit immediate values.

   The ADDS and SUBS instructions implicitly contain the value 1 or 2 as immediate
bit manipulation instructions contain 3-bit immediate data in the second or fourth b
instruction, specifying a bit number.

7. **Program-Counter Relative—@(d:8, PC):** This mode is used in the Bcc and BSR
instructions. An 8-bit displacement in byte 2 of the instruction code is sign-extende
and added to the program counter contents to generate a branch destination address
possible branching range is –126 to +128 bytes (–63 to +64 words) from the curren
The displacement should be an even number.

8. **Memory Indirect—@@aa:8:** This mode can be used by the JMP and JSR instruc
second byte of the instruction code specifies an 8-bit absolute address. The word lo
address contains the branch destination address.

   The upper 8 bits of the absolute address are assumed to be 0 (H'00), so the address
from H'0000 to H'00FF (0 to 255). Note that with the H8/300L Series, the lower en
address area is also used as a vector area. See 3.3, Interrupts, for details on the vec

If an odd address is specified as a branch destination or as the operand address of a MO
instruction, the least significant bit is regarded as 0, causing word access to be perform
address preceding the specified address. See 2.3.2, Memory Data Formats, for further

Bit manipulation instructions can use register direct (1), register indirect (2), or 8-bit ab
addressing (5) to specify the operand. Register indirect (1) (BSET, BCLR, BNOT, and
instructions) or 3-bit immediate addressing (6) can be used independently to specify a b
in the operand.

Table 2.2 Effective Address Calculation

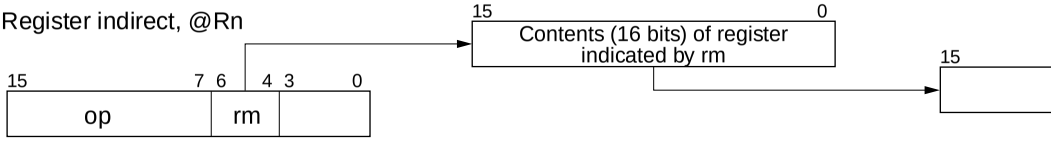
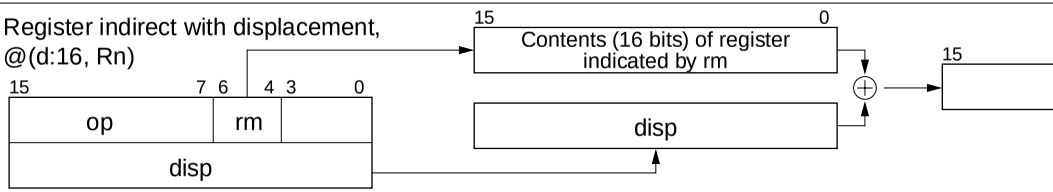
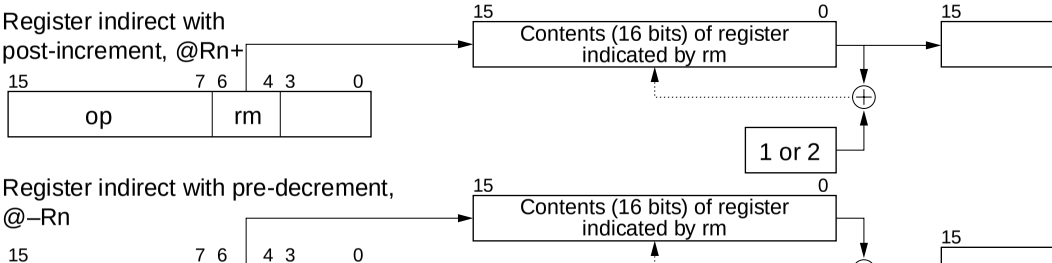
No.	Addressing Mode and Instruction Format	Effective Address Calculation Method	Effective A
1	Register direct, Rn		
2	Register indirect, @Rn		
3	Register indirect with displacement, @(d:16, Rn)		
4	Register indirect with post-increment, @Rn+ Register indirect with pre-decrement, @–Rn		

Table 2.2 Effective Address Calculation (cont)

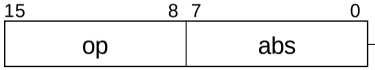
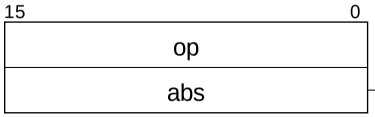
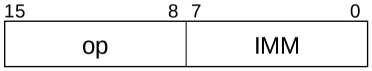
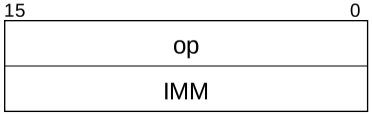
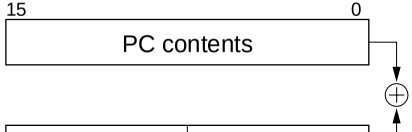
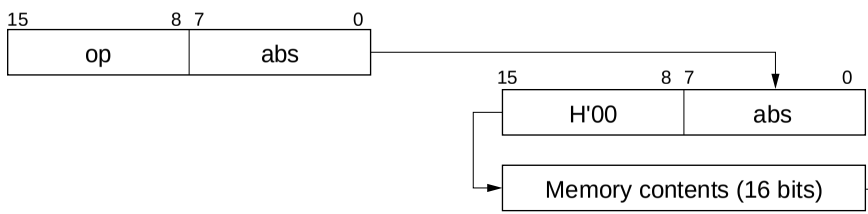
No.	Addressing Mode and Instruction Format	Effective Address Calculation Method	Effective
5	Absolute address @aa:8  @aa:16 		 
6	Immediate #xx:8  #xx:16 		Operand is 1
7	Program-counter relative @(d:8, PC)		

Table 2.2 Effective Address Calculation (cont)

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

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

	DAS, MULXU, DIVXU, CMP, NEG
Logic operations	AND, OR, XOR, NOT
Shift	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[*2], JMP, BSR, JSR, RTS
System control	RTE, SLEEP, LDC, STC, ANDC, ORC, XORC, NOP
Block data transfer	EEPMOV

Notes: 1. PUSH Rn is equivalent to MOV.W Rn, @–SP.
POP Rn is equivalent to MOV.W @SP+, Rn. The same applies to the machi
language.
2. Bcc is a conditional branch instruction in which cc represents a condition coc

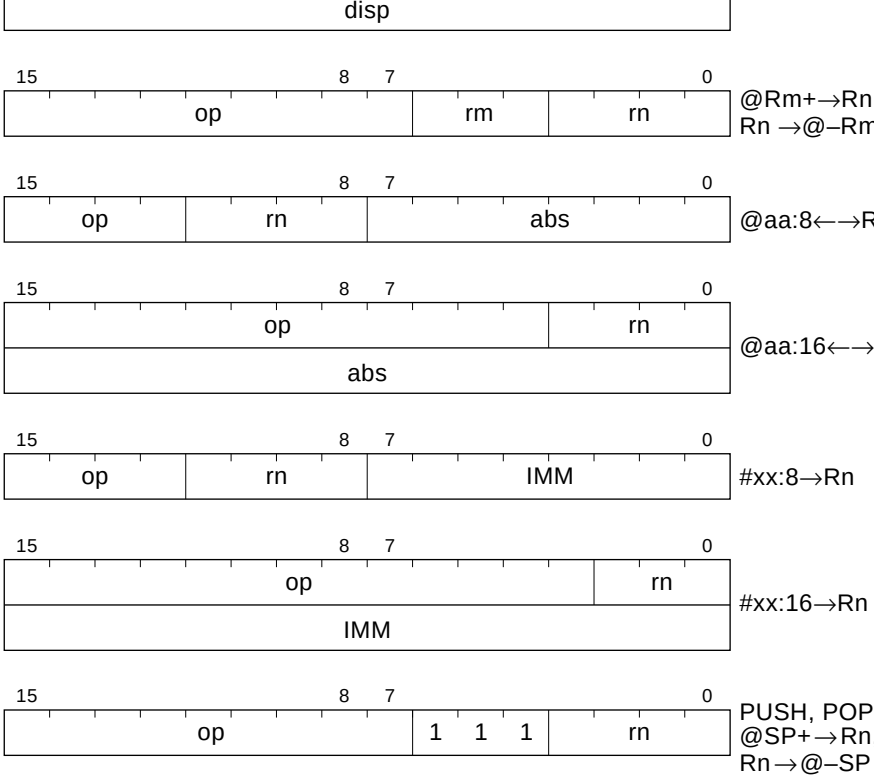
The following sections give a concise summary of the instructions in each category, and the bit patterns of their object code. The notation used is defined next.

N	N (negative) flag of CCR
Z	Z (zero) flag of CCR
V	V (overflow) flag of CCR
C	C (carry) flag of CCR
PC	Program counter
SP	Stack pointer
#IMM	Immediate data
disp	Displacement
+	Addition
–	Subtraction
×	Multiplication
÷	Division
∧	AND logical
∨	OR logical
⊕	Exclusive OR logical
→	Move
~	Logical negation (logical complement)
:3	3-bit length
:8	8-bit length
:16	16-bit length
( ), < >	Contents of operand indicated by effective address

		register and memory, or moves immediate data to a ger register. The Rn, @Rn, @(d:16, Rn), @aa:16, #xx:16, @–Rn, ar addressing modes are available for word data. The @a addressing mode is available for byte data only. The @–R7 and @R7+ modes require word operands. D specify byte size for these two modes.
POP	W	@SP+ → Rn Pops a 16-bit general register from the stack. Equivaler MOV.W @SP+, Rn.
PUSH	W	Rn → @–SP Pushes a 16-bit general register onto the stack. Equiva MOV.W Rn, @–SP.

Notes: * Size: Operand size
B: Byte
W: Word

Certain precautions are required in data access. See 2.9.1, Notes on Data Access, for de



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

Figure 2.5 Data Transfer Instruction Codes

		or addition on immediate data and data in a general reg Immediate data cannot be subtracted from data in a ger register. Word data can be added or subtracted only wh words are in general registers.
ADDX SUBX	B	Rd ± Rs ± C → Rd, Rd ± #IMM ± C → Rd Performs addition or subtraction with carry or borrow on in two general registers, or addition or subtraction on im data and data in a general register.
INC DEC	B	Rd ± 1 → Rd Increments or decrements a general register by 1.
ADDS SUBS	W	Rd ± 1 → Rd, Rd ± 2 → Rd Adds or subtracts 1 or 2 to or from a general register
DAA DAS	B	Rd decimal adjust → Rd Decimal-adjusts (adjusts to 4-bit BCD) an addition or su result in a general register by referring to the CCR
MULXU	B	Rd × Rs → Rd Performs 8-bit × 8-bit unsigned multiplication on data in general registers, providing a 16-bit result
DIVXU	B	Rd ÷ Rs → Rd Performs 16-bit ÷ 8-bit unsigned division on data in two registers, providing an 8-bit quotient and 8-bit remainde
CMP	B/W	Rd – Rs, Rd – #IMM Compares data in a general register with data in anothe register or with immediate data, and indicates the result CCR. Word data can be compared only between two g registers.
NEG	B	0 – Rd → Rd Obtains the two's complement (arithmetic complement) general register

Notes: * Size: Operand size
B: Byte
W: Word

		another general register or immediate data
OR	B	Rd ∨ Rs → Rd, Rd ∨ #IMM → Rd Performs a logical OR operation on a general register general register or immediate data
XOR	B	Rd ⊕ Rs → Rd, Rd ⊕ #IMM → Rd Performs a logical exclusive OR operation on a genera and another general register or immediate data
NOT	B	~ Rd → Rd Obtains the one's complement (logical complement) of register contents

Notes: * Size: Operand size
B: Byte

2.5.4 Shift Operations

Table 2.7 describes the eight shift instructions.

Table 2.7 Shift Instructions

Instruction	Size*	Function
SHAL SHAR	B	Rd shift → Rd Performs an arithmetic shift operation on general regis
SHLL SHLR	B	Rd shift → Rd Performs a logical shift operation on general register c
ROTL ROTR	B	Rd rotate → Rd Rotates general register contents
ROTXL ROTXR	B	Rd rotate through carry → Rd Rotates general register contents through the C (carry)

Notes: * Size: Operand size
B: Byte

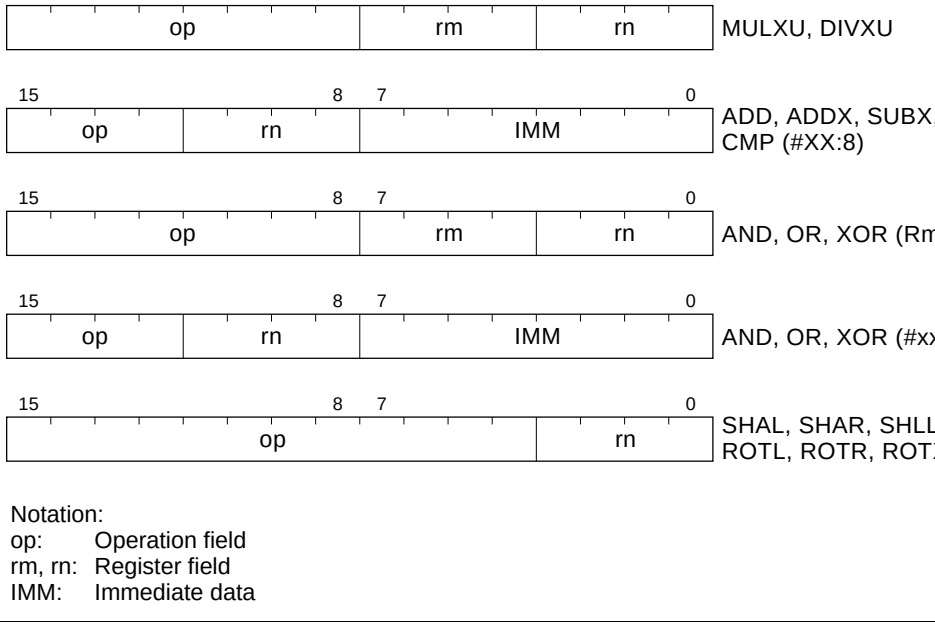


Figure 2.6 Arithmetic, Logic, and Shift Instruction Codes

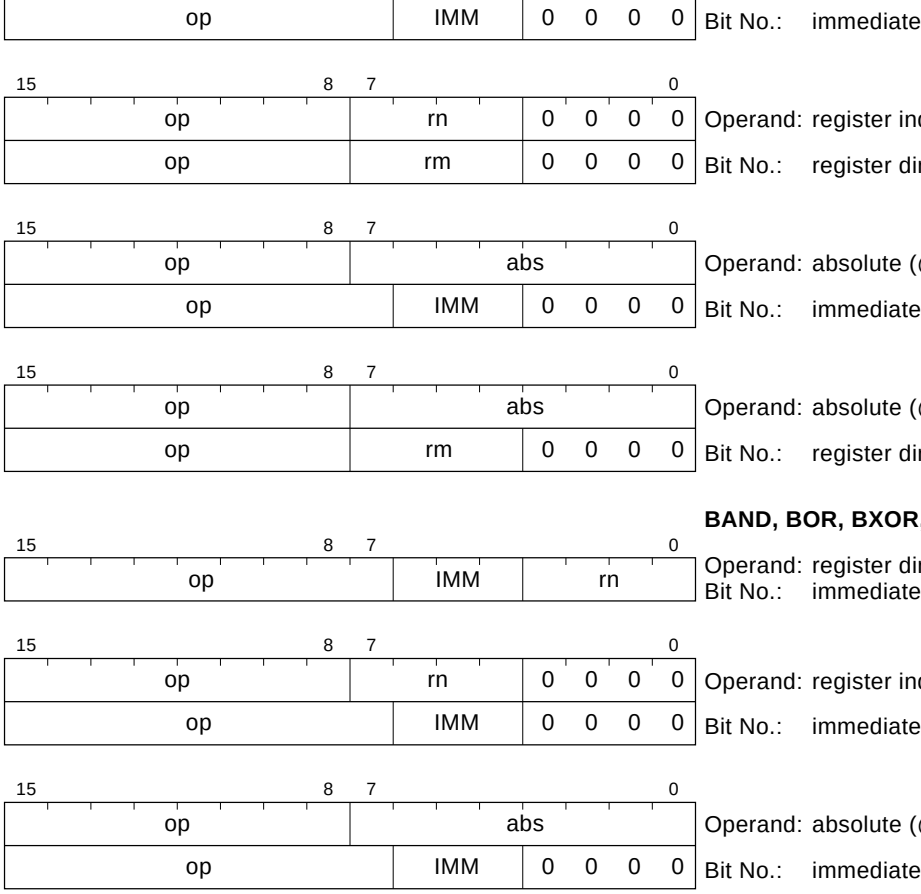
		number is specified by 3-bit immediate data or the lowe of a general register.
BCLR	B	0 → (<bit-No.> of <EAd>) Clears a specified bit in a general register or memory number is specified by 3-bit immediate data or the lowe of a general register.
BNOT	B	~ (<bit-No.> of <EAd>) → (<bit-No.> of <EAd>) Inverts a specified bit in a general register or memory. number is specified by 3-bit immediate data or the lowe of a general register.
BTST	B	~ (<bit-No.> of <EAd>) → Z Tests a specified bit in a general register or memory an clears the Z flag accordingly. The bit number is specifi immediate data or the lower three bits of a general reg
BAND	B	C ∧ (<bit-No.> of <EAd>) → C ANDs the C flag with a specified bit in a general registe memory, and stores the result in the C flag.
BIAND	B	C ∧ [~ (<bit-No.> of <EAd>)] → C ANDs the C flag with the inverse of a specified bit in a register or memory, and stores the result in the C flag. The bit number is specified by 3-bit immediate data.
BOR	B	C ∨ (<bit-No.> of <EAd>) → C ORs the C flag with a specified bit in a general register and stores the result in the C flag.
BIOR	B	C ∨ [~ (<bit-No.> of <EAd>)] → C ORs the C flag with the inverse of a specified bit in a g register or memory, and stores the result in the C flag. The bit number is specified by 3-bit immediate data.

Notes: * Size: Operand size
B: Byte

		The bit number is specified by 3-bit immediate data.
BLD	B	(<bit-No.> of <EAd>) → C
		Copies a specified bit in a general register or memory to
BILD	B	~ (<bit-No.> of <EAd>) → C
		Copies the inverse of a specified bit in a general registe memory to the C flag.
		The bit number is specified by 3-bit immediate data.
BST	B	C → (<bit-No.> of <EAd>)
		Copies the C flag to a specified bit in a general register
BIST	B	~ C → (<bit-No.> of <EAd>)
		Copies the inverse of the C flag to a specified bit in a ge register or memory.
		The bit number is specified by 3-bit immediate data.

Notes: * Size: Operand size
B: Byte

Certain precautions are required in bit manipulation. See 2.9.2, Notes on Bit Manipulat details.



Notation:
op: Operation field
rm, rn: Register field
abs: Absolute address
IMM: Immediate data

Figure 2.7 Bit Manipulation Instruction Codes

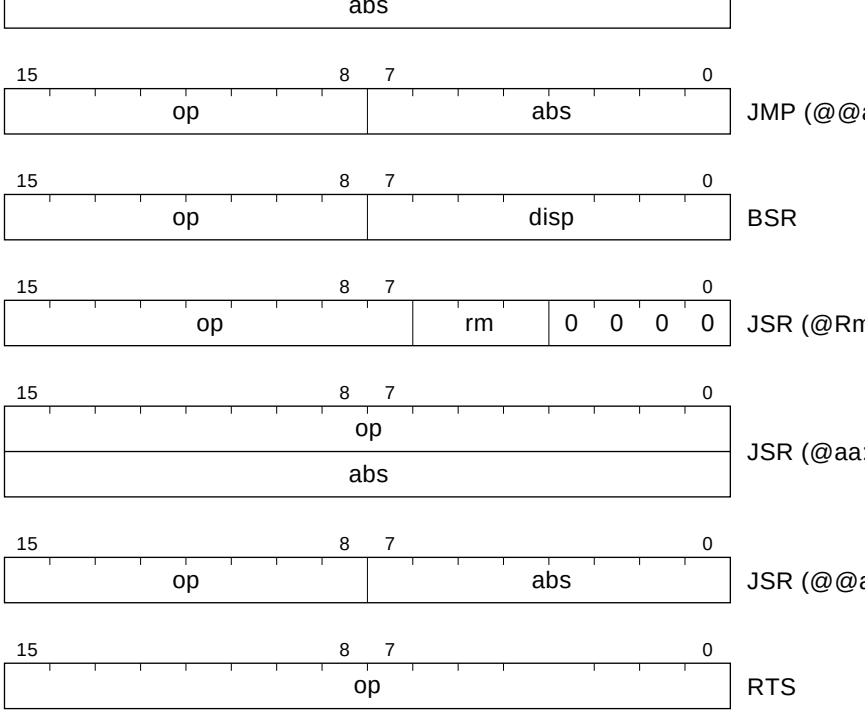
op	abs	Operand: absolute (@a
op	IMM	0 0 0 0 Bit No.: immediate (#

Notation:
op: Operation field
rm, rn: Register field
abs: Absolute address
IMM: Immediate data

Figure 2.7 Bit Manipulation Instruction Codes (cont)

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

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



Notation:
op: Operation field
cc: Condition field
rm: Register field
disp: Displacement
abs: Absolute address

Figure 2.8 Branching Instruction Codes

		section 5, Power-Down Modes, for details.
LDC	B	Rs → CCR, #IMM → CCR Moves immediate data or general register contents to t code register
STC	B	CCR → Rd Copies the condition code register to a specified gener
ANDC	B	CCR ∧ #IMM → CCR Logically ANDs the condition code register with immed
ORC	B	CCR ∨ #IMM → CCR Logically ORs the condition code register with immedia
XORC	B	CCR ⊕ #IMM → CCR Logically exclusive-ORs the condition code register wit data
NOP	—	PC + 2 → PC Only increments the program counter

Notes: * Size: Operand size
B: Byte

Notation:
op: Operation field
rn: Register field
IMM: Immediate data

Figure 2.9 System Control Instruction Codes

2.5.8 Block Data Transfer Instruction

Table 2.11 describes the block data transfer instruction. Figure 2.10 shows its object co

Table 2.11 Block Data Transfer Instruction

Instruction	Size	Function
EEPMOV	—	If R4L 0 then repeat @R5+ → @R6+ R4L – 1 → R4L until R4L = 0 else next; Block transfer instruction. Transfers the number of data specified by R4L from locations starting at the address i R5 to locations starting at the address indicated by R6. transfer, the next instruction is executed.

Certain precautions are required in using the EEPMOV instruction. See 2.9.3, Notes on EEPMOV Instruction, for details.

Access to on-chip memory takes place in two states. The data bus width is 16 bits, allo
access in byte or word size. Figure 2.11 shows the on-chip memory access cycle.

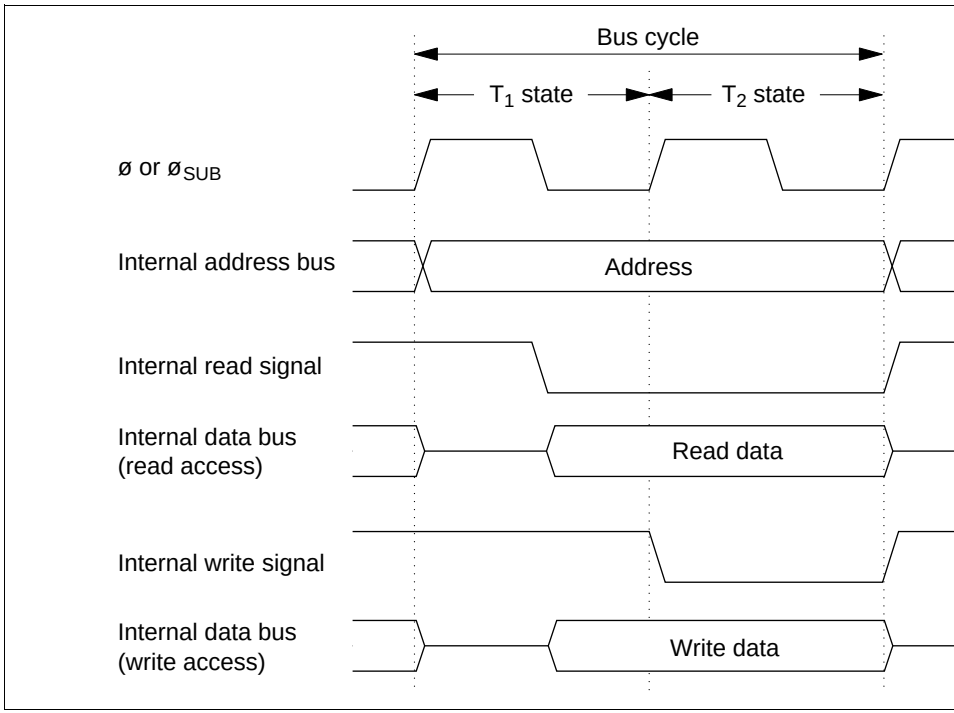


Figure 2.11 On-Chip Memory Access Cycle

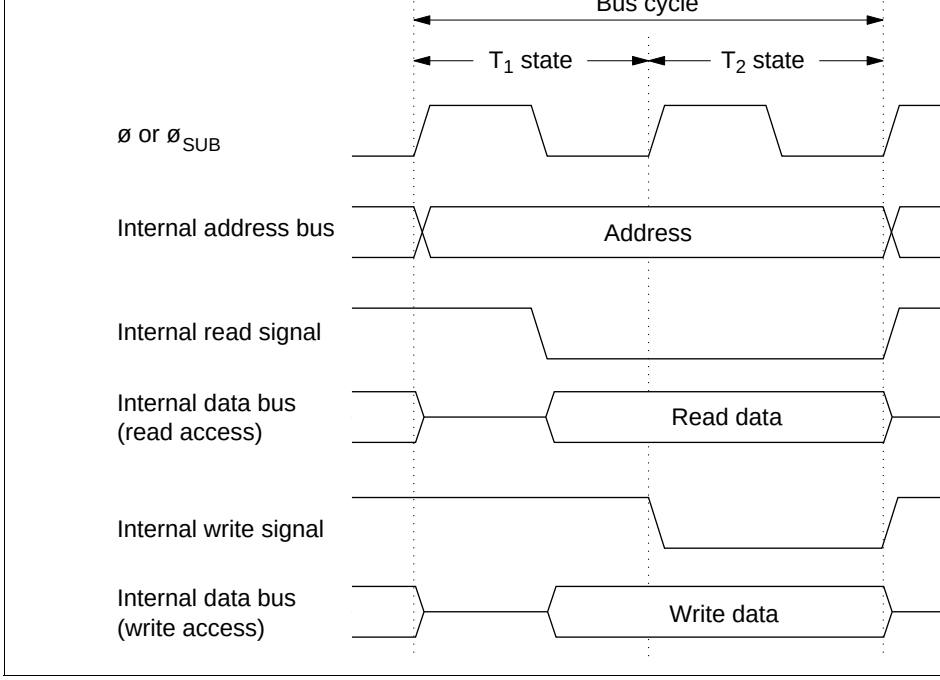


Figure 2.12 On-Chip Peripheral Module Access Cycle (2-State Acces

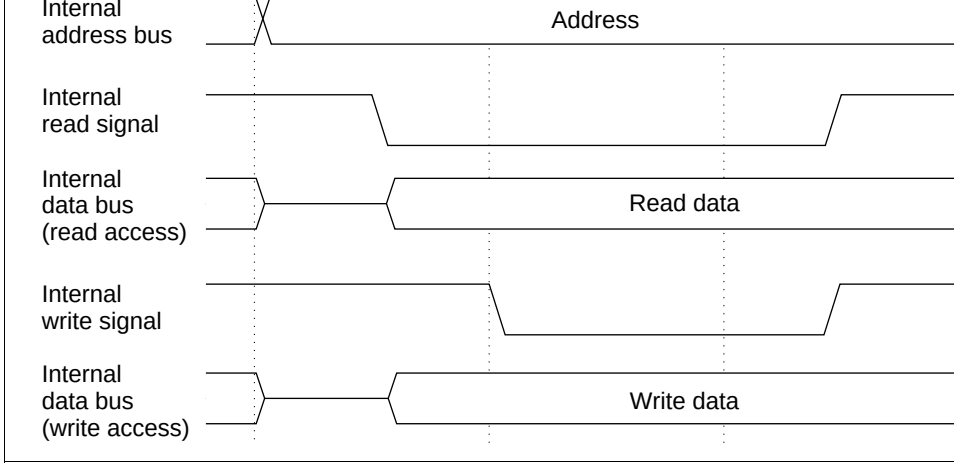


Figure 2.13 On-Chip Peripheral Module Access Cycle (3-State Access)

figure 2.14. Figure 2.15 shows the state transitions.

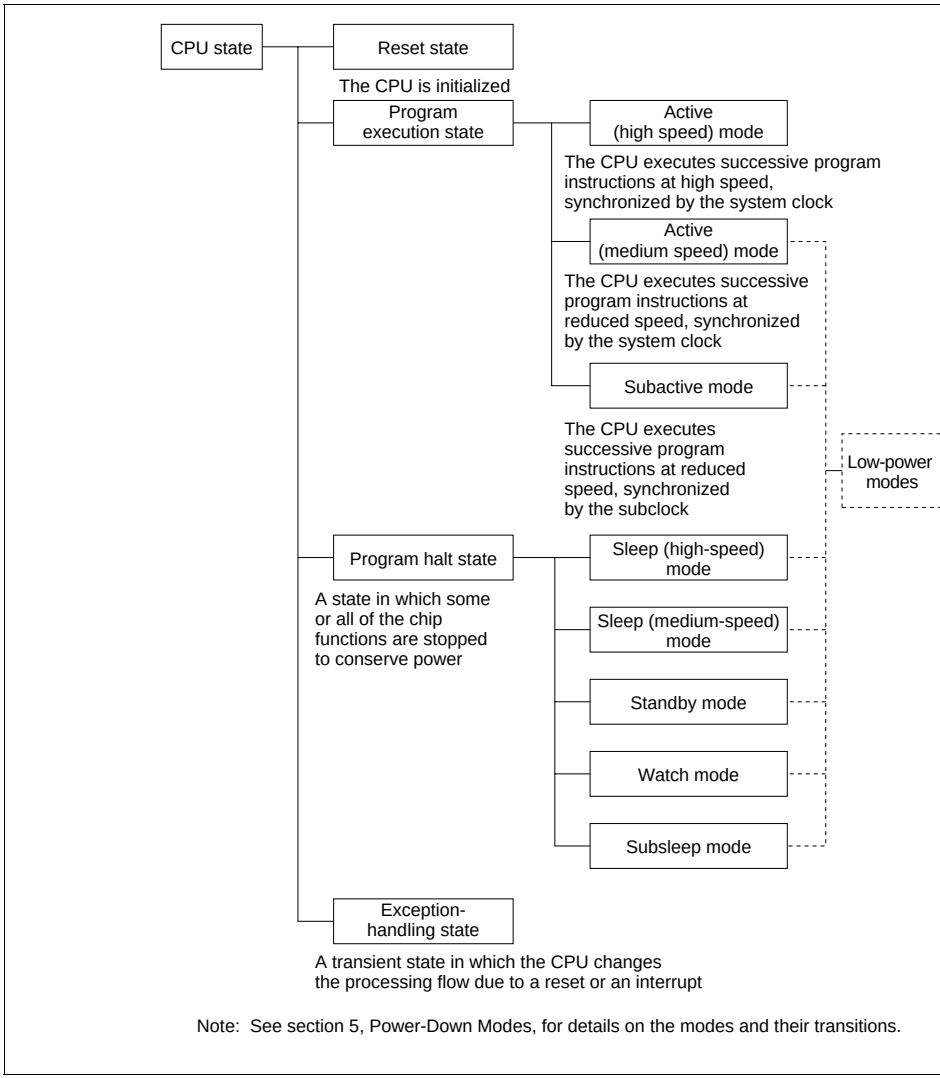


Figure 2.14 CPU Operation States

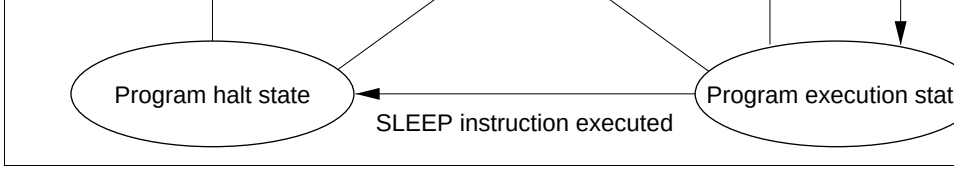


Figure 2.15 State Transitions

2.7.2 Program Execution State

In the program execution state the CPU executes program instructions in sequence.

There are three modes in this state, two active modes (high speed and medium speed) a
subactive mode. Operation is synchronized with the system clock in active mode (high
medium speed), and with the subclock in subactive mode. See section 5, Power-Down M
details on these modes.

2.7.3 Program Halt State

In the program halt state there are five modes: two sleep modes (high speed and mediu
standby mode, watch mode, and subsleep mode. See section 5, Power-Down Modes fo
these modes.

2.7.4 Exception-Handling State

The exception-handling state is a transient state occurring when exception handling is s
reset or interrupt and the CPU changes its normal processing flow. In exception handli
by an interrupt, SP (R7) is referenced and the PC and CCR values are saved on the stac

For details on interrupt handling, see section 3.3, Interrupts.

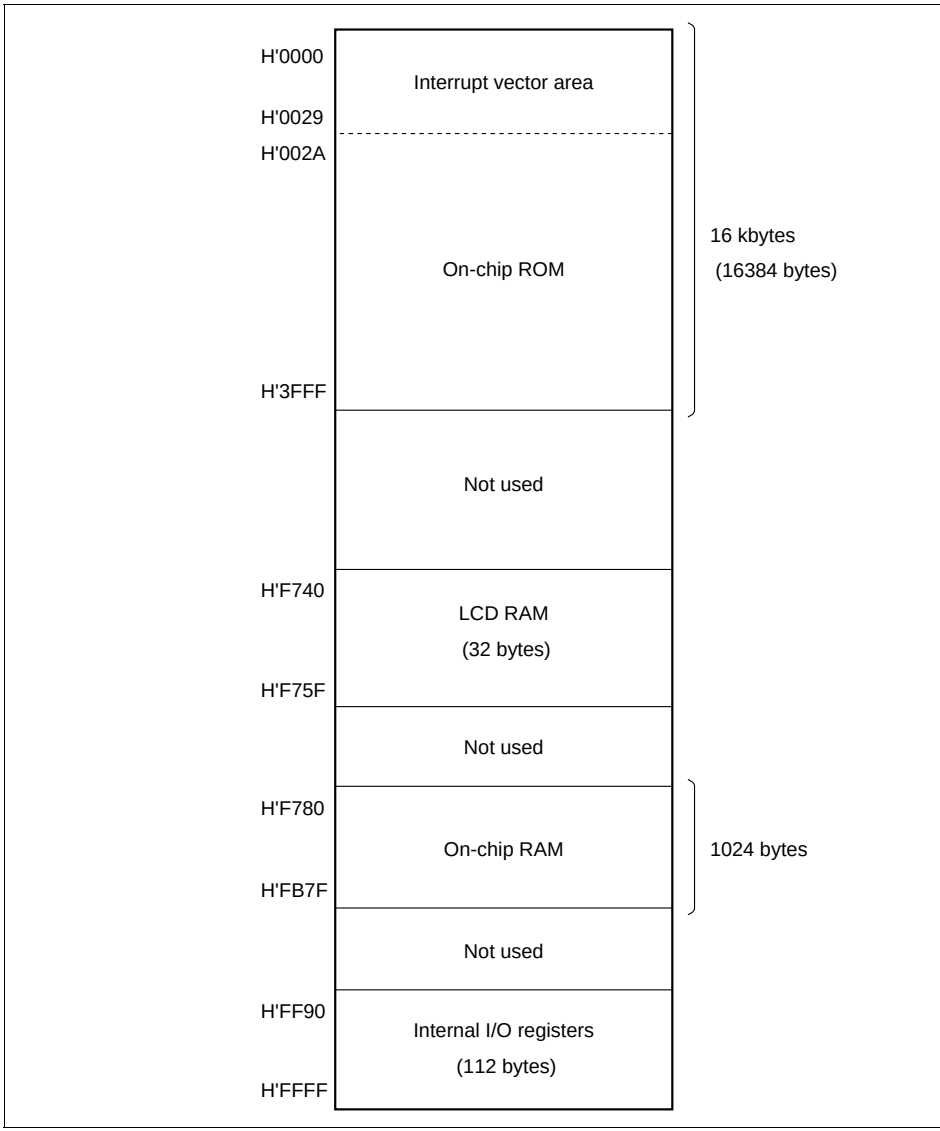


Figure 2.16 (1) H8/3862 and H8/3822 Memory Map

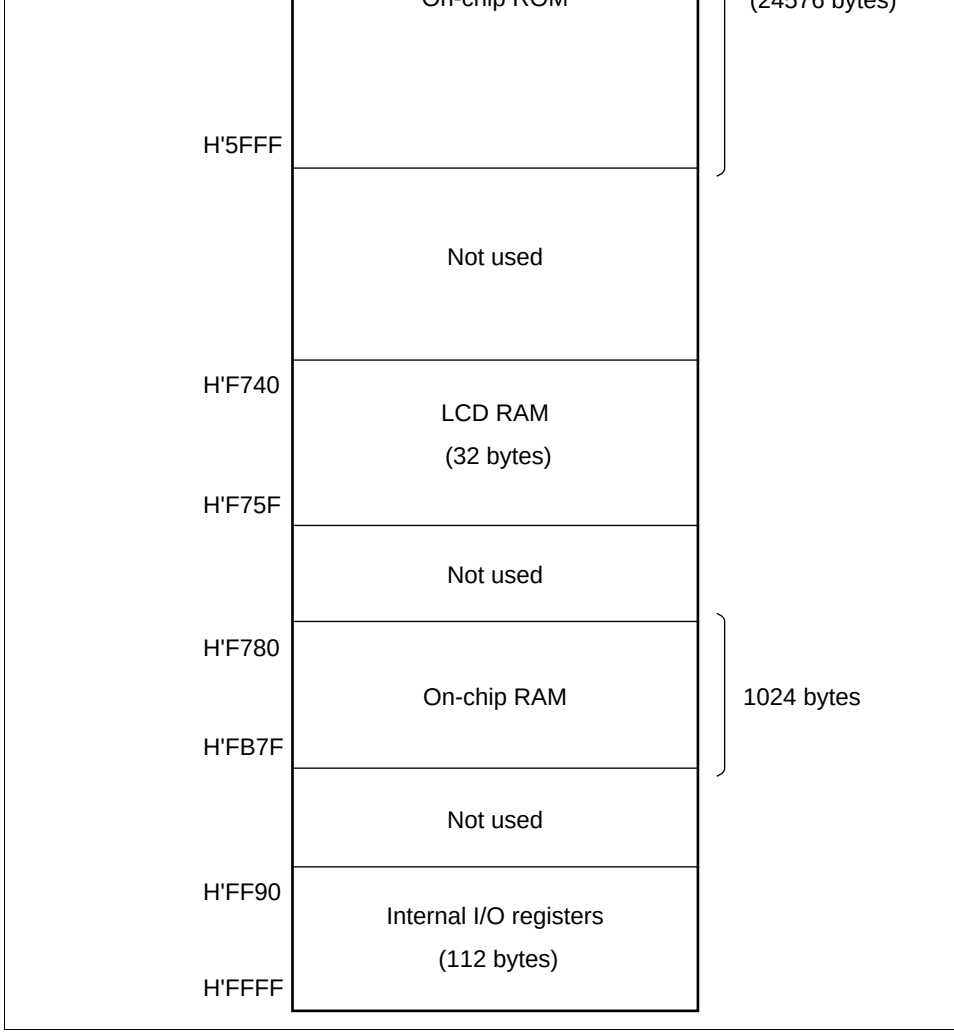


Figure 2.16 (2) H8/3863 and H8/3823 Memory Map

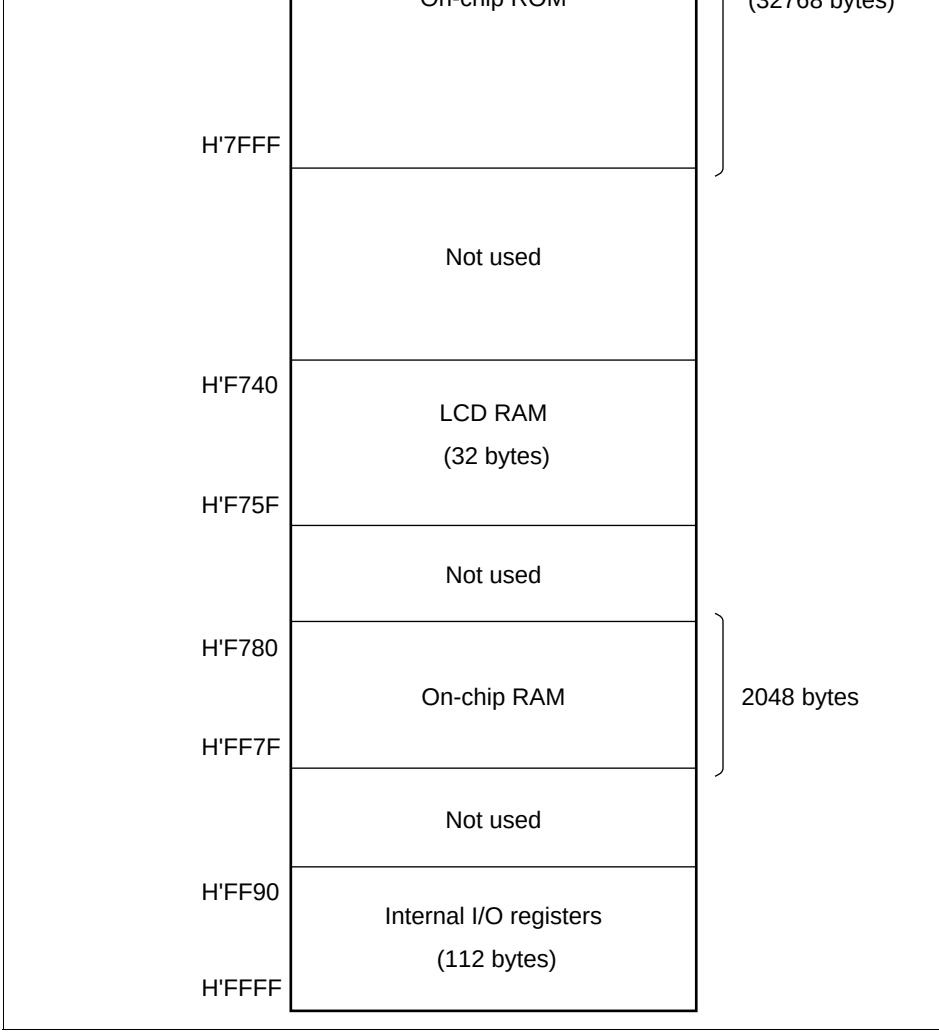


Figure 2.16 (3) H8/3864 and H8/3824 Memory Map

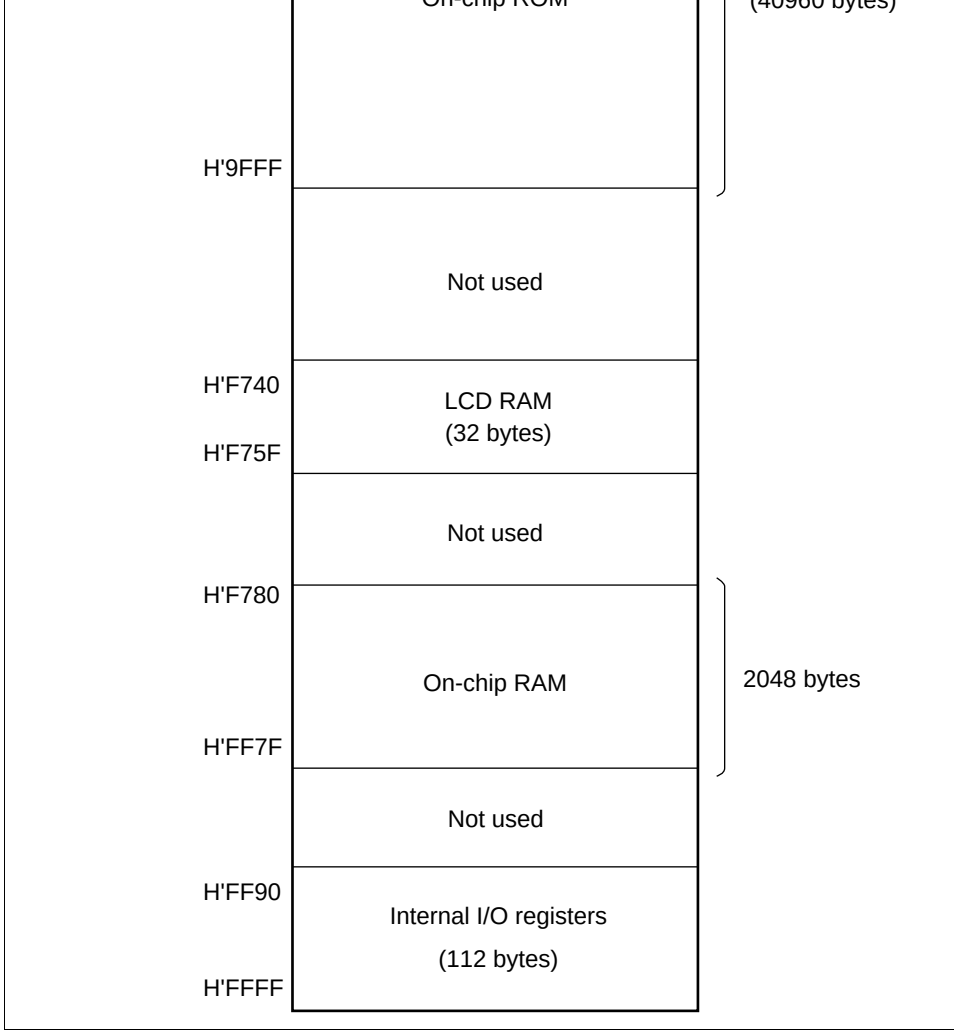


Figure 2.16 (4) H8/3865 and H8/3825 Memory Map

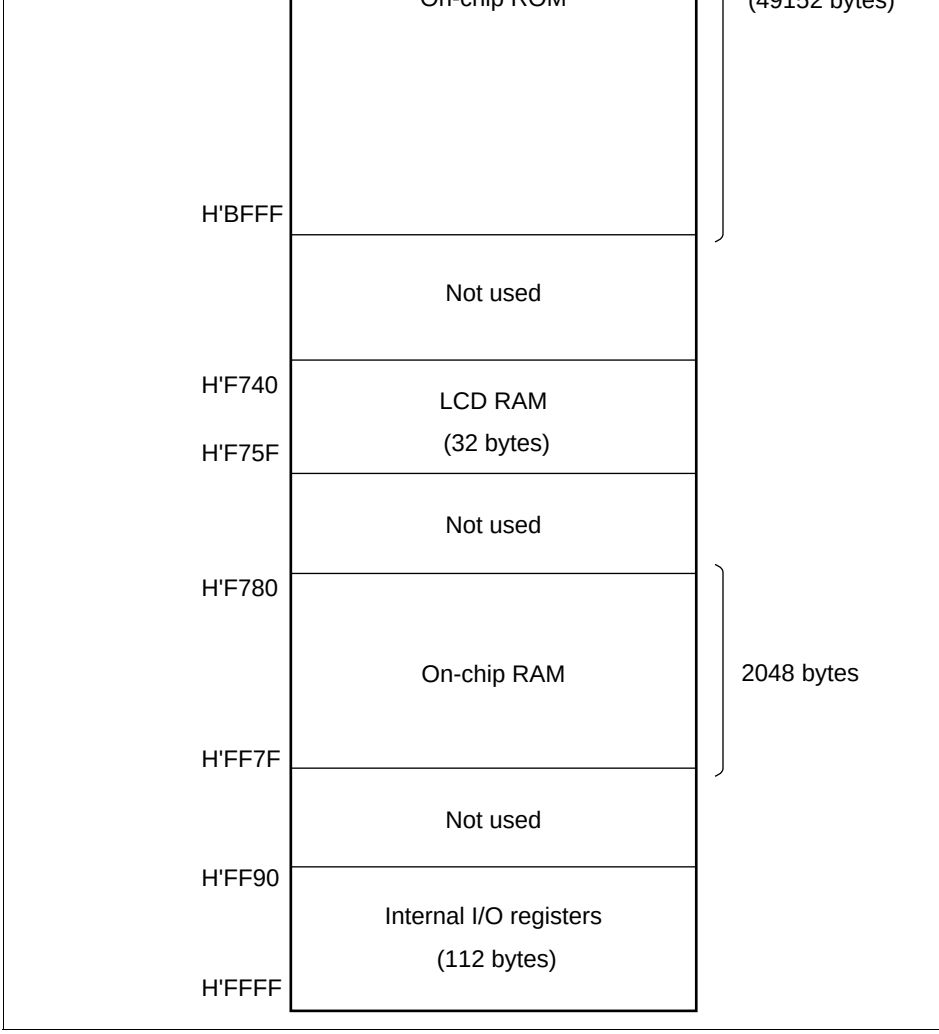


Figure 2.16 (5) H8/3866 and H8/3826 Memory Map

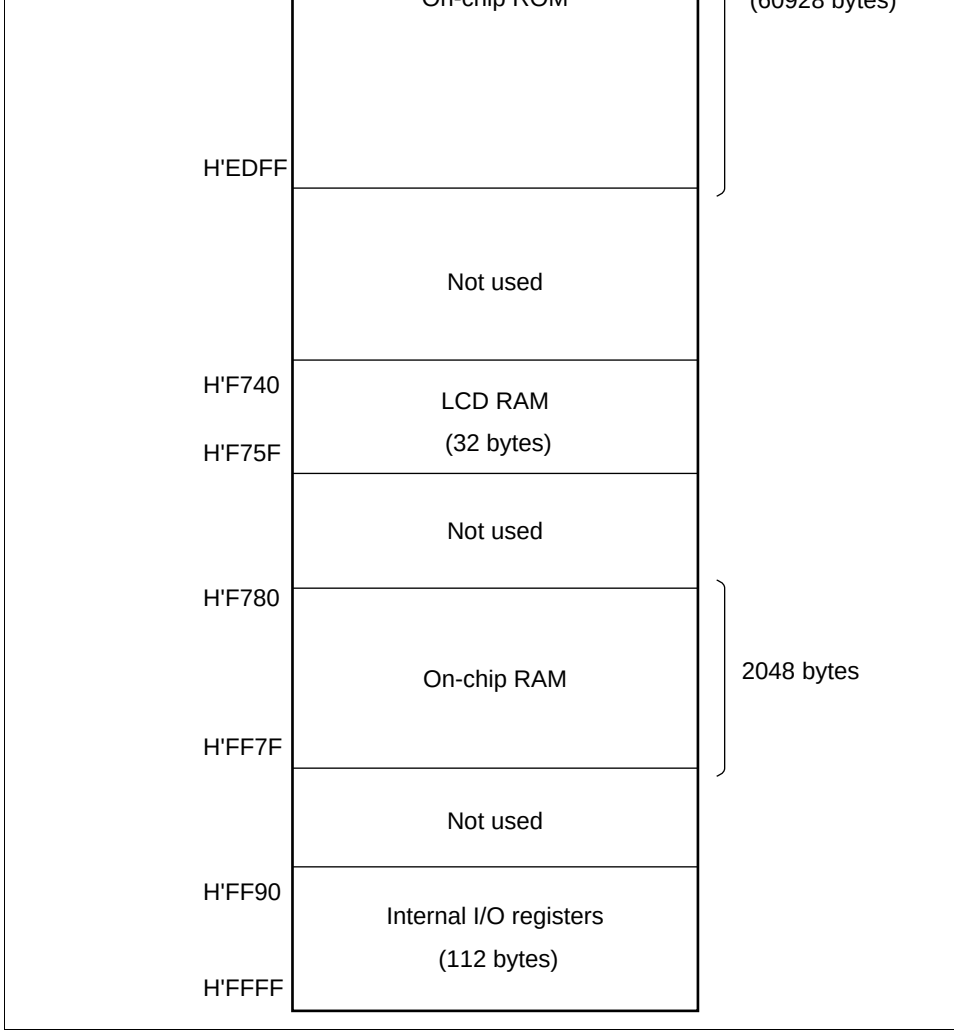


Figure 2.16 (6) H8/3867 and H8/3827 Memory Map

Data transfer from CPU to empty area:

The transferred data will be lost. This action may also cause the CPU to miso

Data transfer from empty area to CPU:

Unpredictable data is transferred.

2. Access to Internal I/O Registers:

Internal data transfer to or from on-chip modules other than the ROM and RAM ar use of an 8-bit data width. If word access is attempted to these areas, the following occur.

Word access from CPU to I/O register area:

Upper byte: Will be written to I/O register.

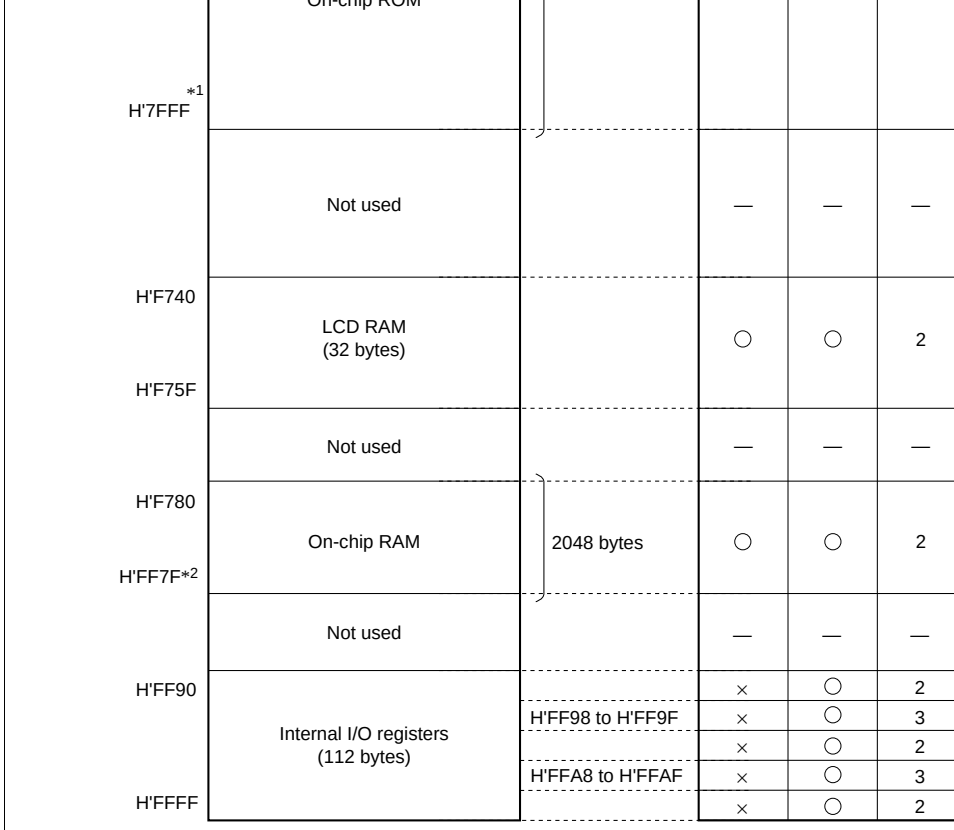
Lower byte: Transferred data will be lost.

Word access from I/O register to CPU:

Upper byte: Will be written to upper part of CPU register.

Lower byte: Unpredictable data will be written to lower part of CPU register.

Byte size instructions should therefore be used when transferring data to or from I/O r other than the on-chip ROM and RAM areas. Figure 2.17 shows the data size and nur states in which on-chip peripheral modules can be accessed.



Notes: The example of the H8/3864 and H8/3824 is shown here.

1. This address is H'3FFF in the H8/3862 and H8/3822 (16-kbyte on-chip ROM), H'5FFF in the H8/3863 and H8/3823 (24-kbyte on-chip ROM), H'9FFF in the H8/3865 and H8/3825 (40-kbyte on-chip ROM), H'BFFF in the H8/3866 and H8/3826 (48-kbyte on-chip ROM), and H'EDFF in the H8/3867 and H8/3827 (60-kbyte on-chip ROM).
2. This address is H'FB7F in the H8/3862, H8/3822, H8/3863, and H8/3823 (1024 bytes of on-chip RAM).

Figure 2.17 Data Size and Number of States for Access to and from On-Chip Peripheral Modules

2	Modify	Modify a designated bit in the read data
3	Write	Write the altered byte data to the designated address

1. Bit manipulation in two registers assigned to the same address

Example 1: timer load register and timer counter

Figure 2.18 shows an example in which two timer registers share the same address. W
manipulation instruction accesses the timer load register and timer counter of a reload
since these two registers share the same address, the following operations take place.

Order of Operation		Operation
1	Read	Timer counter data is read (one byte)
2	Modify	The CPU modifies (sets or resets) the bit designated in the in
3	Write	The altered byte data is written to the timer load register

The timer counter is counting, so the value read is not necessarily the same as the valu
timer load register. As a result, bits other than the intended bit in the timer load regist
modified to the timer counter value.

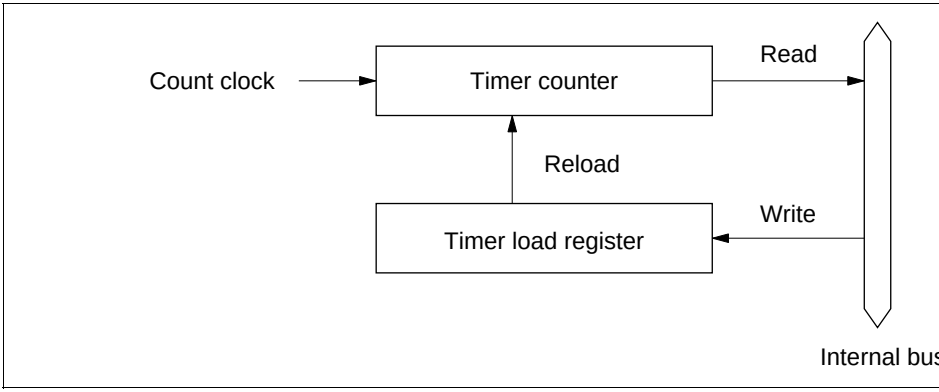


Figure 2.18 Timer Configuration Example

Input/output	Input	Input	Output	Output	Output	Output	Output
Pin state	Low level	High level	Low level	Low level	Low level	Low level	Low leve
PCR3	0	0	1	1	1	1	1
PDR3	1	0	0	0	0	0	0

[B: BSET instruction executed]

```
BSET    #0  ,  @PDR3
```

The BSET instruction is executed designating port

[C: After executing BSET]

	$P3_7$	**$P3_6$**	**$P3_5$**	**$P3_4$**	**$P3_3$**	**$P3_2$**	**$P3_1$**
Input/output	Input	Input	Output	Output	Output	Output	Output
Pin state	Low level	High level	Low level	Low level	Low level	Low level	Low leve
PCR3	0	0	1	1	1	1	1
PDR3	0	1	0	0	0	0	0

[D: Explanation of how BSET operates]

When the BSET instruction is executed, first the CPU reads port 3.

Since $P3_7$ and $P3_6$ are input pins, the CPU reads the pin states (low-level and high-level $P3_5$ to $P3_0$ are output pins, so the CPU reads the value in PDR3. In this example PDR3 value of H'80, but the value read by the CPU is H'40.

Next, the CPU sets bit 0 of the read data to 1, changing the PDR3 data to H'41. Finally writes this value (H'41) to PDR3, completing execution of BSET.

As a result of this operation, bit 0 in PDR3 becomes 1, and $P3_0$ outputs a high-level sig However, bits 7 and 6 of PDR3 end up with different values.

To avoid this problem, store a copy of the PDR3 data in a work area in memory. Perfo manipulation on the data in the work area, then write this data to PDR3.

Pin state	Low level	High level	Low level	Low level	Low level	Low level	Low lev
PCR3	0	0	1	1	1	1	1
PDR3	1	0	0	0	0	0	0
RAM0	1	0	0	0	0	0	0

[B: BSET instruction executed]

```
BSET   #0  ,  @RAM0
```

The BSET instruction is executed designating the work area (RAM0).

[C: After executing BSET]

```
MOV. B  @RAM0, R0L
MOV. B  R0L,  @PDR3
```

The work area (RAM0) value is written to PDR3.

	$P3_7$	**$P3_6$**	**$P3_5$**	**$P3_4$**	**$P3_3$**	**$P3_2$**	**$P3_1$**
Input/output	Input	Input	Output	Output	Output	Output	Output
Pin state	Low level	High level	Low level	Low level	Low level	Low level	Low lev
PCR3	0	0	1	1	1	1	1
PDR3	1	0	0	0	0	0	0
RAM0	1	0	0	0	0	0	0

[A: Before executing BCLR]

	$P3_7$	$P3_6$	$P3_5$	$P3_4$	$P3_3$	$P3_2$	$P3_1$
Input/output	Input	Input	Output	Output	Output	Output	Output
Pin state	Low level	High level	Low level	Low level	Low level	Low level	Low leve
PCR3	0	0	1	1	1	1	1
PDR3	1	0	0	0	0	0	0

[B: BCLR instruction executed]

```
BCLR   #0  ,  @PCR3
```

The BCLR instruction is executed designating PCR

[C: After executing BCLR]

	$P3_7$	$P3_6$	$P3_5$	$P3_4$	$P3_3$	$P3_2$	$P3_1$
Input/output	Output	Output	Output	Output	Output	Output	Output
Pin state	Low level	High level	Low level	Low level	Low level	Low level	Low leve
PCR3	1	1	1	1	1	1	1
PDR3	1	0	0	0	0	0	0

[D: Explanation of how BCLR operates]

When the BCLR instruction is executed, first the CPU reads PCR3. Since PCR3 is a w
register, the CPU reads a value of H'FF, even though the PCR3 value is actually H'3F.

Next, the CPU clears bit 0 in the read data to 0, changing the data to H'FE. Finally, this
(H'FE) is written to PCR3 and BCLR instruction execution ends.

As a result of this operation, bit 0 in PCR3 becomes 0, making $P3_0$ an input port. How
and 6 in PCR3 change to 1, so that $P3_7$ and $P3_6$ change from input pins to output pins.

To avoid this problem, store a copy of the PCR3 data in a work area in memory. Perfo
manipulation on the data in the work area, then write this data to PCR3.

Pin state	Low level	High level	Low level	Low level	Low level	Low level	Low lev
PCR3	0	0	1	1	1	1	1
PDR3	1	0	0	0	0	0	0
RAM0	0	0	1	1	1	1	1

[B: BCLR instruction executed]

```
BCLR   #0  ,  @RAM0
```

The BCLR instruction is executed designating the work area (RAM0).

[C: After executing BCLR]

```
MOV. B  @RAM0, R0L
MOV. B  R0L,  @PCR3
```

The work area (RAM0) value is written to PCR3.

	$P3_7$	**$P3_6$**	**$P3_5$**	**$P3_4$**	**$P3_3$**	**$P3_2$**	**$P3_1$**
Input/output	Input	Input	Output	Output	Output	Output	Output
Pin state	Low level	High level	Low level	Low level	Low level	Low level	Low lev
PCR3	0	0	1	1	1	1	1
PDR3	1	0	0	0	0	0	0
RAM0	0	0	1	1	1	1	1

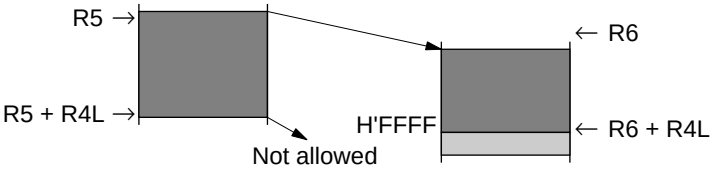
Port data register 4*	PDR4	H'FFD7
Port data register 5*	PDR5	H'FFD8
Port data register 6*	PDR6	H'FFD9
Port data register 7*	PDR7	H'FFDA
Port data register 8*	PDR8	H'FFDB
Port data register A*	PDRA	H'FFDD

Note: * Port data registers have the same addresses as input pins.

Table 2.13 Registers with Write-Only Bits

Register Name	Abbreviation	Address
Port control register 1	PCR1	H'FFE4
Port control register 3	PCR3	H'FFE6
Port control register 4	PCR4	H'FFE7
Port control register 5	PCR5	H'FFE8
Port control register 6	PCR6	H'FFE9
Port control register 7	PCR7	H'FFEA
Port control register 8	PCR8	H'FFEB
Port control register A	PCRA	H'FFED
Timer control register F	TCRF	H'FFB6
PWM control register	PWCR	H'FFD0
PWM data register U	PWDRU	H'FFD1
PWM data register L	PWDRL	H'FFD2

- When setting R4L and R6, make sure that the final destination address (R6 + R4L) exceed H'FFFF. The value in R6 must not change from H'FFFF to H'0000 during e the instruction.



Priority	Exception Source	Time of Start of Exception Handling
High	Reset	Exception handling starts as soon as the reset state
↑	Interrupt	When an interrupt is requested, exception handling s execution of the present instruction or the exception progress is completed
Low		

3.2 Reset

3.2.1 Overview

A reset is the highest-priority exception. The internal state of the CPU and the registe chip peripheral modules are initialized.

3.2.2 Reset Sequence

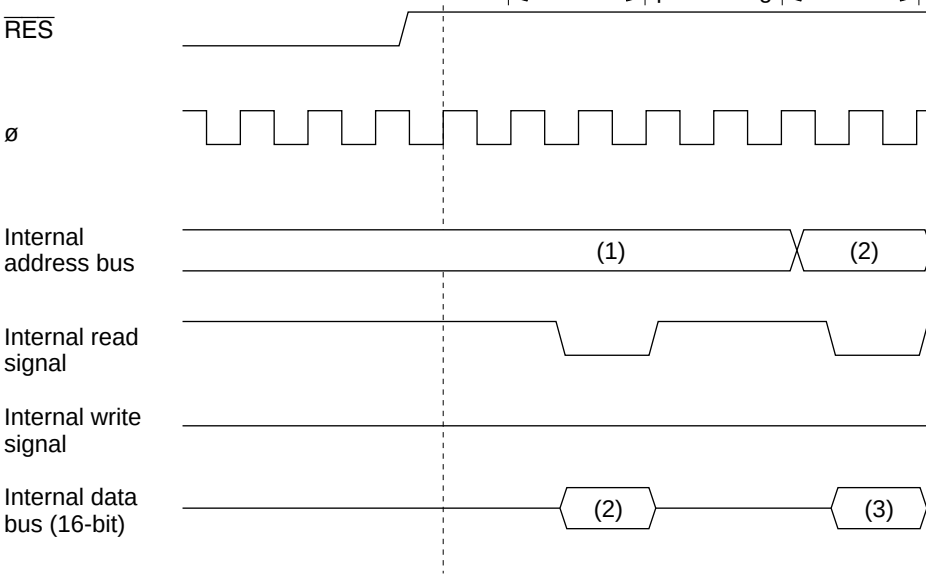
As soon as the $\overline{\text{RES}}$ pin goes low, all processing is stopped and the chip enters the res

To make sure the chip is reset properly, observe the following precautions.

- At power on: Hold the $\overline{\text{RES}}$ pin low until the clock pulse generator output stabilize
- Resetting during operation: Hold the $\overline{\text{RES}}$ pin low for at least 10 system clock cycl

Reset exception handling takes place as follows.

- The CPU internal state and the registers of on-chip peripheral modules are initializ I bit of the condition code register (CCR) set to 1.
- The PC is loaded from the reset exception handling vector address (H'0000 to H'00 which the program starts executing from the address indicated in PC.



(1) Reset exception handling vector address (H'0000)
(2) Program start address
(3) First instruction of program

Figure 3.1 Reset Sequence

3.3 Interrupts

3.3.1 Overview

The interrupt sources include 13 external interrupts (IRQ_4 to IRQ_0, WKP_7 to WKP_0) a
internal interrupts from on-chip peripheral modules. Table 3.2 shows the interrupt sou
priorities, and their vector addresses. When more than one interrupt is requested, the i
the highest priority is processed.

The interrupts have the following features:

- Internal and external interrupts can be masked by the I bit in CCR. When the I bit
  interrupt request flags can be set but the interrupts are not accepted.
- IRQ_4 to IRQ_0 and WKP_7 to WKP_0 can be set to either rising edge sensing or fallin
  sensing.

$\overline{IRQ_4}$	IRQ_4	8	H'0010 to H'0011
$\overline{WKP_0}$ $\overline{WKP_1}$ $\overline{WKP_2}$ $\overline{WKP_3}$ $\overline{WKP_4}$ $\overline{WKP_5}$ $\overline{WKP_6}$ $\overline{WKP_7}$	WKP_0 WKP_1 WKP_2 WKP_3 WKP_4 WKP_5 WKP_6 WKP_7	9	H'0012 to H'0013
Timer A	Timer A overflow	11	H'0016 to H'0017
Asynchronous counter	Asynchronous counter overflow	12	H'0018 to H'0019
Timer C	Timer C overflow or underflow	13	H'001A to H'001B
Timer FL	Timer FL compare match Timer FL overflow	14	H'001C to H'001D
Timer FH	Timer FH compare match Timer FH overflow	15	H'001E to H'001F
Timer G	Timer G input capture Timer G overflow	16	H'0020 to H'0021
SCI3-1	SCI3-1 transmit end SCI3-1 transmit data empty SCI3-1 receive data full SCI3-1 overrrun error SCI3-1 framing error SCI3-1 parity error	17	H'0022 to H'0023
SCI3-2	SCI3-2 transmit end SCI3-2 transmit data empty SCI3-2 receive data full SCI3-2 overrun error SCI3-2 framing error SCI3-2 parity error	18	H'0024 to H'0025
A/D	A/D conversion end	19	H'0026 to H'0027
(SLEEP instruction executed)	Direct transfer	20	H'0028 to H'0029

Note: Vector addresses H'0002 to H'0007 and H'0014 to H'0015 are reserved and cann used.

Interrupt enable register 2	IENR2	R/W	H'00
Interrupt request register 1	IRR1	R/W*	H'20
Interrupt request register 2	IRR2	R/W*	H'00
Wakeup interrupt request register	IWPR	R/W*	H'00
Wakeup edge select register	WEGR	R/W	H'00

Note: * Write is enabled only for writing of 0 to clear a flag.

1. IRQ edge select register (IEGR)

Bit	7	6	5	4	3	2	1
	—	—	—	IEG4	IEG3	IEG2	IEG1
Initial value	1	1	1	0	0	0	0
Read/Write	—	—	—	R/W	R/W	R/W	R/W

IEGR is an 8-bit read/write register used to designate whether pins $\overline{IRQ_4}$ to $\overline{IRQ_0}$ are s
edge sensing or falling edge sensing.

Bits 7 to 5: Reserved bits

Bits 7 to 5 are reserved: they are always read as 1 and cannot be modified.

Bit 4: IRQ_4 edge select (IEG4)

Bit 4 selects the input sensing of the $\overline{IRQ_4}$ pin and $\overline{ADTRG}$ pin.

Bit 4 IEG4	Description
0	Falling edge of $\overline{IRQ_4}$ and $\overline{ADTRG}$ pin input is detected
1	Rising edge of $\overline{IRQ_4}$ and $\overline{ADTRG}$ pin input is detected

Bit 2: IRQ_2 edge select (IEG2)

Bit 2 selects the input sensing of pin $\overline{IRQ}_2$.

Bit 2 IEG2	Description	
0	Falling edge of $\overline{IRQ}_2$ pin input is detected	(i
1	Rising edge of $\overline{IRQ}_2$ pin input is detected	

Bit 1: IRQ_1 edge select (IEG1)

Bit 3 selects the input sensing of the $\overline{IRQ}_1$ pin and TMIC pin.

Bit 1 IEG1	Description	
0	Falling edge of $\overline{IRQ}_1$ and TMIC pin input is detected	(i
1	Rising edge of $\overline{IRQ}_1$ and TMIC pin input is detected	

Bit 0: IRQ_0 edge select (IEG0)

Bit 0 selects the input sensing of pin $\overline{IRQ}_0$.

Bit 0 IEG0	Description	
0	Falling edge of $\overline{IRQ}_0$ pin input is detected	(i
1	Rising edge of $\overline{IRQ}_0$ pin input is detected	

Bit 7: Timer A interrupt enable (IENTA)

Bit 7 enables or disables timer A overflow interrupt requests.

Bit 7 IENTA	Description
0	Disables timer A interrupt requests
1	Enables timer A interrupt requests

Bit 6: Reserved bit

Bit 6 is a readable/writable reserved bit. It is initialized to 0 by a reset.

Bit 5: Wakeup interrupt enable (IENWP)

Bit 5 enables or disables WKP_7 to WKP_0 interrupt requests.

Bit 5 IENWP	Description
0	Disables $\overline{WKP}_7$ to $\overline{WKP}_0$ interrupt requests
1	Enables $\overline{WKP}_7$ to $\overline{WKP}_0$ interrupt requests

Bits 4 to 0: IRQ_4 to IRQ_0 interrupt enable (IEN4 to IEN0)

Bits 4 to 0 enable or disable IRQ_4 to IRQ_0 interrupt requests.

Bit n IENn	Description
0	Disables interrupt requests from pin $\overline{IRQn}$
1	Enables interrupt requests from pin $\overline{IRQn}$

Bit 7: Direct transfer interrupt enable (IENDT)

Bit 7 enables or disables direct transfer interrupt requests.

Bit 7 IENDT	Description	
0	Disables direct transfer interrupt requests	(ir
1	Enables direct transfer interrupt requests	

Bit 6: A/D converter interrupt enable (IENAD)

Bit 6 enables or disables A/D converter interrupt requests.

Bit 6 IENAD	Description	
0	Disables A/D converter interrupt requests	(ir
1	Enables A/D converter interrupt requests	

Bit 5: Reserved bit

Bit 5 is a readable/writable reserved bit. It is initialized to 0 by a reset.

Bit 4: Timer G interrupt enable (IENTG)

Bit 4 enables or disables timer G input capture or overflow interrupt requests.

Bit 4 IENTG	Description	
0	Disables timer G interrupt requests	(ir
1	Enables timer G interrupt requests	

Bit 2: Timer FL interrupt enable (IENTFL)

Bit 2 enables or disables timer FL compare match and overflow interrupt requests.

Bit 2 IENTFL	Description
0	Disables timer FL interrupt requests
1	Enables timer FL interrupt requests

Bit 1: Timer C interrupt enable (IENTC)

Bit 1 enables or disables timer C overflow and underflow interrupt requests.

Bit 1 IENTC	Description
0	Disables timer C interrupt requests
1	Enables timer C interrupt requests

Bit 0: Asynchronous event counter interrupt enable (IENEC)

Bit 0 enables or disables asynchronous event counter interrupt requests.

Bit 0 IENEC	Description
0	Disables asynchronous event counter interrupt requests
1	Enables asynchronous event counter interrupt requests

For details of SCI3-1 and SCI3-2 interrupt control, see 6. Serial control register 3 (SCI
section 10.4.2.

IRR1 is an 8-bit read/write register, in which a corresponding flag is set to 1 when a tim
IRQ_4 to IRQ_0 interrupt is requested. The flags are not cleared automatically when an in
accepted. It is necessary to write 0 to clear each flag.

Bit 7: Timer A interrupt request flag (IRRTA)

Bit 7 IRRTA	Description
0	Clearing conditions: When IRRTA = 1, it is cleared by writing 0 (ir
1	Setting conditions: When the timer A counter value overflows from H'FF to H'00

Bit 6: Reserved bit

Bit 6 is a readable/writable reserved bit. It is initialized to 0 by a reset.

Bit 5: Reserved bit

Bit 5 is reserved; it is always read as 1 and cannot be modified.

Bits 4 to 0: IRQ_4 to IRQ_0 interrupt request flags (IRRI4 to IRRI0)

Bit n IRRIn	Description
0	Clearing conditions: When IRRIn = 1, it is cleared by writing 0 (ir
1	Setting conditions: When pin $\overline{\text{IRQn}}$ is designated for interrupt input and the designated signal input

IRR2 is an 8-bit read/write register, in which a corresponding flag is set to 1 when a di
transfer, A/D converter, Timer G, Timer FH, Timer FC, or Timer C interrupt is reques
flags are not cleared automatically when an interrupt is accepted. It is necessary to wr
each flag.

Bit 7: Direct transfer interrupt request flag (IRRDT)

Bit 7 IRRDT	Description
0	Clearing conditions: When IRRDT = 1, it is cleared by writing 0
1	Setting conditions: When a direct transfer is made by executing a SLEEP instruction while D SYSCR2

Bit 6: A/D converter interrupt request flag (IRRAD)

Bit 6 IRRAD	Description
0	Clearing conditions: When IRRAD = 1, it is cleared by writing 0
1	Setting conditions: When A/D conversion is completed and ADSF is cleared to 0 in ADSR

Bit 5: Reserved bit

Bit 5 is a readable/writable reserved bit. It is initialized to 0 by a reset.

Bit 3: Timer FH interrupt request flag (IRRTFH)

Bit 3 IRRTFH	Description
0	Clearing conditions: When IRRTFH = 1, it is cleared by writing 0 (ir
1	Setting conditions: When TCFH and OCRFH match in 8-bit timer mode, or when TCF (TCFL, and OCRF (OCRFL, OCRFH) match in 16-bit timer mode

Bit 2: Timer FL interrupt request flag (IRRTFL)

Bit 2 IRRTFL	Description
0	Clearing conditions: When IRRTFL= 1, it is cleared by writing 0 (ir
1	Setting conditions: When TCFL and OCRFL match in 8-bit timer mode

Bit 1: Timer C interrupt request flag (IRRTC)

Bit 1 IRRTC	Description
0	Clearing conditions: When IRRTC= 1, it is cleared by writing 0 (ir
1	Setting conditions: When the timer C counter value overflows (from H'FF to H'00) or underflo (from H'00 to H'FF)

6. Wakeup Interrupt Request Register (IWPR)

Bit	7	6	5	4	3	2	1
	IWPF7	IWPF6	IWPF5	IWPF4	IWPF3	IWPF2	IWPF1
Initial value	0	0	0	0	0	0	0
Read/Write	R/W*	R/W*	R/W*	R/W*	R/W*	R/W*	R/W*

Note: * Only a write of 0 for flag clearing is possible

IWPR is an 8-bit read/write register containing wakeup interrupt request flags. When $\overline{WKP}_7$ to $\overline{WKP}_0$ is designated for wakeup input and a rising or falling edge is input at corresponding flag in IWPR is set to 1. A flag is not cleared automatically when the corresponding interrupt is accepted. Flags must be cleared by writing 0.

Bits 7 to 0: Wakeup interrupt request flags (IWPF7 to IWPF0)

Bit n IWPFn	**Description**
0	Clearing conditions: When IWPFn= 1, it is cleared by writing 0
1	Setting conditions: When pin $\overline{WKP}_n$ is designated for wakeup input and a rising or falling ed that pin

WEGR is initialized to H'00 by a reset.

Bit n: $\overline{\text{WKP}}$n edge select (WKEGSn)

Bit n selects $\overline{\text{WKP}}$n pin input sensing.

Bit n WKEGSn	Description	
0	$\overline{\text{WKP}}$n pin falling edge detected	(i
1	$\overline{\text{WKP}}$n pin rising edge detected	

3.3.3 External Interrupts

There are 13 external interrupts: IRQ_4 to IRQ_0 and WKP_7 to WKP_0.

1. Interrupts WKP_7 to WKP_0

Interrupts WKP_7 to WKP_0 are requested by either rising or falling edge input to pins $\overline{W}$ $\overline{WKP}_0$. When these pins are designated as pins $\overline{WKP}_7$ to $\overline{WKP}_0$ in port mode register 5 rising or falling edge is input, the corresponding bit in IWPR is set to 1, requesting an i Recognition of wakeup interrupt requests can be disabled by clearing the IENWP bit to IENR1. These interrupts can all be masked by setting the I bit to 1 in CCR.

When WKP_7 to WKP_0 interrupt exception handling is initiated, the I bit is set to 1 in C number 9 is assigned to interrupts WKP_7 to WKP_0. All eight interrupt sources have the vector number, so the interrupt-handling routine must discriminate the interrupt source.

to 0 in IENR1. These interrupts can all be masked by setting the I bit to 1 in CCR.

When IRQ_4 to IRQ_0 interrupt exception handling is initiated, the I bit is set to 1 in CC numbers 8 to 4 are assigned to interrupts IRQ_4 to IRQ_0. The order of priority is from I to IRQ_4 (low). Table 3.2 gives details.

3.3.4 Internal Interrupts

There are 23 internal interrupts that can be requested by the on-chip peripheral module peripheral module requests an interrupt, the corresponding bit in IRR1 or IRR2 is set t Recognition of individual interrupt requests can be disabled by clearing the correspon IENR1 or IENR2. All these interrupts can be masked by setting the I bit to 1 in CCR. internal interrupt handling is initiated, the I bit is set to 1 in CCR. Vector numbers fro are assigned to these interrupts. Table 3.2 shows the order of priority of interrupts fro peripheral modules.

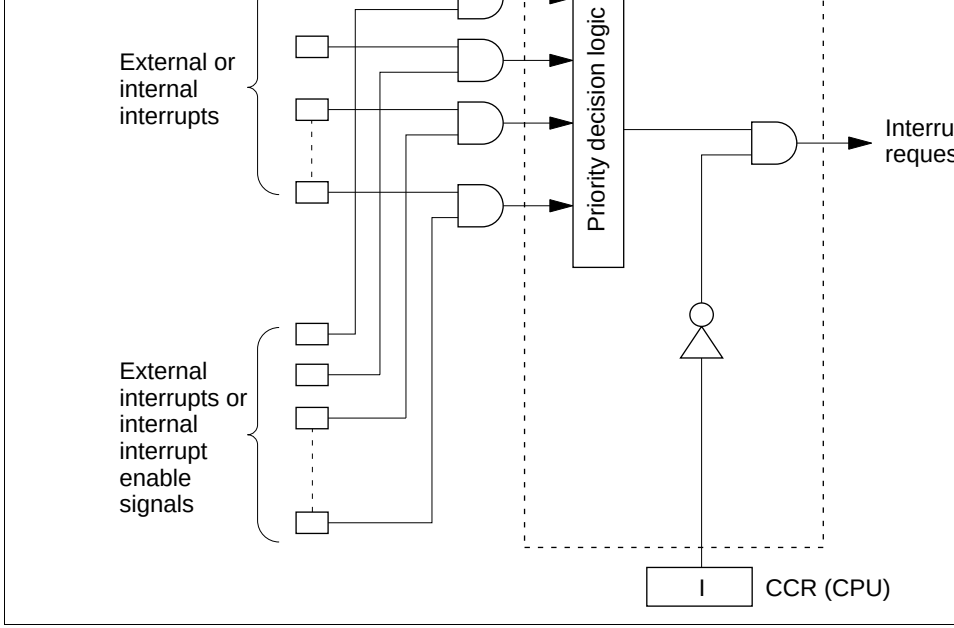


Figure 3.2 Block Diagram of Interrupt Controller

Interrupt operation is described as follows.

- When an interrupt condition is met while the interrupt enable register bit is set to 1, interrupt request signal is sent to the interrupt controller.
- When the interrupt controller receives an interrupt request, it sets the interrupt reque
- From among the interrupts with interrupt request flags set to 1, the interrupt controll the interrupt request with the highest priority and holds the others pending. (Refer t for a list of interrupt priorities.)
- The interrupt controller checks the I bit of CCR. If the I bit is 0, the selected interru is accepted; if the I bit is 1, the interrupt request is held pending.

Notes:

1. When disabling interrupts by clearing bits in an interrupt enable register, or when c
   in an interrupt request register, always do so while interrupts are masked (I = 1).
2. If the above clear operations are performed while I = 0, and as a result a conflict ar
   the clear instruction and an interrupt request, exception processing for the interrupt
   executed after the clear instruction has been executed.

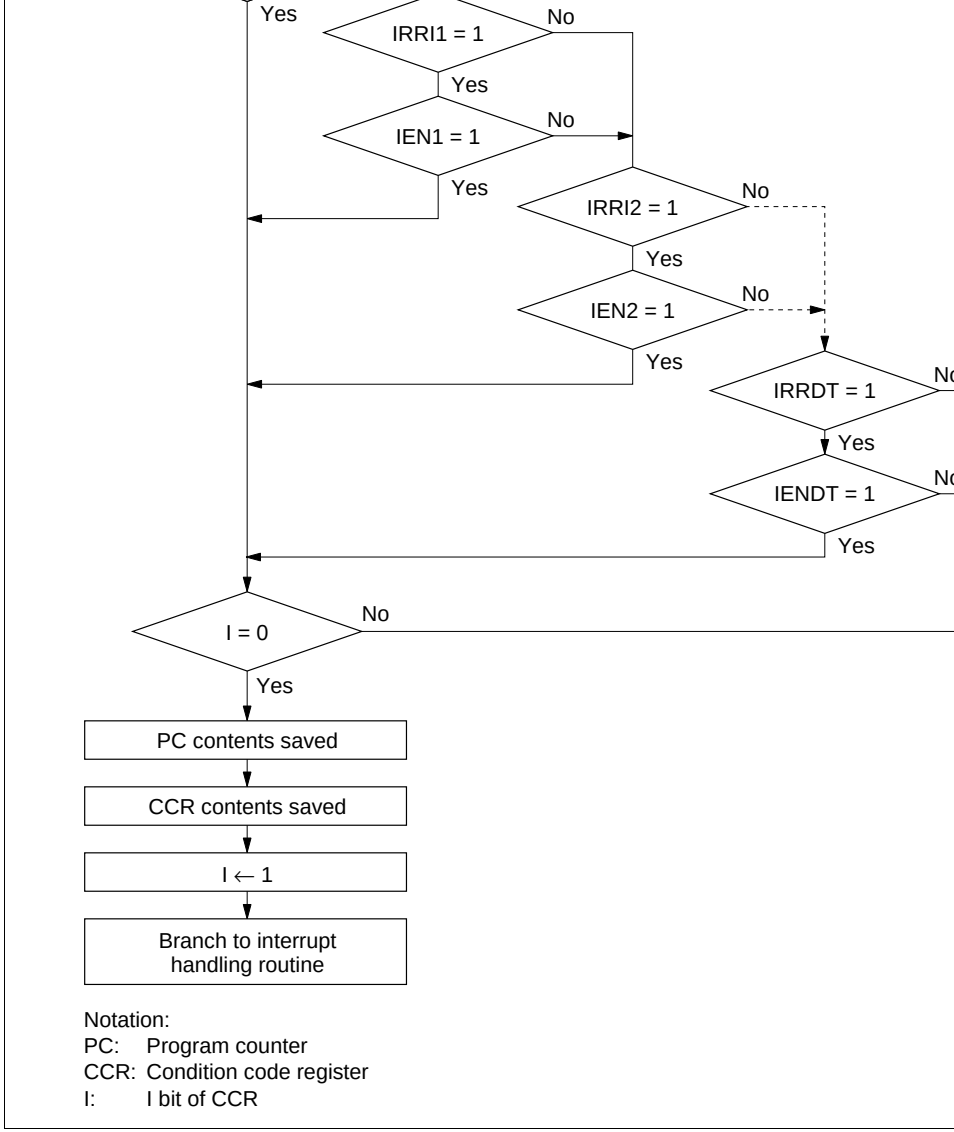
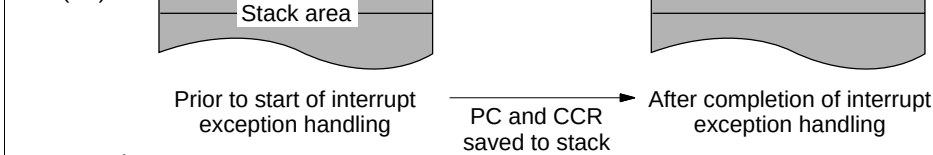


Figure 3.3 Flow up to Interrupt Acceptance



Notation:
PC_H: Upper 8 bits of program counter (PC)
PC_L: Lower 8 bits of program counter (PC)
CCR: Condition code register
SP: Stack pointer

Notes: 1. PC shows the address of the first instruction to be executed upon return from the interrupt handling routine.
2. Register contents must always be saved and restored by word access, starting from an even-numbered address.
* Ignored on return.

Figure 3.4 Stack State after Completion of Interrupt Exception Handli

Figure 3.5 shows a typical interrupt sequence.

Figure 3.5 Interrupt Sequence

Saving of PC and CCR to stack	4
Vector fetch	2
Instruction fetch	4
Internal processing	4

Note: * Not including EEPMOV instruction.

Setting an odd address in SP may cause a program to crash. An example is shown in fi

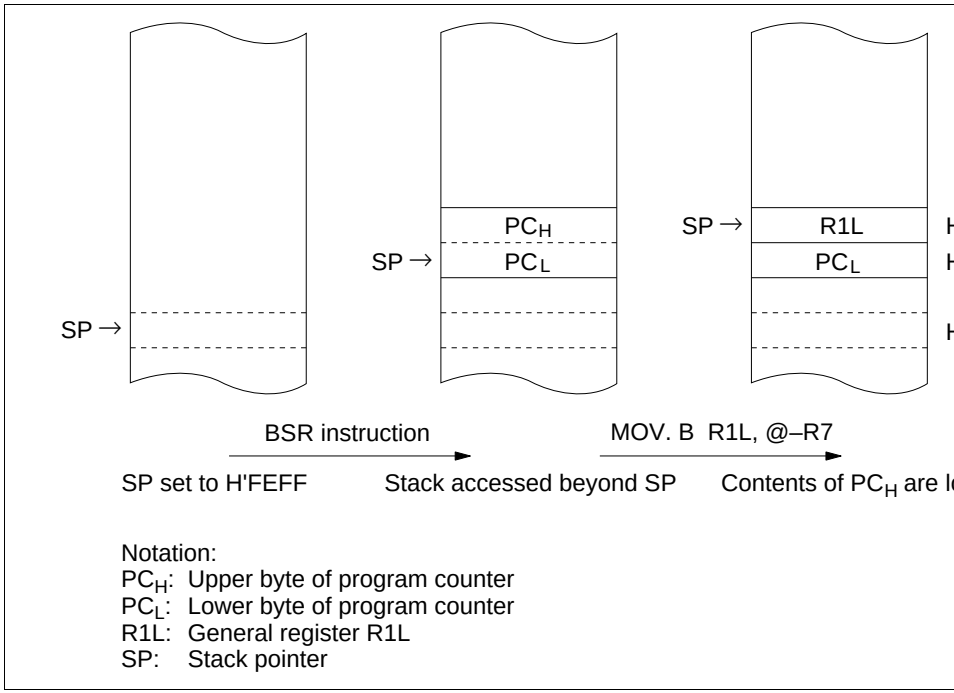


Figure 3.6 Operation when Odd Address is Set in SP

When CCR contents are saved to the stack during interrupt exception handling or restor
RTE is executed, this also takes place in word size. Both the upper and lower bytes of
are saved to the stack; on return, the even address contents are restored to CCR while th
address contents are ignored.

which interrupt request flags are set to 1 in this way.

Table 3.5 Conditions under which Interrupt Request Flag is Set to 1

Interrupt Request Flags Set to 1		Conditions
IRR1	IRRI4	When PMR1 bit IRQ4 is changed from 0 to 1 while pin $\overline{IRQ}_4$ is l bit IEG4 = 0.
		When PMR1 bit IRQ4 is changed from 1 to 0 while pin $\overline{IRQ}_4$ is l bit IEG4 = 1.
	IRRI3	When PMR1 bit IRQ3 is changed from 0 to 1 while pin $\overline{IRQ}_3$ is l bit IEG3 = 0.
		When PMR1 bit IRQ3 is changed from 1 to 0 while pin $\overline{IRQ}_3$ is l bit IEG3 = 1.
	IRRI2	When PMR1 bit IRQ2 is changed from 0 to 1 while pin $\overline{IRQ}_2$ is l bit IEG2 = 0.
		When PMR1 bit IRQ2 is changed from 1 to 0 while pin $\overline{IRQ}_2$ is l bit IEG2 = 1.
	IRRI1	When PMR1 bit IRQ1 is changed from 0 to 1 while pin $\overline{IRQ}_1$ is l bit IEG1 = 0.
		When PMR1 bit IRQ1 is changed from 1 to 0 while pin $\overline{IRQ}_1$ is l bit IEG1 = 1.
	IRRI0	When PMR3 bit IRQ0 is changed from 0 to 1 while pin $\overline{IRQ}_0$ is l bit IEG0 = 0.
		When PMR3 bit IRQ0 is changed from 1 to 0 while pin $\overline{IRQ}_0$ is l bit IEG0 = 1.
IWPR	IWPF7	When PMR5 bit WKP7 is changed from 0 to 1 while pin $\overline{WKP}_7$ i
	IWPF6	When PMR5 bit WKP6 is changed from 0 to 1 while pin $\overline{WKP}_6$ i
	IWPF5	When PMR5 bit WKP5 is changed from 0 to 1 while pin $\overline{WKP}_5$ i
	IWPF4	When PMR5 bit WKP4 is changed from 0 to 1 while pin $\overline{WKP}_4$ i
	IWPF3	When PMR5 bit WKP3 is changed from 0 to 1 while pin $\overline{WKP}_3$ i
	IWPF2	When PMR5 bit WKP2 is changed from 0 to 1 while pin $\overline{WKP}_2$ i
	IWPF1	When PMR5 bit WKP1 is changed from 0 to 1 while pin $\overline{WKP}_1$ i
	IWPF0	When PMR5 bit WKP0 is changed from 0 to 1 while pin $\overline{WKP}_0$ i

An alternative method is to avoid the setting of interrupt request flags when pin functio
switched by keeping the pins at the high level so that the conditions in table 3.5 do not

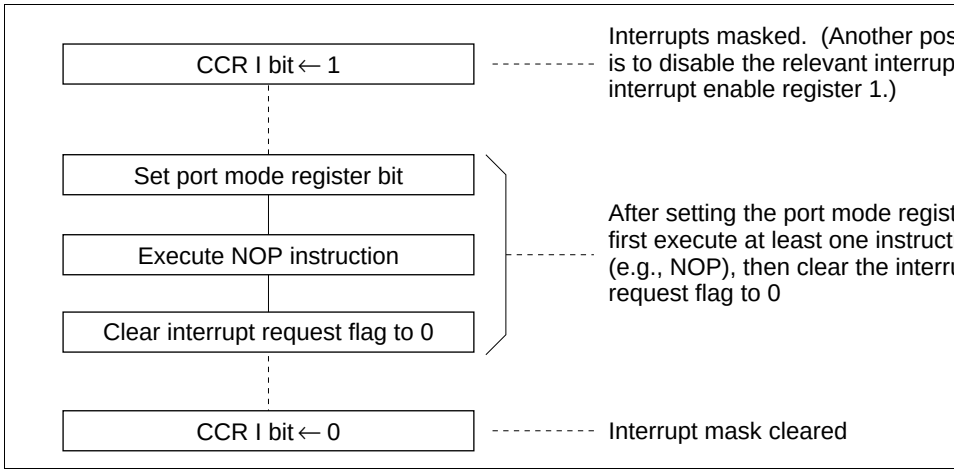


Figure 3.7 Port Mode Register Setting and Interrupt Request Flag Clearing Pr

4.1.1 Block Diagram

Figure 4.1 shows a block diagram of the clock pulse generators.

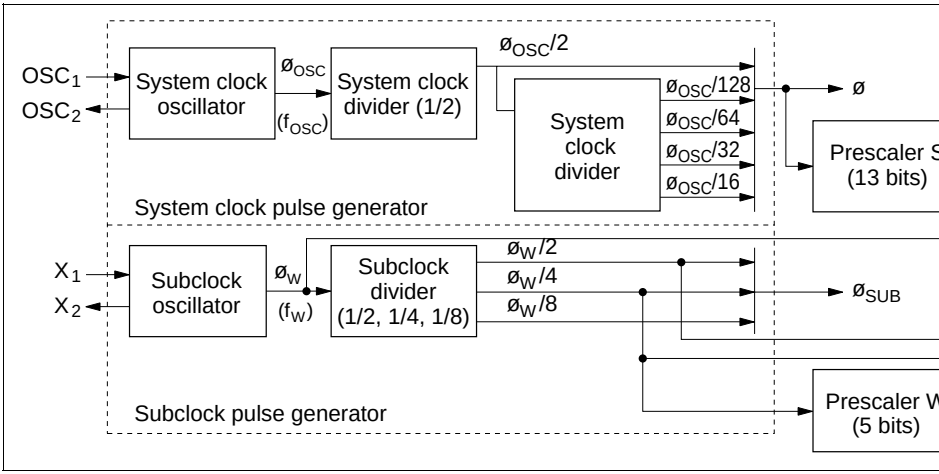
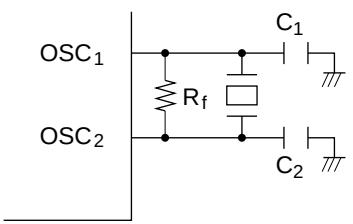


Figure 4.1 Block Diagram of Clock Pulse Generators

4.1.2 System Clock and Subclock

The basic clock signals that drive the CPU and on-chip peripheral modules are ø and ø of the clock signals have names: ø is the system clock, $ø_{SUB}$ is the subclock, $ø_{OSC}$ is t clock, and $ø_W$ is the watch clock.

The clock signals available for use by peripheral modules are ø/2, ø/4, ø/8, ø/16, ø/32, ø/256, ø/512, ø/1024, ø/2048, ø/4096, ø/8192, $ø_W$, $ø_W/2$, $ø_W/4$, $ø_W/8$, $ø_W/16$, $ø_W/32$, $ø_W/128$. The clock requirements differ from one module to another.



R_f = 1 MΩ ±20%

Frequency	Crystal oscillator	C_1, C_2 Recommendation value	Produ
1.0 MHz	NDK	27 pF ±10%	NR-1
4.0 MHz	NDK	12 pF ±20%	NR-1

Figure 4.2 Typical Connection to Crystal Oscillator

Figure 4.3 shows the equivalent circuit of a crystal oscillator. An oscillator having the characteristics given in table 4.1 should be used.

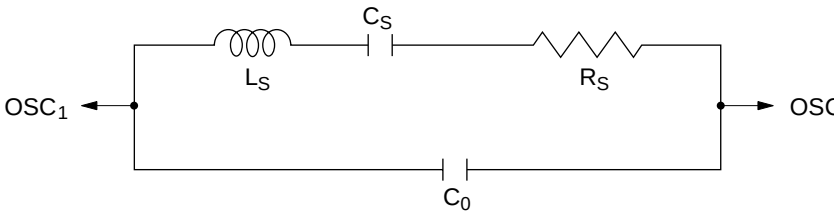


Figure 4.3 Equivalent Circuit of Crystal Oscillator

Table 4.1 Crystal Oscillator Parameters

Frequency (MHz)	1	4.193
RS max ( )	40	100
C_0 (pF)	3.5 pF max	16 pF

4.0 MHz	Murata	30 pF ±10%	CSA

Figure 4.4 Typical Connection to Ceramic Oscillator

3. Notes on board design

When generating clock pulses by connecting a crystal or ceramic oscillator, pay carefu
to the following points.

Avoid running signal lines close to the oscillator circuit, since the oscillator may be ad
affected by induction currents. (See figure 4.5.)

The board should be designed so that the oscillator and load capacitors are located as c
possible to pins OSC_1 and OSC_2.

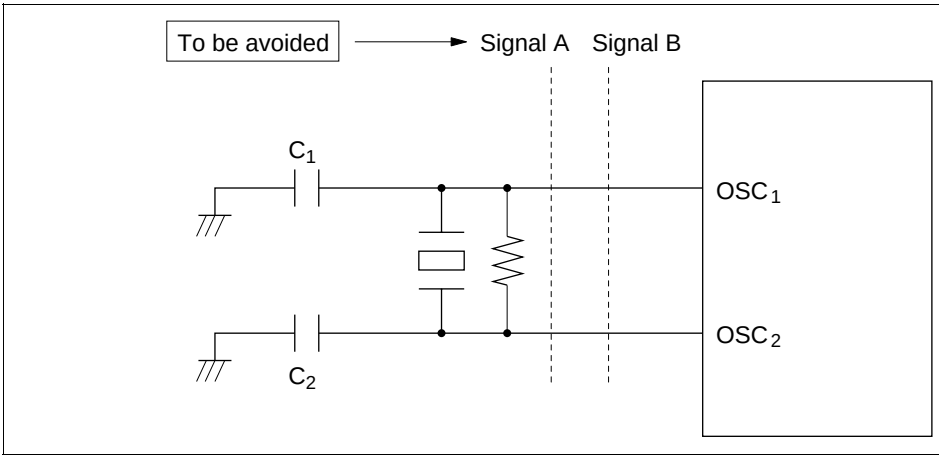


Figure 4.5 Board Design of Oscillator Circuit

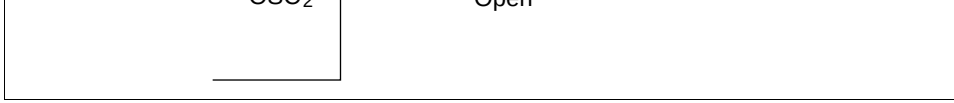
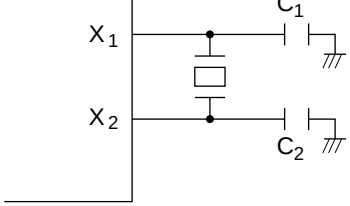


Figure 4.6 External Clock Input (Example)

Frequency	Oscillator Clock ($ø_{OSC}$)
Duty cycle	45% to 55%

Note: The circuit parameters above are recommended by the crystal or ceramic oscilla manufacturer.
The circuit parameters are affected by the crystal or ceramic oscillator and float capacitance when designing the board. When using the oscillator, consult with or ceramic oscillator manufacturer to determine the circuit parameters.



$C_1 = C_2$ = 15 pF (typ.)

Frequency	Crystal oscillator	Products
32.768 kHz	NDK	MX73P
38.4 kHz	Seiko Instrument Inc.	VTC-200

Figure 4.7 Typical Connection to 32.768-kHz/38.4 kHz Crystal Oscillator (S

Figure 4.8 shows the equivalent circuit of the 32.768-kHz/38.4 kHz crystal oscillator.

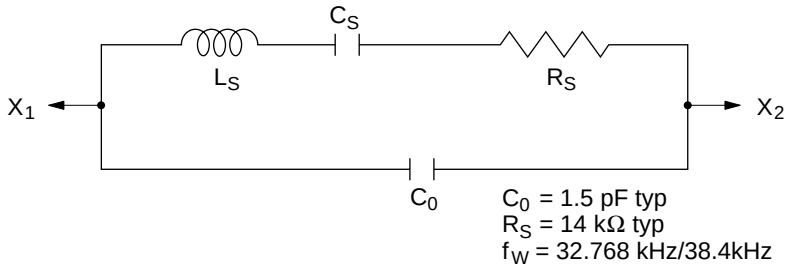


Figure 4.8 Equivalent Circuit of 32.768-kHz/38.4 kHz Crystal Oscillat

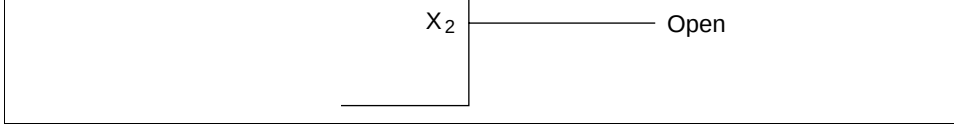


Figure 4.9 Pin Connection when not Using Subclock

3. External clock input

Connect the external clock to the X1 pin and leave the X2 pin open, as shown in figure

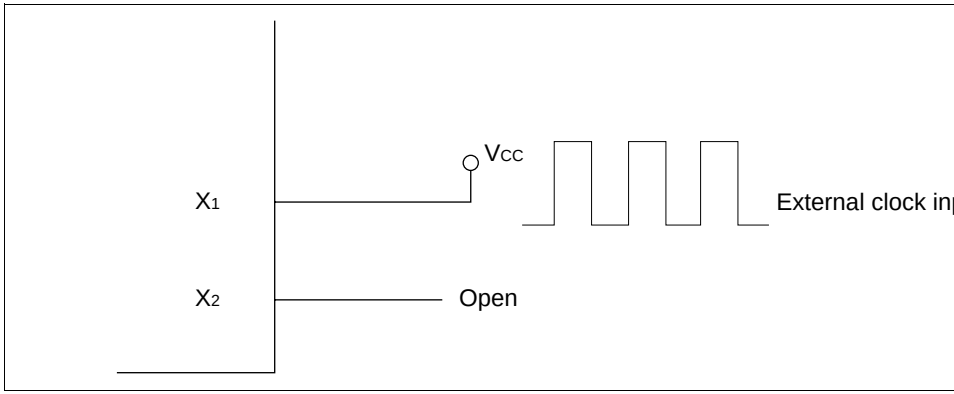


Figure 4.10 Pin Connection when Inputting External Clock

Frequency	Subclock (øw)
Duty	45% to 55%

Prescaler S is a 13-bit counter using the system clock (ø) as its input clock. It is incre
per clock period.

Prescaler S is initialized to H'0000 by a reset, and starts counting on exit from the rese

In standby mode, watch mode, subactive mode, and subsleep mode, the system clock
generator stops. Prescaler S also stops and is initialized to H'0000.

The CPU cannot read or write prescaler S.

The output from prescaler S is shared by timer A, timer C, timer F, timer G, SCI3-1, S
A/D converter, the LCD controller, the watchdog timer, and the 14-bit PWM. The div
can be set separately for each on-chip peripheral function.

In active (medium-speed) mode the clock input to prescaler S is øosc/16, øosc/32, øos
øosc/128.

2. Prescaler W (PSW)

Prescaler W is a 5-bit counter using a 32.768 kHz/38.4 kHz signal divided by 4 ($ø_W/4$
clock.

Prescaler W is initialized to H'00 by a reset, and starts counting on exit from the reset

Even in standby mode, watch mode, subactive mode, or subsleep mode, prescaler W c
functioning so long as clock signals are supplied to pins X1 and X2.

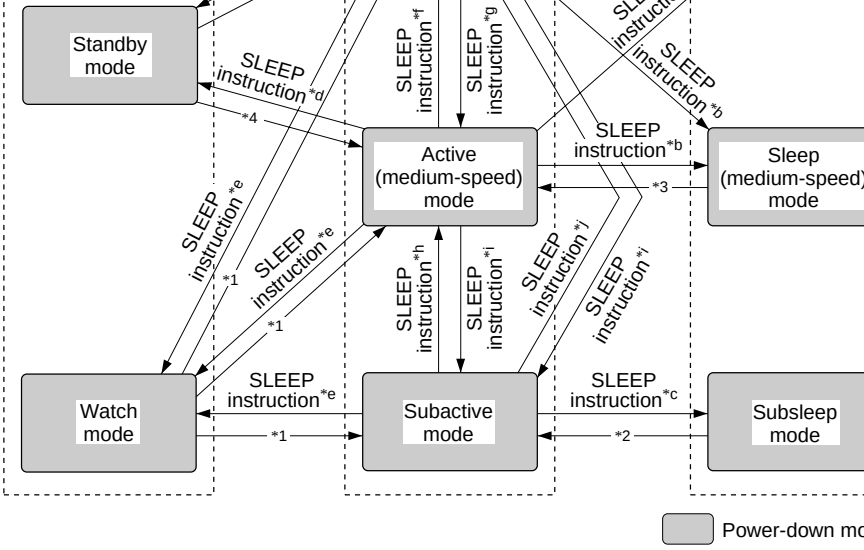
Prescaler W can be reset by setting 1s in bits TMA3 and TMA2 of timer mode registe

Output from prescaler W can be used to drive timer A, in which case timer A function
base for timekeeping.

Table 5.1 Operating Modes

Operating Mode	Description
Active (high-speed) mode	The CPU and all on-chip peripheral functions are o the system clock in high-speed operation
Active (medium-speed) mode	The CPU and all on-chip peripheral functions are o the system clock in low-speed operation
Subactive mode	The CPU is operable on the subclock in low-speed
Sleep (high-speed) mode	The CPU halts. On-chip peripheral functions are o the system clock
Sleep (medium-speed) mode	The CPU halts. On-chip peripheral functions oper frequency of 1/64, 1/32, 1/16, or 1/8 of the system frequency
Subsleep mode	The CPU halts. The time-base function of timer A, timer G, timer F,WDT, SCI3-1, SCI3-2, AEC and L controller/driver are operable on the subclock
Watch mode	The CPU halts. The time-base function of timer A, timer G, AEC and LCD controller/driver are operal subclock
Standby mode	The CPU and all on-chip peripheral functions halt
Module standby mode	Individual on-chip peripheral functions specified by enter standby mode and halt

Of these nine operating modes, all but the active (high-speed) mode are power-down m this section the two active modes (high-speed and medium speed) will be referred to c as active mode.



Mode Transition Conditions (1)

	LSON	MSON	SSBY	TMA3	DTON
a	0	0	0	*	0
b	0	1	0	*	0
c	1	*	0	1	0
d	0	*	1	0	0
e	*	*	1	1	0
f	0	0	0	*	1
g	0	1	0	*	1
h	0	1	1	1	1
i	1	*	1	1	1
J	0	0	1	1	1

* : Don't care

Mode Transition Conditions (2)

	Interrupt Sources
1	Timer A, Timer F, Timer G interrupt, IRQ
	WKP_7 to WKP_0 interrupts
2	Timer A, Timer C, Timer F, Timer G, SCI
	SCI3-2 interrupt, IRQ_4 to IRQ_0 interrupts,
	WKP_7 to WKP_0 interrupts, AEC
3	All interrupts
4	IRQ_1 or IRQ_0 interrupt, WKP_7 to WKP_0 in

Notes: 1. A transition between different modes cannot be made to occur simply because an interrupt request is generated. Make sure that interrupt handling is performed after the interrupt is accepted.
2. Details on the mode transition conditions are given in the explanations of each mode, in sections 5-2 through 5-8.

Figure 5.1 Mode Transition Diagram

	I/O ports							
External interrupts	IRQ_0	Functions	Functions	Functions	Functions	Functions	Functions	Functions
	IRQ_1					Retained[*6]		
	IRQ_2							
	IRQ_3							
	IRQ_4							
	WKP_0	Functions	Functions	Functions	Functions	Functions	Functions	Functions
	WKP_1							
	WKP_2							
	WKP_3							
	WKP_4							
	WKP_5							
	WKP_6							
	WKP_7							
Peripheral functions	Timer A	Functions	Functions	Functions	Functions	Functions[*5]	Functions[*5]	Functions
	Asynchronous counter					Functions[*8]	Functions	Functions
	Timer C					Retained	Functions/ Retained[*2]	Functions Retained
	WDT						Functions/ Retained[*7]	Retained
	Timer G, Timer F					Functions/ Retained[*9]	Functions/ Retained[*9]	Functions Retained
	SCI3-1					Reset	Functions/ Retained[*3]	Functions
	SCI3-2							
	PWM					Retained	Retained	Retained
	A/D converter					Retained	Retained	Retained
	LCD					Functions/ Retained[*4]	Functions/ Retained[*4]	Functions Retained

Notes:
1. Register contents are retained, but output is high-impedance state.
2. Functions if an external clock or the $\phi_W/4$ internal clock is selected; otherwise halted and retained
3. Functions if $\phi_W/2$ is selected as the internal clock; otherwise halted and retained.
4. Functions if ϕ_W, $\phi_W/2$ or $\phi_W/4$ is selected as the operating clock; otherwise halted and retained.
5. Functions if the timekeeping time-base function is selected.
6. External interrupt requests are ignored. Interrupt request register contents are not altered.
7. Functions if $\phi_W/32$ is selected as the internal clock; otherwise halted and retained.
8. Incrementing is possible, but interrupt generation is not.
9. Functions if an external clock or the $\phi_W/4$ internal clock is selected; otherwise halted and retained

1. System control register 1 (SYSCR1)

Bit	7	6	5	4	3	2	1
	SSBY	STS2	STS1	STS0	LSON	—	MA1
Initial value	0	0	0	0	0	1	1
Read/Write	R/W	R/W	R/W	R/W	R/W	—	R/W

SYSCR1 is an 8-bit read/write register for control of the power-down modes.

Upon reset, SYSCR1 is initialized to H'07.

Bit 7: Software standby (SSBY)

This bit designates transition to standby mode or watch mode.

Bit 7 SSBY	**Description**
0	• When a SLEEP instruction is executed in active mode, (ir a transition is made to sleep mode • When a SLEEP instruction is executed in subactive mode, a transition subsleep mode
1	• When a SLEEP instruction is executed in active mode, a transition is n standby mode or watch mode • When a SLEEP instruction is executed in subactive mode, a transition watch mode

0	0	0	Wait time = 8,192 states	(
0	0	1	Wait time = 16,384 states	
0	1	0	Wait time = 32,768 states	
0	1	1	Wait time = 65,536 states	
1	0	0	Wait time = 131,072 states	
1	0	1	Wait time = 2 states	(External
1	1	0	Wait time = 8 states	
1	1	1	Wait time = 16 states	

Note: In the case that external clock is input, set up the “Standby timer select” selecti
External clock mode before Mode Transition. Also, do not set up to external clo
the case that it does not use external clock.

Bit 3: Low speed on flag (LSON)

This bit chooses the system clock (ø) or subclock ($ø_{SUB}$) as the CPU operating clock w
mode is cleared. The resulting operation mode depends on the combination of other c
and interrupt input.

Bit 3 LSON	Description	
0	The CPU operates on the system clock (ø)	(
1	The CPU operates on the subclock (øSUB)	

Bits 2: Reserved bits

Bit 2 is reserved: it is always read as 1 and cannot be modified.

0	1	$ø_{osc}$/32	
1	0	$ø_{osc}$/64	
1	1	$ø_{osc}$/128	(in

2. System control register 2 (SYSCR2)

Bit	7	6	5	4	3	2	1	
	—	—	—	NESEL	DTON	MSON	SA1	
Initial value	1	1	1	1	0	0	0	
Read/Write	—	—	—	R/W	R/W	R/W	R/W	

SYSCR2 is an 8-bit read/write register for power-down mode control.

Bits 7 to 5: Reserved bits

These bits are reserved; they are always read as 1, and cannot be modified.

Bit 4: Noise elimination sampling frequency select (NESEL)

This bit selects the frequency at which the watch clock signal ($ø_W$) generated by the sul pulse generator is sampled, in relation to the oscillator clock ($ø_{OSC}$) generated by the sy pulse generator. When $ø_{OSC}$ = 2 to 6 MHz, clear NESEL to 0.

Bit 4 NESEL	Description	
0	Sampling rate is $ø_{OSC}$/16	
1	Sampling rate is $ø_{OSC}$/4	(in

0	• When a SLEEP instruction is executed in active mode, a transition is made to standby mode, watch mode, or sleep mode • When a SLEEP instruction is executed in subactive mode, a transitio watch mode or subsleep mode
1	• When a SLEEP instruction is executed in active (high-speed) mode, a transition is made to active (medium-speed) mode if SSBY = 0, MSO LSON = 0, or to subactive mode if SSBY = 1, TMA3 = 1, and LSON = • When a SLEEP instruction is executed in active (medium-speed) mo transition is made to active (high-speed) mode if SSBY = 0, MSON = LSON = 0, or to subactive mode if SSBY = 1, TMA3 = 1, and LSON = • When a SLEEP instruction is executed in subactive mode, a direct tra made to active (high-speed) mode if SSBY = 1, TMA3 = 1, LSON = 0 = 0, or to active (medium-speed) mode if SSBY = 1, TMA3 = 1, LSO MSON = 1

Bit 2: Medium speed on flag (MSON)

After standby, watch, or sleep mode is cleared, this bit selects active (high-speed) or a (medium-speed) mode.

Bit 2 MSON	Description
0	Operation in active (high-speed) mode
1	Operation in active (medium-speed) mode

1	*	$ø_W/2$

*

in SYSCR2 are cleared to 0. In sleep mode CPU operation is halted but the on-chip pe
functions. CPU register contents are retained.

2. Transition to sleep (medium-speed) mode

The system goes from active mode to sleep (medium-speed) mode when a SLEEP inst
executed while the SSBY and LSON bits in SYSCR1 are cleared to 0, the MSON bit i
is set to 1, and the DTON bit in SYSCR2 is cleared to 0. In sleep (medium-speed) mo
sleep (high-speed) mode, CPU operation is halted but the on-chip peripheral functions
operational. The clock frequency in sleep (medium-speed) mode is determined by the
MA0 bits in SYSCR1. CPU register contents are retained.
Furthermore, it sometimes acts with half state early timing at the time of transition to s
(medium-speed) mode.

5.2.2 Clearing Sleep Mode

Sleep mode is cleared by any interrupt (timer A, timer C, timer F, timer G, asynchrono
IRQ_4 to IRQ_0, WKP_7 to WKP_0, SCI3-1, SCI3-2, A/D converter, or), or by input at the

- Clearing by interrupt

When an interrupt is requested, sleep mode is cleared and interrupt exception handling
transition is made from sleep (high-speed) mode to active (high-speed) mode, or from
(medium-speed) mode to active (medium-speed) mode. Sleep mode is not cleared if th
condition code register (CCR) is set to 1 or the particular interrupt is disabled in the in
enable register.
Interrupt signal and system clock are mutually asynchronous. Synchronization error t
maximum is 2/ø (s).

- Clearing by $\overline{\text{RES}}$ input

When the $\overline{\text{RES}}$ pin goes low, the CPU goes into the reset state and sleep mode is clear

The system goes from active mode to standby mode when a SLEEP instruction is execu
the SSBY bit in SYSCR1 is set to 1, the LSON bit in SYSCR1 is cleared to 0, and bit T
TMA is cleared to 0. In standby mode the clock pulse generator stops, so the CPU and
peripheral modules stop functioning, but as long as the rated voltage is supplied, the co
CPU registers, on-chip RAM, and some on-chip peripheral module registers are retaine
RAM contents will be further retained down to a minimum RAM data retention voltage
ports go to the high-impedance state.

5.3.2 Clearing Standby Mode

Standby mode is cleared by an interrupt (IRQ_1 or IRQ_0), WKP_7 to WKP_0 or by input a
pin.

- Clearing by interrupt

When an interrupt is requested, the system clock pulse generator starts. After the time
STS2 to STS0 in SYSCR1 has elapsed, a stable system clock signal is supplied to the e
standby mode is cleared, and interrupt exception handling starts. Operation resumes in
(high-speed) mode if MSON = 0 in SYSCR2, or active (medium-speed) mode if MSON
Standby mode is not cleared if the I bit of CCR is set to 1 or the particular interrupt is d
the interrupt enable register.

- Clearing by $\overline{RES}$ input

When the $\overline{RES}$ pin goes low, the system clock pulse generator starts. After the pulse ge
output has stabilized, if the $\overline{RES}$ pin is driven high, the CPU starts reset exception hand
system clock signals are supplied to the entire chip as soon as the system clock pulse ge
starts functioning, the $\overline{RES}$ pin should be kept at the low level until the pulse generator
stabilizes.

Table 5.4 Clock Frequency and Settling Time (times are in ms)

STS2	STS1	STS0	Waiting Time	2 MHz	1 MHz
0	0	0	8,192 states	4.1	8.2
0	0	1	16,384 states	8.2	16.4
0	1	0	32,768 states	16.4	32.8
0	1	1	65,536 states	32.8	65.5
1	0	0	131,072 states	65.5	131.1
1	0	1	2 states (Use prohibited)	0.001	0.002
1	1	0	8 states	0.004	0.008
1	1	1	16 states	0.008	0.016

- When an external clock is used

STS2 = 1, STS1 = 0, and STS0 = 1 should be set. Other values possible use, but CPU will start operation before waiting time completion.

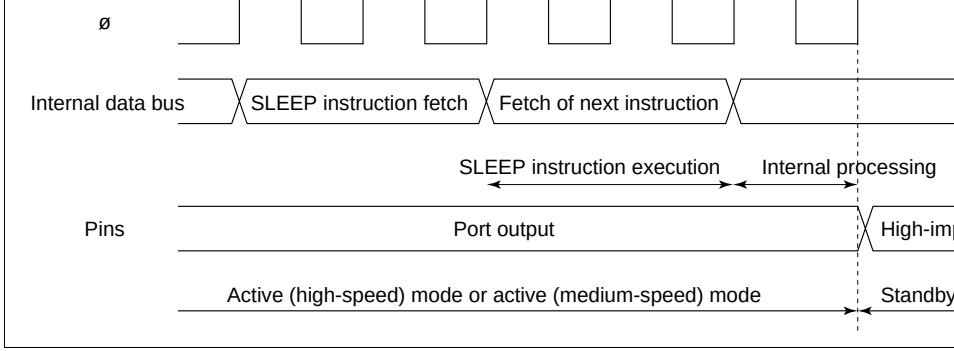


Figure 5.2 Standby Mode Transition and Pin States

timer G, AEC and the LCD controller/driver (for which operation or halting can be se
As long as a minimum required voltage is applied, the contents of CPU registers, the c
RAM and some registers of the on-chip peripheral modules, are retained. I/O ports ke
states as before the transition.

5.4.2 Clearing Watch Mode

Watch mode is cleared by an interrupt (timer A, timer F, timer G, IRQ_0, or WKP_7 to V
input at the $\overline{RES}$ pin.

- Clearing by interrupt

When watch mode is cleared by interrupt, the mode to which a transition is made depe
settings of LSON in SYSCR1 and MSON in SYSCR2. If both LSON and MSON are
transition is to active (high-speed) mode; if LSON = 0 and MSON = 1, transition is to
(medium-speed) mode; if LSON = 1, transition is to subactive mode. When the transit
active mode, after the time set in SYSCR1 bits STS2 to STS0 has elapsed, a stable clo
supplied to the entire chip, watch mode is cleared, and interrupt exception handling sta
mode is not cleared if the I bit of CCR is set to 1 or the particular interrupt is disabled
interrupt enable register.

- Clearing by $\overline{RES}$ input

Clearing by $\overline{RES}$ pin is the same as for standby mode; see 2. Clearing by $\overline{RES}$ pin in 5
Clearing Standby Mode.

5.4.3 Oscillator Settling Time after Watch Mode is Cleared

The waiting time is the same as for standby mode; see 5.3.3, Oscillator Settling Time
Standby Mode is Cleared.

are retained. I/O ports keep the same states as before the transition.

5.5.2 Clearing Subsleep Mode

Subsleep mode is cleared by an interrupt (timer A, timer C, timer F, timer G, asynchron
counter, SCI3-2, SCI3-1, IRQ_4 to IRQ_0, WKP_7 to WKP_0) or by a low input at the $\overline{RES}$

- Clearing by interrupt

When an interrupt is requested, subsleep mode is cleared and interrupt exception handli
Subsleep mode is not cleared if the I bit of CCR is set to 1 or the particular interrupt is
the interrupt enable register.
Interrupt signal and system clock are mutually asynchronous. Synchronization error ti
maximum is 2/ø (s).

- Clearing by $\overline{RES}$ input

Clearing by $\overline{RES}$ pin is the same as for standby mode; see 2. Clearing by $\overline{RES}$ pin in 5.
Clearing Standby Mode.

does not take place if the I bit of CCR is set to 1 or the particular interrupt is disabled i
interrupt enable register.

5.6.2 Clearing Subactive Mode

Subactive mode is cleared by a SLEEP instruction or by a low input at the $\overline{\text{RES}}$ pin.

- Clearing by SLEEP instruction

If a SLEEP instruction is executed while the SSBY bit in SYSCR1 is set to 1 and TM
TMA is set to 1, subactive mode is cleared and watch mode is entered. If a SLEEP ins
executed while SSBY = 0 and LSON = 1 in SYSCR1 and TMA3 = 1 in TMA, subslee
entered. Direct transfer to active mode is also possible; see 5.8, Direct Transfer, belov

- Clearing by $\overline{\text{RES}}$ pin

Clearing by $\overline{\text{RES}}$ pin is the same as for standby mode; see 2. Clearing by $\overline{\text{RES}}$ pin in 5
Clearing Standby Mode.

5.6.3 Operating Frequency in Subactive Mode

The operating frequency in subactive mode is set in bits SA1 and SA0 in SYSCR2. T
are $\phi_W/2$, $\phi_W/4$, and $\phi_W/8$.

transition to active (medium-speed) mode does not take place if the I bit of CCR is set t
particular interrupt is disabled in the interrupt enable register.
Furthermore, it sometimes acts with half state early timing at the time of transition to ac
(medium-speed) mode.

5.7.2 Clearing Active (Medium-Speed) Mode

Active (medium-speed) mode is cleared by a SLEEP instruction.

- Clearing by SLEEP instruction

A transition to standby mode takes place if the SLEEP instruction is executed while the
in SYSCR1 is set to 1, the LSON bit in SYSCR1 is cleared to 0, and the TMA3 bit in T
cleared to 0. The system goes to watch mode if the SSBY bit in SYSCR1 is set to 1 and
in TMA is set to 1 when a SLEEP instruction is executed.

When both SSBY and LSON are cleared to 0 in SYSCR1 and a SLEEP instruction is ex
sleep mode is entered. Direct transfer to active (high-speed) mode or to subactive mode
possible. See 5.8, Direct Transfer, below for details.

- Clearing by $\overline{\text{RES}}$ pin

When the $\overline{\text{RES}}$ pin is driven low, a transition is made to the reset state and active (medi
mode is cleared.

5.7.3 Operating Frequency in Active (Medium-Speed) Mode

Operation in active (medium-speed) mode is clocked at the frequency designated by the
MA0 bits in SYSCR1.

If the direct transfer interrupt is disabled in interrupt enable register 2, a transition is m
to sleep mode or watch mode. Note that if a direct transition is attempted while the I b
set to 1, sleep mode or watch mode will be entered, and it will be impossible to clear t
mode by means of an interrupt.

- Direct transfer from active (high-speed) mode to active (medium-speed) mode

When a SLEEP instruction is executed in active (high-speed) mode while the SSBY a
bits in SYSCR1 are cleared to 0, the MSON bit in SYSCR2 is set to 1, and the DTON
SYSCR2 is set to 1, a transition is made to active (medium-speed) mode via sleep mo

- Direct transfer from active (medium-speed) mode to active (high-speed) mode

When a SLEEP instruction is executed in active (medium-speed) mode while the SSB
LSON bits in SYSCR1 are cleared to 0, the MSON bit in SYSCR2 is cleared to 0, and
bit in SYSCR2 is set to 1, a transition is made to active (high-speed) mode via sleep m

- Direct transfer from active (high-speed) mode to subactive mode

When a SLEEP instruction is executed in active (high-speed) mode while the SSBY a
bits in SYSCR1 are set to 1, the DTON bit in SYSCR2 is set to 1, and the TMA3 bit i
to 1, a transition is made to subactive mode via watch mode.

- Direct transfer from subactive mode to active (high-speed) mode

When a SLEEP instruction is executed in subactive mode while the SSBY bit in SYS
1, the LSON bit in SYSCR1 is cleared to 0, the MSON bit in SYSCR2 is cleared to 0,
bit in SYSCR2 is set to 1, and the TMA3 bit in TMA is set to 1, a transition is made d
active (high-speed) mode via watch mode after the waiting time set in SYSCR1 bits S
has elapsed.

SYSCR2 is set to 1, and the TMA3 bit in TMA is set to 1, a transition is made directly
(medium-speed) mode via watch mode after the waiting time set in SYSCR1 bits STS2
has elapsed.

5.8.2 Direct Transition Times

1. Time for direct transition from active (high-speed) mode to active (medium-speed)

A direct transition from active (high-speed) mode to active (medium-speed) mode is pe
executing a SLEEP instruction in active (high-speed) mode while bits SSBY and LSON
cleared to 0 in SYSCR1, and bits MSON and DTON are both set to 1 in SYSCR2. The
execution of the SLEEP instruction to the end of interrupt exception handling (the direc
time) is given by equation (1) below.

Direct transition time = { (Number of SLEEP instruction execution states) + (number o
processing states) } × (tcyc before transition) + (number of int
exception handling execution states) × (tcyc after transition)
.....

Example: Direct transition time = (2 + 1) × 2tosc + 14 × 16tosc = 230tosc (when ø/8 is
as the CPU operating clock)

Notation:
tosc: OSC clock cycle time
tcyc: System clock (ø) cycle time

processing states) } × (tcyc before transition) + (number of in
exception handling execution states) × (tcyc after transition)
.....

Example: Direct transition time = (2 + 1) × 16tosc + 14 × 2tosc = 76tosc (when ø/8 is
the CPU operating clock)

Notation:
tosc: OSC clock cycle time
tcyc: System clock (ø) cycle time

3. Time for direct transition from subactive mode to active (high-speed) mode

A direct transition from subactive mode to active (high-speed) mode is performed by e
SLEEP instruction in subactive mode while bit SSBY is set to 1 and bit LSON is clear
SYSCR1, bit MSON is cleared to 0 and bit DTON is set to 1 in SYSCR2, and bit TMA
in TMA. The time from execution of the SLEEP instruction to the end of interrupt ex
handling (the direct transition time) is given by equation (3) below.

Direct transition time = { (Number of SLEEP instruction execution states) + (number
processing states) } × (tsubcyc before transition) + { (wait tim
STS2 to STS0) + (number of interrupt exception handling exe
states) } × (tcyc after transition) .....

Example: Direct transition time = (2 + 1) × 8tw + (8192 + 14) × 2tosc = 24tw + 1641
øw/8 is selected as the CPU operating clock, and wait time = 8192 states)

Notation:
tosc: OSC clock cycle time
tw: Watch clock cycle time
tcyc: System clock (ø) cycle time
tsubcyc: Subclock (øSUB) cycle time

processing states) } × (tsubcyc before transition) + { (wait time
STS2 to STS0) + (number of interrupt exception handling exec
states) } × (tcyc after transition) .....

Example: Direct transition time = (2 + 1) × 8tw + (8192 + 14) × 16tosc = 24tw + 1312
(when øw/8 or ø8 is selected as the CPU operating clock, and wait time = 81

Notation:

tosc: OSC clock cycle time
tw: Watch clock cycle time
tcyc: System clock (ø) cycle time
tsubcyc: Subclock (øSUB) cycle time

Module standby mode is set for a particular module by setting the corresponding bit to stop register 1 (CKSTPR1) or clock stop register 2 (CKSTPR2). (See table 5.5.)

5.9.2 Clearing Module Standby Mode

Module standby mode is cleared for a particular module by setting the corresponding b clock stop register 1 (CKSTPR1) or clock stop register 2 (CKSTPR2). (See table 5.5.

Following a reset, clock stop register 1 (CKSTPR1) and clock stop register 2 (CKSTP initialized to H'FF.

Table 5.5

Register Name	Bit Name		Operation
CKSTPR1	TACKSTP	1	Timer A module standby mode is cleared
		0	Timer A is set to module standby mode
	TCCKSTP	1	Timer C module standby mode is cleared
		0	Timer C is set to module standby mode
	TFCKSTP	1	Timer F module standby mode is cleared
		0	Timer F is set to module standby mode
	TGCKSTP	1	Timer G module standby mode is cleared
		0	Timer G is set to module standby mode
	ADCKSTP	1	A/D converter module standby mode is c
		0	A/D converter is set to module standby m
	S32CKSTP	1	SCI3-2 module standby mode is cleared
		0	SCI3-2 is set to module standby mode
	S31CKSTP	1	SCI3-1 module standby mode is cleared
		0	SCI3-1 is set to module standby mode

	0	Watchdog timer is set to module standby r
AECKSTP	1	Asynchronous event counter module stand is cleared
	0	Asynchronous event counter is set to mod mode

Note: For details of module operation, see the sections on the individual modules.

byte data and word data. The H8/3867 and H8/3827 have a ZTAT™ version with 60-k
PROM.

6.1.1 Block Diagram

Figure 6.1 shows a block diagram of the on-chip ROM.

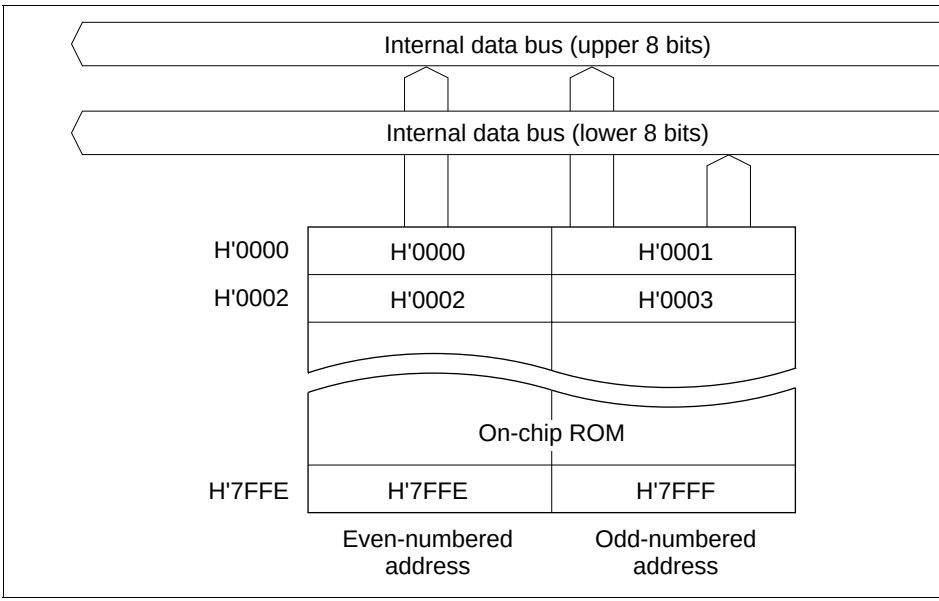


Figure 6.1 ROM Block Diagram (H8/3864 and H8/3824)

Table 6.1 Setting to PROM Mode

Pin Name	Setting
TEST	High level
PB_4/AN_4	Low level
PB_5/AN_5	
PB_6/AN_6	High level

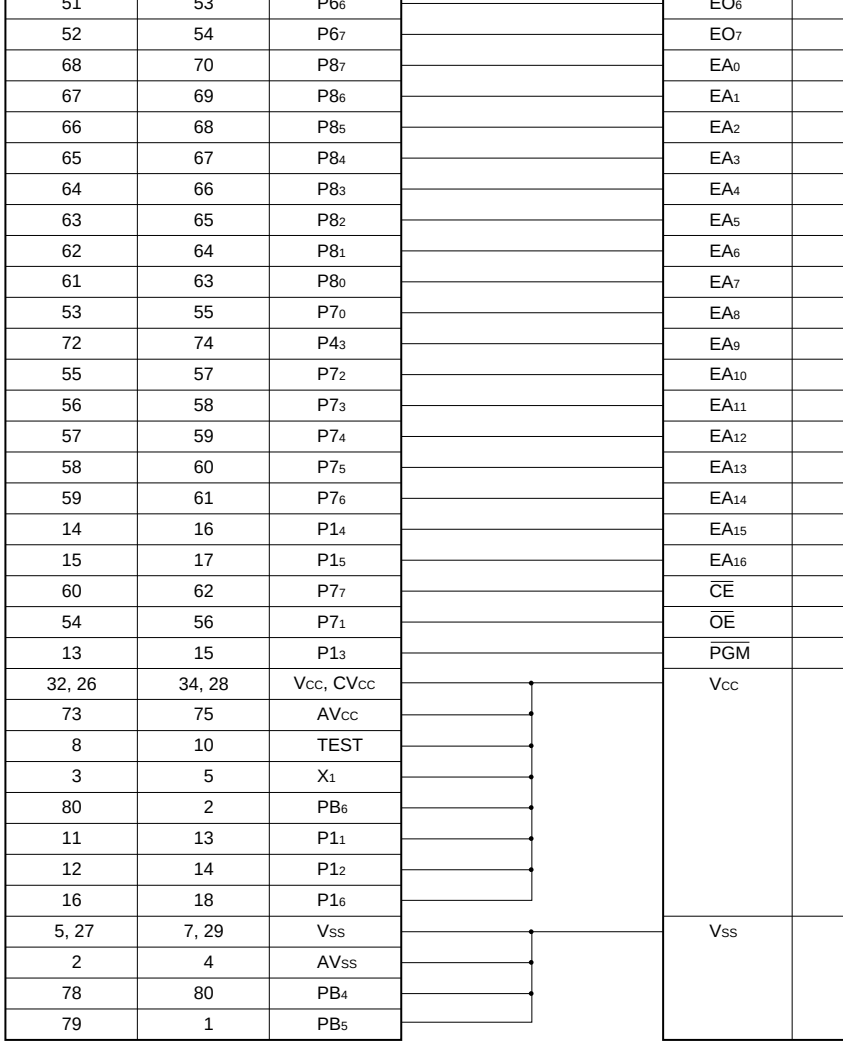
6.2.2 Socket Adapter Pin Arrangement and Memory Map

A standard PROM programmer can be used to program the PROM. A socket adapter is
for conversion to 32 pins, as listed in table 6.2.

Figure 6.2 shows the pin-to-pin wiring of the socket adapter. Figure 6.3 shows a memo

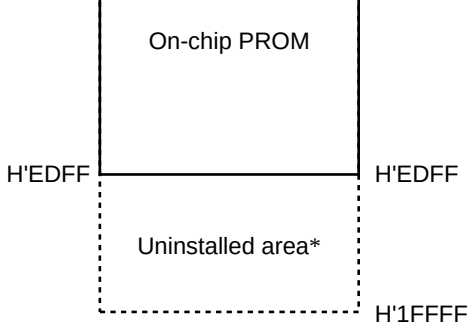
Table 6.2 Socket Adapter

Package	Socket Adapters (Manufacturer)
80-pin (FP-80B)	ME3867ESFS1H (MINATO) H7386BQ080D3201 (DATA-I/O)
80-pin (FP-80A)	ME3867ESHS1H (MINATO) H7386AQ080D3201 (DATA-I/O)
80-pin (TFP-80C)	ME3867ESNS1H (MINATO) H7386CT080D3201 (DATA-I/O)



Note: Pins not indicated in the figure should be left open.

Figure 6.2 Socket Adapter Pin Correspondence (with HN27C101)



Note: * The output data is not guaranteed if this address area is read in PROM mo
fore, when programming with a PROM programmer, be sure to specify addr
H'0000 to H'EDFF. If programming is inadvertently performed from H'EE0
may not be possible to continue PROM programming and verification.
When programming, H'FF should be set as the data in this address area
H'1FFFF).

Figure 6.3 H8/3867 and H8/3827 Memory Map in PROM Mode

Write	L	H	L	V_{PP}	V_{CC}	Data input	Add
Verify	L	L	H	V_{PP}	V_{CC}	Data output	Add
Programming disabled	L	L	L	V_{PP}	V_{CC}	High impedance	Add
	L	H	H				
	H	L	L				
	H	H	H				

Notation

L: Low level

H: High level

V_{PP}: V_{PP} level

V_{CC}: V_{CC} level

The specifications for writing and reading are identical to those for the standard HN27 EPROM. However, page programming is not supported, and so page programming mo be set. A PROM programmer that only supports page programming mode cannot be u selecting a PROM programmer, ensure that it supports high-speed, high-reliability byt programming. Also, be sure to specify addresses from H'0000 to H'EDFF.

6.3.1 Writing and Verifying

An efficient, high-speed, high-reliability method is available for writing and verifying data. This method achieves high speed without voltage stress on the device and withou the reliability of written data. The basic flow of this high-speed, high-reliability progra method is shown in figure 6.4.

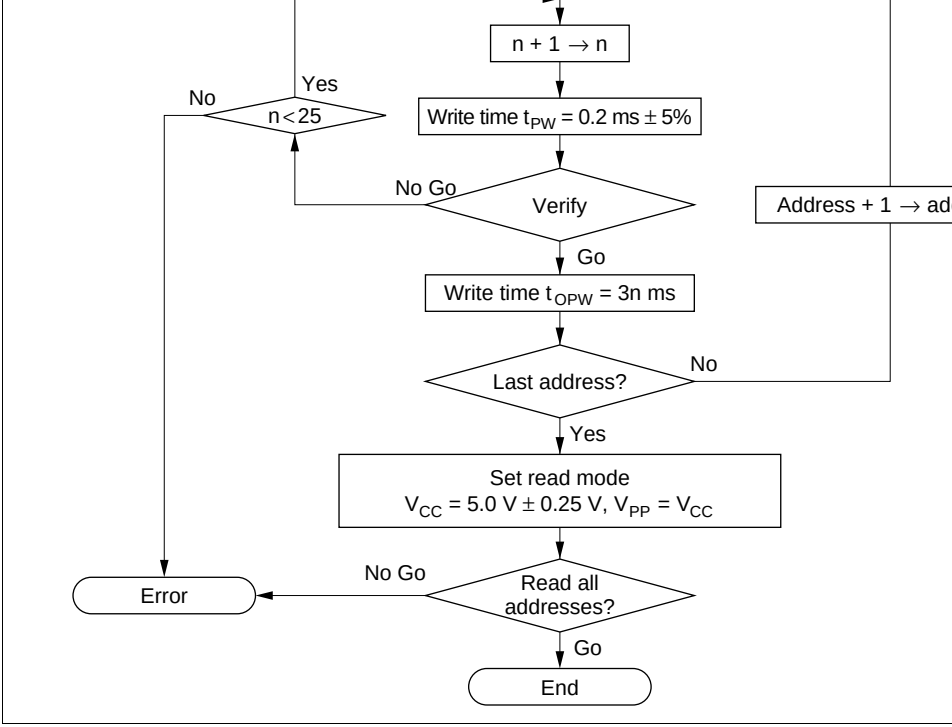


Figure 6.4 High-Speed, High-Reliability Programming Flow Chart

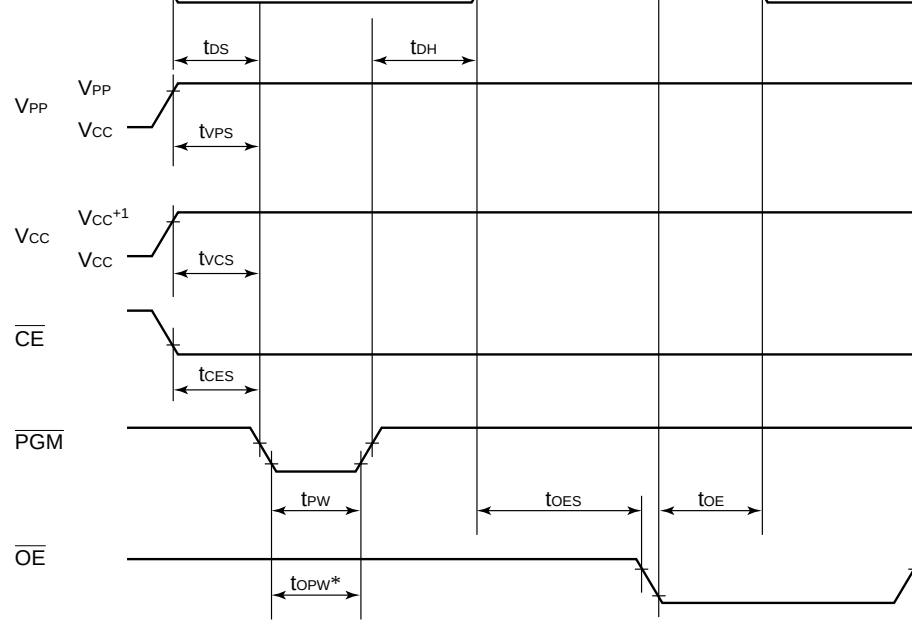
Input low-level voltage	$\overline{EO_7}$ to $\overline{EO_0}$, EA_{16} to EA_0 $\overline{OE}$, $\overline{CE}$, $\overline{PGM}$	V_{IL}	–0.3	—	0.8	V	
Output high-level voltage	$\overline{EO_7}$ to $\overline{EO_0}$	V_{OH}	2.4	—	—	V	I_O
Output low-level voltage	$\overline{EO_7}$ to $\overline{EO_0}$	V_{OL}	—	—	0.45	V	I_O
Input leakage current	$\overline{EO_7}$ to $\overline{EO_0}$, EA_{16} to EA_0 $\overline{OE}$, $\overline{CE}$, $\overline{PGM}$	$\|IL_I\|$	—	—	2	µA	V_i 0.
V_{CC} current		I_{CC}	—	—	40	mA	
V_{PP} current		I_{PP}	—	—	40	mA	

Address hold time	t_{AH}	0			µs
Data hold time	t_{DH}	2	—	—	µs
Data output disable time	t_{DF}[*2]	—	—	130	ns
V_{PP} setup time	t_{VPS}	2	—	—	µs
Programming pulse width	t_{PW}	0.19	0.20	0.21	ms
$\overline{PGM}$ pulse width for overwrite programming	t_{OPW}[*3]	0.19	—	5.25	ms
$\overline{CE}$ setup time	t_{CES}	2	—	—	µs
V_{CC} setup time	t_{VCS}	2	—	—	µs
Data output delay time	t_{OE}	0	—	200	ns

Notes: 1. Input pulse level: 0.45 V to 2.2 V
Input rise time/fall time 20 ns
Timing reference levels Input: 0.8 V, 2.0 V
Output: 0.8 V, 2.0 V

2. t_{DF} is defined at the point at which the output is floating and the output level
read.

3. t_{OPW} is defined by the value given in figure 6.4, High-Speed, High-Reliability
Programming Flow Chart.



Note: * t_{opw} is defined by the value shown in figure 6.4, High-Speed, High-Reliability Programm

Figure 6.5 PROM Write/Verify Timing

- Make sure the index marks on the PROM programmer socket, socket adapter, and c
  properly aligned. If they are not, the chip may be destroyed by excessive current flo
  programming, be sure that the chip is properly mounted in the PROM programmer.
- Avoid touching the socket adapter or chip while programming, since this may cause
  faults and write errors.
- Take care when setting the programming mode, as page programming is not suppor
- When programming with a PROM programmer, be sure to specify addresses from H
  H'EDFF. If programming is inadvertently performed from H'EE00 onward, it may
  possible to continue PROM programming and verification. When programming, H'
  be set as the data in address area H'EE00 to H'1FFFF.

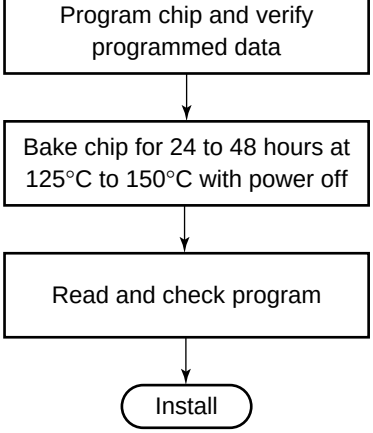


Figure 6.6 Recommended Screening Procedure

If a series of programming errors occurs while the same PROM programmer is in use, programming and check the PROM programmer and socket adapter for defects. Pleas Hitachi of any abnormal conditions noted during or after programming or in screening data after high-temperature baking.

7.1.1 Block Diagram

Figure 7.1 shows a block diagram of the on-chip RAM.

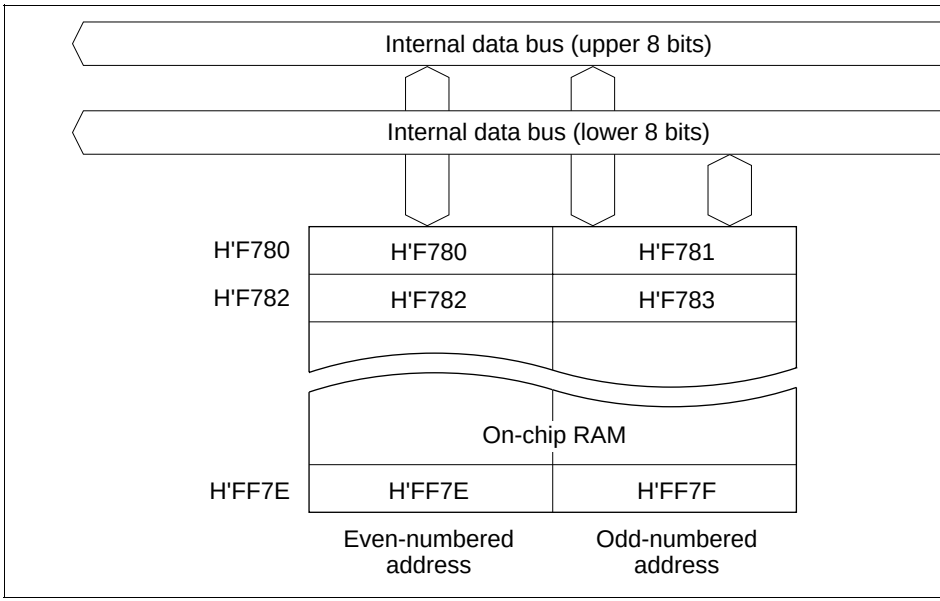


Figure 7.1 RAM Block Diagram (H8/3864 and H8/3824)

register (PDR) for storing output data. Input or output can be assigned to individual b
See 2.9.2, Notes on Bit Manipulation, for information on executing bit-manipulation i
to write data in PCR or PDR.

Ports 5, 6, 7, 8, and A are also used as liquid crystal display segment and common pin
in 8-bit units.

Block diagrams of each port are given in Appendix C, I/O Port Block Diagrams.

Table 8.1 Port Functions

Port	Description	Pins	Other Functions	F S R
Port 1	• 8-bit I/O port • MOS input pull-up option	$P1_7$ to $P1_5$/ $\overline{IRQ_3}$ to $\overline{IRQ_1}$/ TMIF, TMIC	External interrupts 3 to 1 Timer event interrupts TMIF, TMIC	P T T
		$P1_4/\overline{IRQ_4}/\overline{ADTRG}$	External interrupt 4 and A/D converter external trigger	P
		$P1_3$/TMIG	Timer G input capture input	P
		$P1_2$, $P1_1$/ TMOFH, TMOFL	Timer F output compare output	P
		$P1_0$/TMOW	Timer A clock output	P
Port 3	• 8-bit I/O port • MOS input pull-up option • Large-current port	$P3_7$/AEVL $P3_6$/AEVH $P3_5/TXD_{31}$ $P3_4/RXD_{31}$ $P3_3/SCK_{31}$	SCI3-1 data output (TXD_{31}), data input (RXD_{31}), clock input/output (SCK_{31}), and asynchronous counter event inputs AEVL, AEVH	P S S
		$P3_2/\overline{RESO}$ $P3_1$/UD $P3_0$/PWM	Reset output, timer C count-up/down select input, and 14-bit PWM output	P

Port 5	• 8-bit I/O port • MOS input pull-up option	$P5_7$ to $P5_0$/ $\overline{WKP_7}$ to $\overline{WKP_0}$/ SEG_8 to SEG_1	Wakeup input ($\overline{WKP_7}$ to $\overline{WKP_0}$), segment output (SEG_8 to SEG_1)	PM LP
Port 6	• 8-bit I/O port • MOS input pull-up option	$P6_7$ to $P6_0$/ SEG_{16} to SEG_9	Segment output (SEG_{16} to SEG_9)	LP
Port 7	• 8-bit I/O port	$P7_7$ to $P7_0$/ SEG_{24} to SEG_{17}	Segment output (SEG_{24} to SEG_{17})	LP
Port 8	• 8-bit I/O port	$P8_7$/SEG_{32}/CL_1 $P8_6$/SEG_{31}/CL_2 $P8_5$/SEG_{30}/DO $P8_4$/SEG_{29}/M $P8_3$ to $P8_0$/ SEG_{28} to SEG_{25}	Segment output (SEG_{32} to SEG_{25}) Segment external expansion latch clock (CL_1), shift clock (CL_2), display data (DO), alternation signal (M)	LP
Port A	4-bit I/O port	PA_3 to PA_0/ COM_4 to COM_1	Common output (COM_4 to COM_1)	LP
Port B	8-bit input port	PB_7 to PB_0/ AN_7 to AN_0	A/D converter analog input	AM

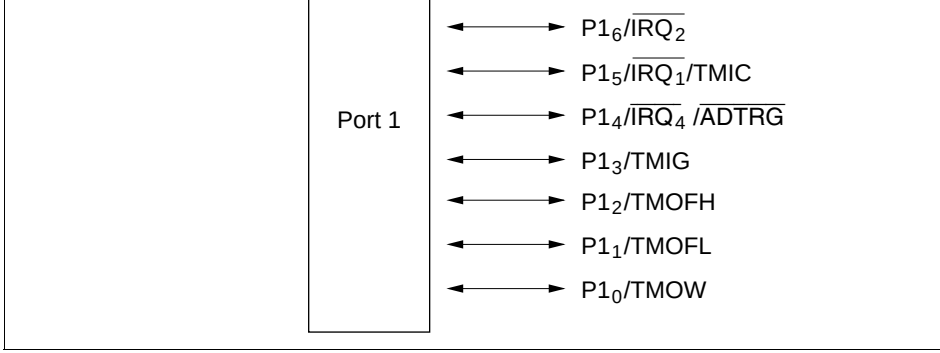


Figure 8.1 Port 1 Pin Configuration

8.2.2 Register Configuration and Description

Table 8.2 shows the port 1 register configuration.

Table 8.2 Port 1 Registers

Name	Abbrev.	R/W	Initial Value
Port data register 1	PDR1	R/W	H'00
Port control register 1	PCR1	W	H'00
Port pull-up control register 1	PUCR1	R/W	H'00
Port mode register 1	PMR1	R/W	H'00

bits are set to 1, the values stored in PDR1 are read, regardless of the actual pin states. read while PCR1 bits are cleared to 0, the pin states are read.

Upon reset, PDR1 is initialized to H'00.

2. Port control register 1 (PCR1)

Bit	7	6	5	4	3	2	1
	$PCR1_7$	$PCR1_6$	$PCR1_5$	$PCR1_4$	$PCR1_3$	$PCR1_2$	$PCR1_1$
Initial value	0	0	0	0	0	0	0
Read/Write	W	W	W	W	W	W	W

PCR1 is an 8-bit register for controlling whether each of the port 1 pins $P1_7$ to $P1_0$ func input pin or output pin. Setting a PCR1 bit to 1 makes the corresponding pin an output clearing the bit to 0 makes the pin an input pin. The settings in PCR1 and in PDR1 are when the corresponding pin is designated in PMR1 as a general I/O pin.

Upon reset, PCR1 is initialized to H'00.

PCR1 is a write-only register, which is always read as all 1s.

a PCR1 bit is cleared to 0, setting the corresponding PUCR1 bit to 1 turns on the MOS
the corresponding pin, while clearing the bit to 0 turns off the MOS pull-up.

Upon reset, PUCR1 is initialized to H'00.

4. Port mode register 1 (PMR1)

Bit	7	6	5	4	3	2	1
	IRQ3	IRQ2	IRQ1	IRQ4	TMIG	TMOFH	TMOFL
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

PMR1 is an 8-bit read/write register, controlling the selection of pin functions for port

Upon reset, PMR1 is initialized to H'00.

Bit 7: $P1_7/\overline{IRQ}_3$/TMIF pin function switch (IRQ3)

This bit selects whether pin $P1_7/\overline{IRQ}_3$/TMIF is used as $P1_7$ or as $\overline{IRQ}_3$/TMIF.

Bit 7 IRQ3	**Description**
0	Functions as $P1_7$ I/O pin
1	Functions as $\overline{IRQ}_3$/TMIF input pin

Note: Rising or falling edge sensing can be designated for IRQ_3, TMIF. For details or settings, see 3. Timer Control Register F (TCRF) in 9.4.2.

Bit 5: $P1_5/\overline{IRQ_1}$/TMIC pin function switch (IRQ1)

This bit selects whether pin $P1_5/\overline{IRQ_1}$/TMIC is used as $P1_5$ or as $\overline{IRQ_1}$/TMIC.

Bit 5 IRQ1	Description	
0	Functions as $P1_5$ I/O pin	(ir
1	Functions as $\overline{IRQ_1}$/TMIC input pin	

Note: Rising or falling edge sensing can be designated for $\overline{IRQ_1}$/TMIC.
For details of TMIC pin setting, see 1. Timer mode register C (TMC) in 9.3.2.

Bit 4: $P1_4/\overline{IRQ_4}/\overline{ADTRG}$ pin function switch (IRQ4)

This bit selects whether pin $P1_4/\overline{IRQ_4}/\overline{ADTRG}$ is used as $P1_4$ or as $\overline{IRQ_4}/\overline{ADTRG}$.

Bit 4 IRQ4	Description	
0	Functions as $P1_4$ I/O pin	(ir
1	Functions as $\overline{IRQ_4}/\overline{ADTRG}$ input pin	

Note: For details of $\overline{ADTRG}$ pin setting, see 12.3.2, Start of A/D Conversion by Externa

Bit 3: $P1_3$/TMIG pin function switch (TMIG)

This bit selects whether pin $P1_3$/TMIG is used as $P1_3$ or as TMIG.

Bit 3 TMIG	Description	
0	Functions as $P1_3$ I/O pin	(ir
1	Functions as TMIG input pin	

Bit 1: $P1_1$/TMOFL pin function switch (TMOFL)

This bit selects whether pin $P1_1$/TMOFL is used as $P1_1$ or as TMOFL.

Bit 1 TMOFL	Description
0	Functions as $P1_1$ I/O pin
1	Functions as TMOFL output pin

Bit 0: $P1_0$/TMOW pin function switch (TMOW)

This bit selects whether pin $P1_0$/TMOW is used as $P1_0$ or as TMOW.

Bit 0 TMOW	Description
0	Functions as $P1_0$ I/O pin
1	Functions as TMOW output pin

IRQ_3	0		1	
$PCR1_7$	0	1	*	
CKSL2 to CKSL0	*		Not 0**	
Pin function	$P1_7$ input pin	$P1_7$ output pin	$\overline{IRQ}_3$ input pin	$\overline{I}$

Note: When this pin is used as the TMIF input pin, clear bit IEN3 to 0 disable the IRQ_3 interrupt.

$P1_6/\overline{IRQ}_2$

The pin function depends on bits IRQ2 in PMR1 and bit $PCR1_6$ in PC

IRQ2	0		1
$PCR1_6$	0	1	*
Pin function	$P1_6$ input pin	$P1_6$ output pin	$\overline{IRQ}_2$ input

$P1_5/\overline{IRQ}_1$
TMIC

The pin function depends on bit IRQ1 in PMR1, bits TMC2 to TMC0 i bit $PCR1_5$ in PCR1.

IRQ1	0		1	
$PCR1_5$	0	1	*	
TMC2 to TMC0	*		Not 111	
Pin function	$P1_5$ input pin	$P1_5$ output pin	$\overline{IRQ}_1$ input pin	$\overline{I}$

Note: When this pin is used as the TMIC input pin, clear bit IEN1 to 0 to disable the IRQ_1 interrupt.

$P1_4/\overline{IRQ}_4$
$\overline{ADTRG}$

The pin function depends on bit IRQ4 in PMR1, bit TRGE in AMR, an in PCR1.

IRQ4	0		1	
$PCR1_4$	0	1	*	
TRGE	*		0	
Pin function	$P1_4$ input pin	$P1_4$ output pin	$\overline{IRQ}_4$ input pin	$\overline{IR}$

Note: When this pin is used as the $\overline{ADTRG}$ input pin, clear bit IEN4 to IENR1 to disable the IRQ_4 interrupt.

$P1_2$/TMOFH	The pin function depends on bit TMOFH in PMR1 and bit $PCR1_2$ in P		
	TMOFH	0	1
	$PCR1_2$	0	1	*
	Pin function	$P1_2$ input pin	$P1_2$ output pin	TMOFH out
$P1_1$/TMOFL	The pin function depends on bit TMOFL in PMR1 and bit $PCR1_1$ in P		
	TMOFL	0	1
	$PCR1_1$	0	1	*
	Pin function	$P1_1$ input pin	$P1_1$ output pin	TMOFL out
$P1_0$/TMOW	The pin function depends on bit TMOW in PMR1 and bit $PCR1_0$ in P		
	TMOW	0	1
	$PCR1_0$	0	1	*
	Pin function	$P1_0$ input pin	$P1_0$ output pin	TMOW outp

$P1_5/\overline{IRQ_1}/TMIC$	state	state	state
$P1_4/\overline{IRQ_4}/\overline{ADTRG}$			
$P1_3$/TMIG			
$P1_2$/TMOFH			
$P1_1$/TMOFL			
$P1_0$/TMOW			

Note: * A high-level signal is output when the MOS pull-up is in the on state.

8.2.5 MOS Input Pull-Up

Port 1 has a built-in MOS input pull-up function that can be controlled by software. W
PCR1 bit is cleared to 0, setting the corresponding PUCR1 bit to 1 turns on the MOS in
for that pin. The MOS input pull-up function is in the off state after a reset.

$PCR1_n$	0	0	1
$PUCR1_n$	0	1	*
MOS input pull-up	Off	On	Off

*

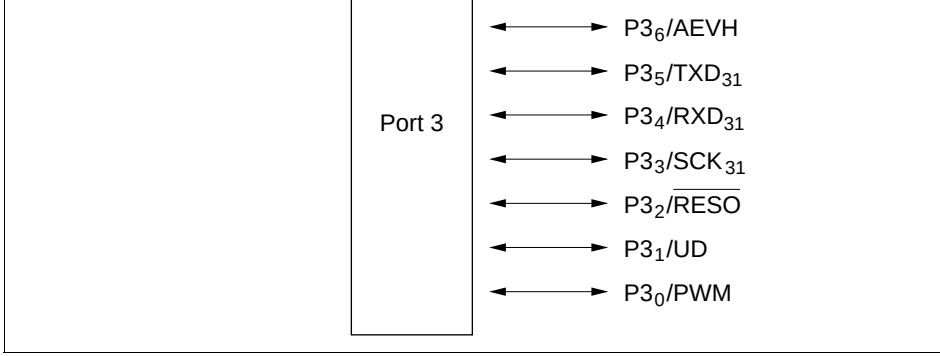


Figure 8.2 Port 3 Pin Configuration

8.3.2 Register Configuration and Description

Table 8.5 shows the port 3 register configuration.

Table 8.5 Port 3 Registers

Name	Abbrev.	R/W	Initial Value
Port data register 3	PDR3	R/W	H'00
Port control register 3	PCR3	W	H'00
Port pull-up control register 3	PUCR3	R/W	H'00
Port mode register 3	PMR3	R/W	H'04

bits are set to 1, the values stored in PDR3 are read, regardless of the actual pin states. read while PCR3 bits are cleared to 0, the pin states are read.

Upon reset, PDR3 is initialized to H'00.

2. Port control register 3 (PCR3)

Bit	7	6	5	4	3	2	1
	$PCR3_7$	$PCR3_6$	$PCR3_5$	$PCR3_4$	$PCR3_3$	$PCR3_2$	$PCR3_1$
Initial value	0	0	0	0	0	0	0
Read/Write	W	W	W	W	W	W	W

PCR3 is an 8-bit register for controlling whether each of the port 3 pins $P3_7$ to $P3_0$ func input pin or output pin. Setting a PCR3 bit to 1 makes the corresponding pin an output clearing the bit to 0 makes the pin an input pin. The settings in PCR3 and in PDR3 are when the corresponding pin is designated in PMR3 as a general I/O pin.

Upon reset, PCR3 is initialized to H'00.

PCR3 is a write-only register, which is always read as all 1s.

3. Port pull-up control register 3 (PUCR3)

Bit	7	6	5	4	3	2	1
	$PUCR3_7$	$PUCR3_6$	$PUCR3_5$	$PUCR3_4$	$PUCR3_3$	$PUCR3_2$	$PUCR3_1$
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

PUCR3 controls whether the MOS pull-up of each of the port 3 pins $P3_7$ to $P3_0$ is on or a PCR3 bit is cleared to 0, setting the corresponding PUCR3 bit to 1 turns on the MOS the corresponding pin, while clearing the bit to 0 turns off the MOS pull-up.

Upon reset, PUCR3 is initialized to H'00.

Upon reset, PMR3 is initialized to H'04.

Bit 7: $P3_7$/AEVL pin function switch (AEVL)

This bit selects whether pin $P3_7$/AEVL is used as $P3_7$ or as AEVL.

Bit 7 AEVL	Description
0	Functions as $P3_7$ I/O pin
1	Functions as AEVL input pin

Bit 6: $P3_6$/AEVH pin function switch (AEVH)

This bit selects whether pin $P3_6$/AEVH is used as $P3_6$ or as AEVH.

Bit 6 AEVH	Description
0	Functions as $P3_6$ I/O pin
1	Functions as AEVH input pin

Bit 5: Watchdog timer source clock select (WDCKS)

This bit selects the watchdog timer source clock.

Bit 5 WDCKS	Description
0	ø/8192 selected
1	øw/32 selected

Bit 3: $P4_3/\overline{IRQ_0}$ pin function switch (IRQ0)

This bit selects whether pin $P4_3/\overline{IRQ_0}$ is used as $P4_3$ or as $\overline{IRQ_0}$.

Bit 3 IRQ0	Description	
0	Functions as $P4_3$ input pin	(ir
1	Functions as $\overline{IRQ_0}$ input pin	

Bit 2: $P3_2/\overline{RESO}$ pin function switch (RESO)

This bit selects whether pin $P3_2/\overline{RESO}$ is used as $P3_2$ or as $\overline{RESO}$.

Bit 2 RESO	Description	
0	Functions as $P3_2$ I/O pin	
1	Functions as $\overline{RESO}$ output pin	(ir

Bit 1: $P3_1$/UD pin function switch (SI1)

This bit selects whether pin $P3_1$/UD is used as $P3_1$ or as UD.

Bit 1 UD	Description	
0	Functions as $P3_1$ I/O pin	(ir
1	Functions as UD input pin	

8.3.3 Pin Functions

Table 8.9 shows the port 3 pin functions.

Table 8.9 Port 3 Pin Functions

Pin	Pin Functions and Selection Method			
$P3_7$/AEVL	The pin function depends on bit SO1 in PMR3 and bit $PCR3_2$ in PCR			
	AEVL	0		
	$PCR3_7$	0	1	
	Pin function	$P3_7$ input pin	$P3_7$ output pin	AEV
$P3_6$/AEVH	The pin function depends on bit AEVH in PMR3 and bit PCR36 in P			
	AEVH	0		
	$PCR3_6$	0	1	
	Pin function	$P3_6$ input pin	$P3_6$ output pin	AEV
$P3_5$/TXD_{31}	The pin function depends on bit TE in SCR3-1, bit SPC31 in SPCR, $PCR3_5$ in PCR3.			
	SPC31	0		
	TE	0		
	$PCR3_5$	0	1	
	Pin function	$P3_5$ input pin	$P3_5$ output pin	TXD_3

$P3_3/SCK_{31}$ The pin function depends on bits CKE1, CKE0, and SMR31 in SCR3 and $PCR3_3$ in PCR3.

CKE1	0			
CKE0	0		1	
$COM3_1$	0		1	*
$PCR3_3$	0	1	*	
Pin function	$P3_3$ input pin	$P3_3$ output pin	SCK_{31} output pin	S

$P3_2/\overline{RESO}$ The pin function depends on bit RESO in PMR3 and bit $PCR3_2$ in PC

RESO	0		
$PCR3_2$	0	1	
Pin function	$P3_2$ input pin	$P3_2$ output pin	RESO

$P3_1/UD$ The pin function depends on bit UD in PMR3 and bit $PCR3_1$ in PCR3

UD	0		
$PCR3_1$	0	1	
Pin function	$P3_1$ input pin	$P3_1$ output pin	UD i

$P3_0/PWM$ The pin function depends on bit PWM in PMR3 and bit $PCR3_0$ in PCR

PWM	0		
$PCR3_0$	0	1	
Pin function	$P3_0$ input pin	$P3_0$ output pin	PWM

*

$P3_5/TXD_{31}$ $P3_4/RXD_{31}$ $P3_3/SCK_{31}$		state	state	state
$P3_2/\overline{RESO}$	RESO output			
$P3_1/UD$ $P3_0/PWM$	High-impedance			

Note: * A high-level signal is output when the MOS pull-up is in the on state.

8.3.5 MOS Input Pull-Up

Port 3 has a built-in MOS input pull-up function that can be controlled by software. W
PCR3 bit is cleared to 0, setting the corresponding PUCR3 bit to 1 turns on the MOS p
that pin. The MOS pull-up function is in the off state after a reset.

$PCR3_n$	0	0	1
$PUCR3_n$	0	1	*
MOS input pull-up	Off	On	Off

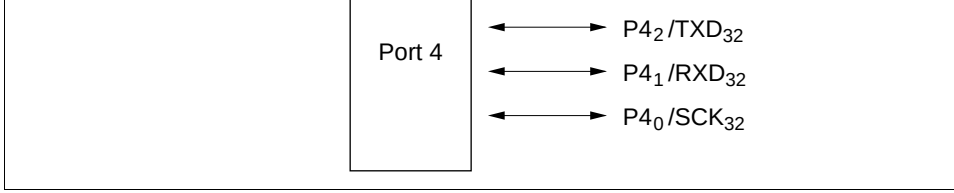


Figure 8.3 Port 4 Pin Configuration

8.4.2 Register Configuration and Description

Table 8.8 shows the port 4 register configuration.

Table 8.8 Port 4 Registers

Name	Abbrev.	R/W	Initial Value
Port data register 4	PDR4	R/W	H'F8
Port control register 4	PCR4	W	H'F8

1. Port data register 4 (PDR4)

Bit	7	6	5	4	3	2	1
	—	—	—	—	P4$_3$	P4$_2$	P4$_1$
Initial value	1	1	1	1	1	0	0
Read/Write	—	—	—	—	R	R/W	R/W

PDR4 is an 8-bit register that stores data for port 4 pins P4$_2$ to P4$_0$. If port 4 is read wh
bits are set to 1, the values stored in PDR4 are read, regardless of the actual pin states.
read while PCR4 bits are cleared to 0, the pin states are read.

Upon reset, PDR4 is initialized to H'F8.

input pin or output pin. Setting a PCR4 bit to 1 makes the corresponding pin an output
clearing the bit to 0 makes the pin an input pin. PCR4 and PDR4 settings are valid wh
corresponding pins are designated for general-purpose input/output by SCR3-2.

Upon reset, PCR4 is initialized to H'F8.

PCR4 is a write-only register, which always reads all 1s.

8.4.3 Pin Functions

Table 8.9 shows the port 4 pin functions.

Table 8.9 Port 4 Pin Functions

Pin	Pin Functions and Selection Method		
$P4_3/\overline{IRQ}_0$	The pin function depends on bit IRQ0 in PMR3.		
	IRQ0	0	1
	Pin function	$P4_3$ input pin	$\overline{IRQ}_0$ inpu
$P4_2/TXD_{32}$	The pin function depends on bit TE in SCR3-2, bit SPC32 in SPCR, $PCR4_2$ in PCR4.		
	SPC32	0	
	TE	0	
	$PCR4_2$	0	1
	Pin function	$P4_2$ input pin	$P4_2$ output pin
$P4_1/RXD_{32}$	The pin function depends on bit RE in SCR3-2 and bit $PCR4_1$ in PC		
	RE_{32}	0	
	$PCR4_1$	0	1
	Pin function	$P4_1$ input pin	$P4_1$ output pin

$PCR4_0$	0	1	*	
Pin function	$P4_0$ input pin	$P4_0$ output pin	SCK_{32} output pin	S

8.4.4 Pin States

Table 8.10 shows the port 4 pin states in each operating mode.

Table 8.10 Port 4 Pin States

Pins	Reset	Sleep	Subsleep	Standby	Watch	Subactive
$P4_3/IRQ_0$ $P4_2/TXD_{32}$ $P4_1/RXD_{32}$ $P4_0/SCK_{32}$	High-impedance	Retains previous state	Retains previous state	High-impedance	Retains previous state	Functional

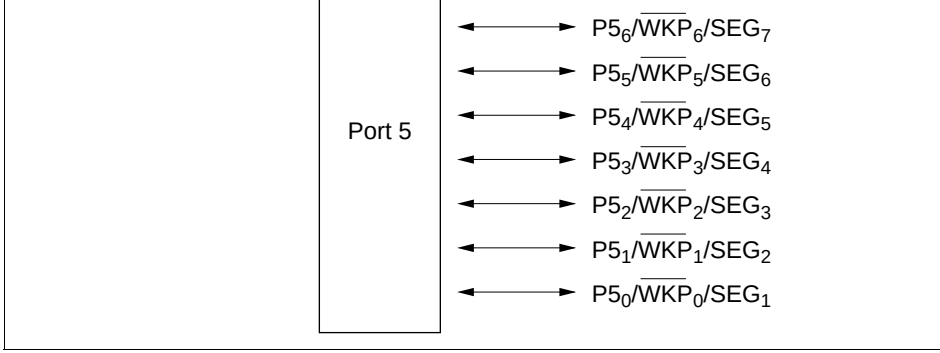


Figure 8.4 Port 5 Pin Configuration

8.5.2 Register Configuration and Description

Table 8.11 shows the port 5 register configuration.

Table 8.11 Port 5 Registers

Name	Abbrev.	R/W	Initial Value
Port data register 5	PDR5	R/W	H'00
Port control register 5	PCR5	W	H'00
Port pull-up control register 5	PUCR5	R/W	H'00
Port mode register 5	PMR5	R/W	H'00

bits are set to 1, the values stored in PDR5 are read, regardless of the actual pin states. read while PCR5 bits are cleared to 0, the pin states are read.

Upon reset, PDR5 is initialized to H'00.

2. Port control register 5 (PCR5)

Bit	7	6	5	4	3	2	1
	$PCR5_7$	$PCR5_6$	$PCR5_5$	$PCR5_4$	$PCR5_3$	$PCR5_2$	$PCR5_1$
Initial value	0	0	0	0	0	0	0
Read/Write	W	W	W	W	W	W	W

PCR5 is an 8-bit register for controlling whether each of the port 5 pins $P5_7$ to $P5_0$ func input pin or output pin. Setting a PCR5 bit to 1 makes the corresponding pin an output clearing the bit to 0 makes the pin an input pin. PCR5 and PDR5 settings are valid whe corresponding pins are designated for general-purpose input/output by PMR5 and bits S SGS0 in LPCR.

Upon reset, PCR5 is initialized to H'00.

PCR5 is a write-only register, which is always read as all 1s.

3. Port pull-up control register 5 (PUCR5)

Bit	7	6	5	4	3	2	1
	$PUCR5_7$	$PUCR5_6$	$PUCR5_5$	$PUCR5_4$	$PUCR5_3$	$PUCR5_2$	$PUCR5_1$
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

PUCR5 controls whether the MOS pull-up of each of port 5 pins $P5_7$ to $P5_0$ is on or off PCR5 bit is cleared to 0, setting the corresponding PUCR5 bit to 1 turns on the MOS pu the corresponding pin, while clearing the bit to 0 turns off the MOS pull-up.

Upon reset, PUCR5 is initialized to H'00.

Upon reset, PMR5 is initialized to H'00.

Bit n: $P5_n/\overline{WKP}_n/SEG_{n+1}$ pin function switch (WKPn)

When pin P5n/$\overline{WKP}$n/SEGn+1 is not used as SEG_{n+1}, these bits select whether the pi
P5n or $\overline{WKP}_n$.

Bit n WKPn	Description	
0	Functions as P5n I/O pin	(
1	Functions as $\overline{WKP}_n$ input pin	

Note: For use as SEG_{n+1}, see 13.2.1, LCD Port Control Register (LPCR).

$P5_0/\overline{WKP}_0/SEG_1$					(n
	SGS3 to SGS0	0***			
	WKP_n	0		1	
	$PCR5_n$	0	1	*	
	Pin function	$P5_n$ input pin	$P5_n$ output pin	$\overline{WKPn}$ input pin	

*

8.5.4 Pin States

Table 8.13 shows the port 5 pin states in each operating mode.

Table 8.13 Port 5 Pin States

Pins	Reset	Sleep	Subsleep	Standby	Watch	Subactive
$P5_7/\overline{WKP}_7/SEG_8$ to $P5_0/\overline{WKP}_0/SEG_1$	High-impedance	Retains previous state	Retains previous state	High-impedance*	Retains previous state	Functional

Note: * A high-level signal is output when the MOS pull-up is in the on state.

MOS input pull-up	Off	On	Off

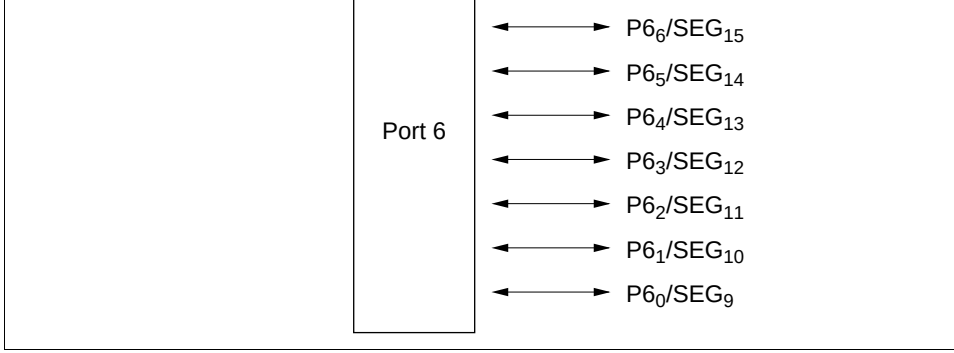


Figure 8.5 Port 6 Pin Configuration

8.6.2 Register Configuration and Description

Table 8.14 shows the port 6 register configuration.

Table 8.14 Port 6 Registers

Name	Abbrev.	R/W	Initial Value
Port data register 6	PDR6	R/W	H'00
Port control register 6	PCR6	W	H'00
Port pull-up control register 6	PUCR6	R/W	H'00

If port 6 is read while PCR6 bits are set to 1, the values stored in PDR6 are read, regar
actual pin states. If port 6 is read while PCR6 bits are cleared to 0, the pin states are r

Upon reset, PDR6 is initialized to H'00.

2. Port control register 6 (PCR6)

Bit	7	6	5	4	3	2	1
	$PCR6_7$	$PCR6_6$	$PCR6_5$	$PCR6_4$	$PCR6_3$	$PCR6_2$	$PCR6_1$
Initial value	0	0	0	0	0	0	0
Read/Write	W	W	W	W	W	W	W

PCR6 is an 8-bit register for controlling whether each of the port 6 pins $P6_7$ to $P6_0$ fur
input pin or output pin.

Setting a PCR6 bit to 1 makes the corresponding pin ($P6_7$ to $P6_0$) an output pin, while
bit to 0 makes the pin an input pin. PCR6 and PDR6 settings are valid when the corre
pins are designated for general-purpose input/output by bits SGS3 to SGS0 in LPCR.

Upon reset, PCR6 is initialized to H'00.

PCR6 is a write-only register, which always reads all 1s.

a PCR6 bit is cleared to 0, setting the corresponding PUCR6 bit to 1 turns on the MOS
the corresponding pin, while clearing the bit to 0 turns off the MOS pull-up.

Upon reset, PUCR6 is initialized to H'00.

8.6.3 Pin Functions

Table 8.15 shows the port 6 pin functions.

Table 8.15 Port 6 Pin Functions

Pin	Pin Functions and Selection Method		
$P6_7/SEG_{16}$ to $P6_0/SEG_9$	The pin function depends on bit PCR6n in PCR6 and bits SGS3 to SG LPCR. (n =		
	SEG3 to SEGS0	00**, 010*	011*
	$PCR6_n$	0	1
	Pin function	$P6_n$ input pin	$P6_n$ output pin	SEG_{n+9}

*

8.6.4 Pin States

Table 8.16 shows the port 6 pin states in each operating mode.

Table 8.16 Port 6 Pin States

Pin	Reset	Sleep	Subsleep	Standby	Watch	Subactive
$P6_7/SEG_{16}$ to $P6_0/SEG_9$	High-impedance	Retains previous state	Retains previous state	High-impedance*	Retains previous state	Functional

Note: * A high-level signal is output when the MOS pull-up is in the on state.

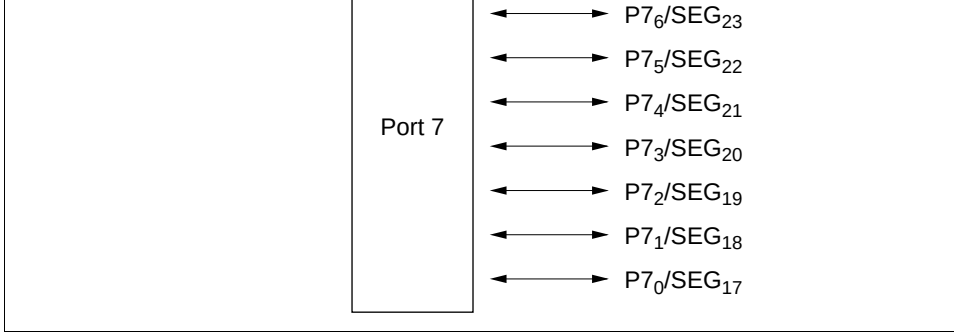


Figure 8.6 Port 7 Pin Configuration

8.7.2 Register Configuration and Description

Table 8.17 shows the port 7 register configuration.

Table 8.17 Port 7 Registers

Name	Abbrev.	R/W	Initial Value
Port data register 7	PDR7	R/W	H'00
Port control register 7	PCR7	W	H'00

bits are set to 1, the values stored in PDR7 are read, regardless of the actual pin states. read while PCR7 bits are cleared to 0, the pin states are read.

Upon reset, PDR7 is initialized to H'00.

2. Port control register 7 (PCR7)

Bit	7	6	5	4	3	2	1
	$PCR7_7$	$PCR7_6$	$PCR7_5$	$PCR7_4$	$PCR7_3$	$PCR7_2$	$PCR7_1$
Initial value	0	0	0	0	0	0	0
Read/Write	W	W	W	W	W	W	W

PCR7 is an 8-bit register for controlling whether each of the port 7 pins $P7_7$ to $P7_0$ fun input pin or output pin. Setting a PCR7 bit to 1 makes the corresponding pin an outpu clearing the bit to 0 makes the pin an input pin. PCR7 and PDR7 settings are valid wh corresponding pins are designated for general-purpose input/output by bits SGS3 to SG LPCR.

Upon reset, PCR7 is initialized to H'00.

PCR7 is a write-only register, which always reads as all 1s.

SEGS3 to SEGS0	00**		01*
$PCR7_n$	0	1	
Pin function	$P7_n$ input pin	$P7_n$ output pin	SEG_{n+1}

*

8.7.4 Pin States

Table 8.19 shows the port 7 pin states in each operating mode.

Table 8.19 Port 7 Pin States

Pins	Reset	Sleep	Subsleep	Standby	Watch	Subactive
$P7_7/SEG_{24}$ to $P7_0/SEG_{17}$	High-impedance	Retains previous state	Retains previous state	High-impedance	Retains previous state	Functional

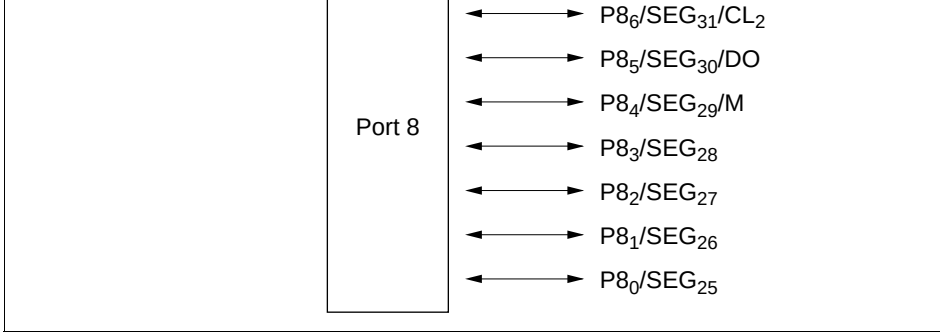


Figure 8.7 Port 8 Pin Configuration

8.8.2 Register Configuration and Description

Table 8.20 shows the port 8 register configuration.

Table 8.20 Port 8 Registers

Name	Abbrev.	R/W	Initial Value
Port data register 8	PDR8	R/W	H'00
Port control register 8	PCR8	W	H'00

bits are set to 1, the values stored in PDR8 are read, regardless of the actual pin states. read while PCR8 bits are cleared to 0, the pin states are read.

Upon reset, PDR8 is initialized to H'00.

2. Port control register 8 (PCR8)

Bit	7	6	5	4	3	2	1
	$PCR8_7$	$PCR8_6$	$PCR8_5$	$PCR8_4$	$PCR8_3$	$PCR8_2$	$PCR8_1$
Initial value	0	0	0	0	0	0	0
Read/Write	W	W	W	W	W	W	W

PCR8 is an 8-bit register for controlling whether each of the port 8 pins $P8_7$ to $P8_0$ func input or output pin. Setting a PCR8 bit to 1 makes the corresponding pin an output pin, clearing the bit to 0 makes the pin an input pin. PCR8 and PDR8 settings are valid whe corresponding pins are designated for general-purpose input/output by bits SGS3 to SG LPCR.

Upon reset, PCR8 is initialized to H'00.

PCR8 is a write-only register, which is always read as all 1s.

SEGS3 to SEGS0	000*		001*, 01**, 1***	
SGX	0		0	
$PCR8_7$	0	1	*	
Pin function	$P8_7$ input pin	$P8_7$ output pin	SEG_{32} output pin	C

$P8_6/SEG_{31}/CL_2$ The pin function depends on bit $PCR8_6$ in PCR8 and bits SGX and SGS in LPCR.

SEGS3 to SEGS0	000*		001*, 01**, 1***	
SGX	0		0	
$PCR8_6$	0	1	*	
Pin function	$P8_6$ input pin	$P8_6$ output pin	SEG_{31} output pin	C

$P8_5/SEG_{30}/DO$ The pin function depends on bit $PCR8_5$ in PCR8 and bits SGX and SGS in LPCR.

SEGS3 to SEGS0	000*		001*, 01**, 1***	
SGX	0		0	
$PCR9_5$	0	1	*	
Pin function	$P8_5$ input pin	$P8_5$ output pin	SEG_{30} output pin	D

$P8_4/SEG_{29}/M$ The pin function depends on bit $PCR8_4$ in PCR8 and bits SGX and SGS in LPCR.

SEGS3 to SEGS0	000*		001*, 01**, 1***	
SGX	0		0	
$PCR9_4$	0	1	*	
Pin function	$P8_4$ input pin	$P8_4$ output pin	SEG_{29} output pin	M

$P8_3/SEG_{28}$ to $P8_0/SEG_{25}$ The pin function depends on bit $PCR8_n$ in PCR8 and bits SGS3 to SGS (n

SEGS3 to SEGS0	000*		001*,
$PCR8_n$	0	1	
Pin function	$P8_n$ input pin	$P8_n$ output pin	SEG_{n+2}

$P8_4/SEG_{29}/M$ $P8_3/SEG_{28}$ to $P8_0/SEG_{25}$			

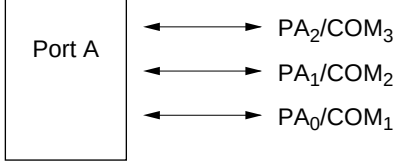


Figure 8.8 Port A Pin Configuration

8.9.2 Register Configuration and Description

Table 8.23 shows the port A register configuration.

Table 8.23 Port A Registers

Name	Abbrev.	R/W	Initial Value
Port data register A	PDRA	R/W	H'F0
Port control register A	PCRA	W	H'F0

1. Port data register A (PDRA)

Bit	7	6	5	4	3	2	1
	—	—	—	—	PA_3	PA_2	PA_1
Initial value	1	1	1	1	0	0	0
Read/Write	—	—	—	—	R/W	R/W	R/W

PDRA is an 8-bit register that stores data for port A pins PA_3 to PA_0. If port A is read PCRA bits are set to 1, the values stored in PDRA are read, regardless of the actual pi port A is read while PCRA bits are cleared to 0, the pin states are read.

Upon reset, PDRA is initialized to H'F0.

Setting a PCRA bit to 1 makes the corresponding pin an output pin, while clearing the t
makes the pin an input pin. PCRA and PDRA settings are valid when the correspondin
designated for general-purpose input/output by LPCR.

Upon reset, PCRA is initialized to H'F0.

PCRA is a write-only register, which always reads all 1s.

$PCRA_3$	0	1	
Pin function	PA_3 input pin	PA_3 output pin	COM

PA_2/COM_3 The pin function depends on bit $PCRA_2$ in PCRA and bits SGS3 to S

SEGS3 to SEGS0	0000	0000	N
$PCRA_2$	0	1	
Pin function	PA_2 input pin	PA_2 output pin	COM

PA_1/COM_2 The pin function depends on bit $PCRA_1$ in PCRA and bits SGS3 to S

SEGS3 to SEGS0	0000	0000	N
$PCRA_1$	0	1	
Pin function	PA_1 input pin	PA_1 output pin	COM

PA_0/COM_1 The pin function depends on bit $PCRA_0$ in PCRA and bits SGS3 to S

SEGS3 to SEGS0	0000		N
$PCRA_0$	0	1	
Pin function	PA_0 input pin	PA_0 output pin	COM

PA_1/COM_2 PA_0/COM_1	state	state	state

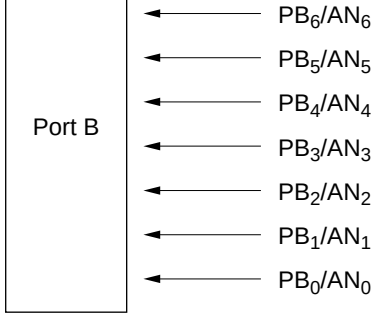


Figure 8.9 Port B Pin Configuration

8.10.2 Register Configuration and Description

Table 8.26 shows the port B register configuration.

Table 8.26 Port B Register

Name	Abbrev.	R/W	Address
Port data register B	PDRB	R	H'FFDE

Port Data Register B (PDRB)

Bit	7	6	5	4	3	2	1
	PB_7	PB_6	PB_5	PB_4	PB_3	PB_2	PB_1
Read/Write	R	R	R	R	R	R	R

Reading PDRB always gives the pin states. However, if a port B pin is selected as an
channel for the A/D converter by AMR bits CH3 to CH0, that pin reads 0 regardless o
voltage.

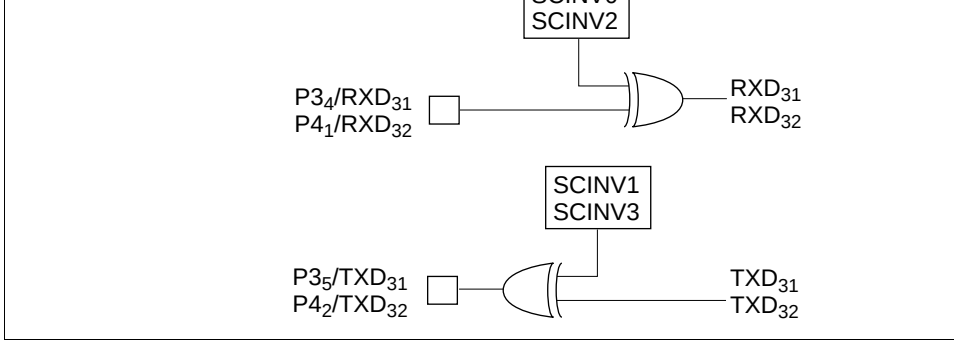


Figure 8.10 Input/Output Data Inversion Function

8.11.2 Register Configuration and Descriptions

Table 8.27 shows the registers used by the input/output data inversion function.

Table 8.27 Register Configuration

Name	Abbreviation	R/W	Address
Serial port control register	SPCR	R/W	H'FF91

Serial Port Control Register (SPCR)

Bit	7	6	5	4	3	2	1	
	—	—	SPC32	SPC31	SCINV3	SCINV2	SCINV1	S
Initial value	1	1	0	0	0	0	0	
Read/Write	—	—	R/W	R/W	R/W	R/W	R/W	

SPCR is an 8-bit readable/writable register that performs RXD$_{31}$, RXD$_{32}$, TXD$_{31}$, and
input/output data inversion switching. SPCR is initialized to H'C0 by a reset.

Bit 1: TXD_{31} pin output data inversion switch

Bit 1 specifies whether or not TXD_{31} pin output data is to be inverted.

Bit 1 SCINV1	Description
0	TXD_{31} output data is not inverted
1	TXD_{31} output data is inverted

Bit 2: RXD_{32} pin input data inversion switch

Bit 2 specifies whether or not RXD_{32} pin input data is to be inverted.

Bit 2 SCINV2	Description
0	RXD_{32} input data is not inverted
1	RXD_{32} input data is inverted

Bit 3: TXD_{32} pin output data inversion switch

Bit 3 specifies whether or not TXD_{32} pin output data is to be inverted.

Bit 3 SCINV3	Description
0	TXD_{32} output data is not inverted
1	TXD_{32} output data is inverted

Bit 5: $P4_2/TXD_{32}$ pin function switch (SPC32)

This bit selects whether pin $P4_2/TXD_{32}$ is used as $P4_2$ or as TXD_{32}.

Bit 5 SPC32	Description	
0	Functions as $P4_2$ I/O pin	(in
1	Functions as TXD_{32} output pin*	

Note: * Set the TE bit in SCR3 after setting this bit to 1.

Bits 7 and 6: Reserved bits

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

8.11.3 Note on Modification of Serial Port Control Register

When a serial port control register is modified, the data being input or output up to that inverted immediately after the modification, and an invalid data change is input or outp modifying a serial port control register, do so in a state in which data changes are invali

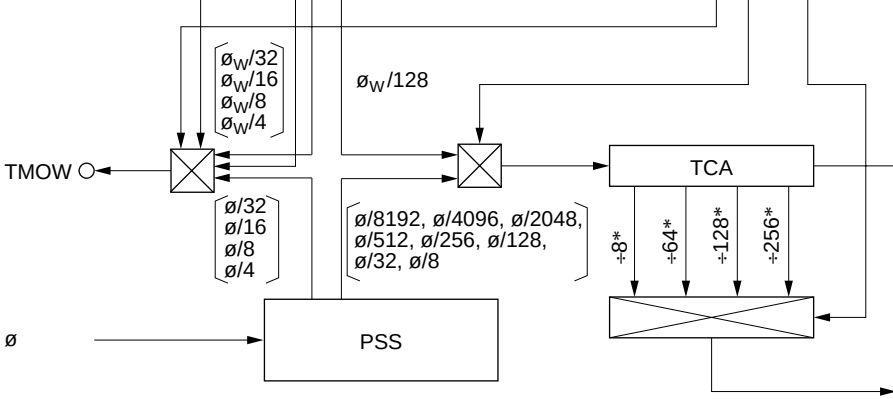
Name	Functions	Internal Clock	Event Input Pin	Waveform Output Pin	Rem
Timer A	• 8-bit interval timer	ø/8 to ø/8192	—	—	
	• Interval function	(8 choices)			
	• Time base	$ø_w$/128 (choice of 4 overflow periods)			
	• Clock output	ø/4 to ø/32 $ø_w$, $ø_w$/4 to $ø_w$/32 (9 choices)	—	TMOW	
Timer C	• 8-bit timer • Interval function • Event counting function • Up-count/down-count selectable	ø/4 to ø/8192, $ø_w$/4 (7 choices)	TMIC	—	Up-c coun cont softw hard
Timer F	16-bit timer Event counting function Also usable as two independent8-bit timers Output compare output function	ø/4 to ø/32, $ø_w$/4 (4 choices)	TMIF	TMOFL TMOFH	
Timer G	• 8-bit timer • Input capture function • Interval function	ø/2 to ø/64, $ø_w$/4 (4 choices)	TMIG	—	• C c c • E c s c

nous event counter	• Also usable as two independent 8-bit counters • Counts events asynchronous to ø and øw	AEVH

1. Features

Features of timer A are given below.

- Choice of eight internal clock sources (ø/8192, ø/4096, ø/2048, ø/512, ø/256, ø/128
- Choice of four overflow periods (1 s, 0.5 s, 0.25 s, 31.25 ms) when timer A is used
  time base (using a 32.768 kHz crystal oscillator).
- An interrupt is requested when the counter overflows.
- Any of nine clock signals can be output at the TMOW pin: 32.768 kHz divided by
  4 (1 kHz, 2 kHz, 4 kHz, 8 kHz) or 38.4 kHz divided by 32, 16, 8, or 4 (1.2 kHz, 2.4
  kHz, 9.6 kHz), and the system clock divided by 32, 16, 8, or 4.
- Use of module standby mode enables this module to be placed in standby mode ind
  when not used.



Notation:

TMA:	Timer mode register A
TCA:	Timer counter A
IRRTA:	Timer A overflow interrupt request flag
PSW:	Prescaler W
PSS:	Prescaler S
CWOSR:	Subclock output select register

Note: * Can be selected only when the prescaler W output ($ø_W$/128) is used as the TCA input

Figure 9.1 Block Diagram of Timer A

3. Pin configuration

Table 9.2 shows the timer A pin configuration.

Table 9.2 Pin Configuration

Name	Abbrev.	I/O	Function
Clock output	TMOW	Output	Output of waveform generated by timer A outp

Clock stop register 1	CKSTPR1	R/W	H'FF
Subclock output select register	CWOSR	R/W	H'FE

9.2.2 Register Descriptions

1. Timer mode register A (TMA)

Bit	7	6	5	4	3	2	1
	TMA7	TMA6	TMA5	—	TMA3	TMA2	TMA1
Initial value	0	0	0	1	0	0	0
Read/Write	R/W	R/W	R/W	—	R/W	R/W	R/W

TMA is an 8-bit read/write register for selecting the prescaler, input clock, and output

Upon reset, TMA is initialized to H'10.

CWOS	TMA7	TMA6	TMA5	Clock Output	
0	0	0	0	ø/32	(ir
			1	ø/16	
		1	0	ø/8	
			1	ø/4	
	1	0	0	$ø_w$/32	
			1	$ø_w$/16	
		1	0	$ø_w$/8	
			1	$ø_w$/4	
1	*	*	*	$ø_w$	

*

Bit 4: Reserved bit

Bit 4 is reserved; it is always read as 1, and cannot be modified.

		1	0	PSS, ø/2048	
			1	PSS, ø/512	
	1	0	0	PSS, ø/256	
			1	PSS, ø/128	
		1	0	PSS, ø/32	
			1	PSS, ø/8	
1	0	0	0	PSW, 1 s	Clo
			1	PSW, 0.5 s	bas
		1	0	PSW, 0.25 s	(wh
			1	PSW, 0.03125 s	32.
	1	0	0	PSW and TCA are reset	
			1		
		1	0		
			1		

source for input to this counter is selected by bits TMA3 to TMA0 in timer mode regist
(TMA). TCA values can be read by the CPU in active mode, but cannot be read in sub
mode. When TCA overflows, the IRRTA bit in interrupt request register 1 (IRR1) is se

TCA is cleared by setting bits TMA3 and TMA2 of TMA to 11.

Upon reset, TCA is initialized to H'00.

3. Clock stop register 1 (CKSTPR1)

Bit:	7	6	5	4	3	2	1
	—	S31CKSTP	S32CKSTP	ADCKSTP	TGCKSTP	TFCKSTP	TCCKSTP
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

CKSTPR1 is an 8-bit read/write register that performs module standby mode control fo
modules. Only the bit relating to timer A is described here. For details of the other bits
sections on the relevant modules.

Bit 0: Timer A module standby mode control (TACKSTP)

Bit 0 controls setting and clearing of module standby mode for timer A.

TACKSTP	Description
0	Timer A is set to module standby mode
1	Timer A module standby mode is cleared (in

CWOSR is initialized to H'FE by a reset.

Bits 7 to 1: Reserved bits

Bits 7 to 1 are reserved; they are always read as 1 and cannot be modified.

Bit 0: TMOW pin clock select (CWOS)

Bit 0 selects the clock to be output from the TMOW pin.

Bit 0 CWOS	Description	
0	Clock output from timer A is output (see TMA)	(
1	ϕ_W is output	

9.2.3 Timer Operation

1. Interval timer operation

When bit TMA3 in timer mode register A (TMA) is cleared to 0, timer A functions as
interval timer.

Upon reset, TCA is cleared to H'00 and bit TMA3 is cleared to 0, so up-counting and
timing resume immediately. The clock input to timer A is selected by bits TMA2 to T
TMA; any of eight internal clock signals output by prescaler S can be selected.

After the count value in TCA reaches H'FF, the next clock signal input causes timer A
overflow, setting bit IRRTA to 1 in interrupt request register 1 (IRR1). If IENTA = 1
enable register 1 (IENR1), a CPU interrupt is requested.*

At overflow, TCA returns to H'00 and starts counting up again. In this mode timer A
an interval timer that generates an overflow output at intervals of 256 input clock puls

Note: * For details on interrupts, see 3.3, Interrupts.

TMOW. Nine different clock output signals can be selected by means of bits TMA7 to
TMA and bit CWOS in CWOSR. The system clock divided by 32, 16, 8, or 4 can be o
active mode and sleep mode. A 32.768 kHz or 38.4 kHz signal divided by 32, 16, 8, or
output in active mode, sleep mode, watch mode, subactive mode, and subsleep mode.
kHz or 38.4 kHz clock is output in all modes except the reset state.

9.2.4 Timer A Operation States

Table 9.4 summarizes the timer A operation states.

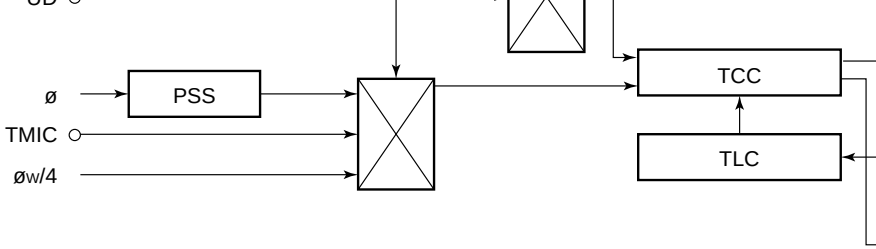
Table 9.4 Timer A Operation States

Operation Mode		Reset	Active	Sleep	Watch	Sub-active	Sub-sleep	Standby
TCA	Interval	Reset	Functions	Functions	Halted	Halted	Halted	Halted
	Clock time base	Reset	Functions	Functions	Functions	Functions	Functions	Halted
TMA		Reset	Functions	Retained	Retained	Functions	Retained	Retained

Note: When the real-time clock time base function is selected as the internal clock of T
active mode or sleep mode, the internal clock is not synchronous with the systen
it is synchronized by a synchronizing circuit. This may result in a maximum error
the count cycle.

Features of timer C are given below.

- Choice of seven internal clock sources (ø/8192, ø/2048, ø/512, ø/64, ø/16, ø/4, øw/
  external clock (can be used to count external events).
- An interrupt is requested when the counter overflows.
- Up/down-counter switching is possible by hardware or software.
- Subactive mode and subsleep mode operation is possible when øw/4 is selected as
  clock, or when an external clock is selected.
- Use of module standby mode enables this module to be placed in standby mode in
  when not used.



Notation:

TMC : Timer mode register C
TCC : Timer counter C
TLC : Timer load register C
IRRTC : Timer C overflow interrupt request flag
PSS : Prescaler S

Figure 9.2 Block Diagram of Timer C

4. Register configuration

Table 9.6 shows the register configuration of timer C.

Table 9.6 Timer C Registers

Name	Abbrev.	R/W	Initial Value
Timer mode register C	TMC	R/W	H'18
Timer counter C	TCC	R	H'00
Timer load register C	TLC	W	H'00
Clock stop register 1	CKSTPR1	R/W	H'FF

9.3.2 Register Descriptions

1. Timer mode register C (TMC)

Bit	7	6	5	4	3	2	1
	TMC7	TMC6	TMC5	—	—	TMC2	TMC1
Initial value	0	0	0	1	1	0	0
Read/Write	R/W	R/W	R/W	—	—	R/W	R/W

TMC is an 8-bit read/write register for selecting the auto-reload function and input clo
performing up/down-counter control.

Upon reset, TMC is initialized to H'18.

Bits 6 and 5: Counter up/down control (TMC6, TMC5)

Selects whether TCC up/down control is performed by hardware using UD pin input, o
TCC functions as an up-counter or a down-counter.

Bit 6 TMC6	Bit 5 TMC5	Description
0	0	TCC is an up-counter (i
0	1	TCC is a down-counter
1	*	Hardware control by UD pin input UD pin input high: Down-counter UD pin input low: Up-counter

*

Bits 4 and 3: Reserved bits

Bits 4 and 3 are reserved; they are always read as 1 and cannot be modified.

0	1	0	Internal clock: ø/512
0	1	1	Internal clock: ø/64
1	0	0	Internal clock: ø/16
1	0	1	Internal clock: ø/4
1	1	0	Internal clock: øw/4
1	1	1	External event (TMIC): rising or falling edge*

Note: * The edge of the external event signal is selected by bit IEG1 in the IRQ edge se (IEGR). See 1. IRQ edge select register (IEGR) in 3.3.2 for details. IRQ2 must in port mode register 1 (PMR1) before setting 111 in bits TMC2 to TMC0.

2. Timer counter C (TCC)

Bit	7	6	5	4	3	2	1
	TCC7	TCC6	TCC5	TCC4	TCC3	TCC2	TCC1
Initial value	0	0	0	0	0	0	0
Read/Write	R	R	R	R	R	R	R

TCC is an 8-bit read-only up-counter, which is incremented by internal clock or extern input. The clock source for input to this counter is selected by bits TMC2 to TMC0 in register C (TMC). TCC values can be read by the CPU at any time.

When TCC overflows from H'FF to H'00 or to the value set in TLC, or underflows fro H'FF or to the value set in TLC, the IRRTC bit in IRR2 is set to 1.

TCC is allocated to the same address as TLC.

Upon reset, TCC is initialized to H'00.

When a reload value is set in TLC, the same value is loaded into timer counter C as wel
starts counting up from that value. When TCC overflows or underflows during operati
reload mode, the TLC value is loaded into TCC. Accordingly, overflow/underflow per
set within the range of 1 to 256 input clocks.

The same address is allocated to TLC as to TCC.

Upon reset, TLC is initialized to H'00.

4. Clock stop register 1 (CKSTPR1)

Bit:	7	6	5	4	3	2	1
	—	S31CKSTP	S32CKSTP	ADCKSTP	TGCKSTP	TFCKSTP	TCCKSTP
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

CKSTPR1 is an 8-bit read/write register that performs module standby mode control fo
modules. Only the bit relating to timer C is described here. For details of the other bits
sections on the relevant modules.

Bit 1: Timer C module standby mode control (TCCKSTP)

Bit 1 controls setting and clearing of module standby mode for timer C.

TCCKSTP	Description
0	Timer C is set to module standby mode
1	Timer C module standby mode is cleared (in

selected from seven internal clock signals output by prescalers S and W, or an externa
at pin TMIC. The selection is made by bits TMC2 to TMC0 in TMC.

TCC up/down-count control can be performed either by software or hardware. The se
made by bits TMC6 and TMC5 in TMC.

After the count value in TCC reaches H'FF (H'00), the next clock input causes timer C
(underflow), setting bit IRRTC to 1 in IRR2. If IENTC = 1 in interrupt enable registe
a CPU interrupt is requested.

At overflow (underflow), TCC returns to H'00 (H'FF) and starts counting up (down) a

During interval timer operation (TMC7 = 0), when a value is set in timer load register
the same value is set in TCC.

Note: For details on interrupts, see 3.3, Interrupts.

depending on the TLC value.

The clock sources, up/down control, and interrupts in auto-reload mode are the same as
mode.

In auto-reload mode (TMC7 = 1), when a new value is set in TLC, the TLC value is als
TCC.

3. Event counter operation

Timer C can operate as an event counter, counting rising or falling edges of an external
signal input at pin TMIC. External event counting is selected by setting bits TMC2 to T
timer mode register C to all 1s (111).

When timer C is used to count external event input, , bit IRQ2 in PMR1 should be set t
IEN2 in IENR1 cleared to 0 to disable interrupt IRQ_2 requests.

4. TCC up/down control by hardware

With timer C, TCC up/down control can be performed by UD pin input. When bit TM
1 in TMC, TCC functions as an up-counter when UD pin input is high, and as a down-c
when low.

When using UD pin input, set bit UD to 1 in PMR3.

						Halted*	Halted*	
	Auto reload	Reset	Functions	Functions	Halted	Functions/ Halted*	Functions/ Halted*	Halted
TMC		Reset	Functions	Retained	Retained	Functions	Retained	Retaine

Note: * When øw/4 is selected as the TCC internal clock in active mode or sleep mode system clock and internal clock are mutually asynchronous, synchronization is by a synchronization circuit. This results in a maximum count cycle error of 1/ø the counter is operated in subactive mode or subsleep mode, either select øw/4 internal clock or select an external clock. The counter will not operate on any o clock. If øw/4 is selected as the internal clock for the counter when øw/8 has b as subclock $ø_{SUB}$, the lower 2 bits of the counter operate on the same cycle, ar operation of the least significant bit is unrelated to the operation of the counter.

1. Features

Features of timer F are given below.

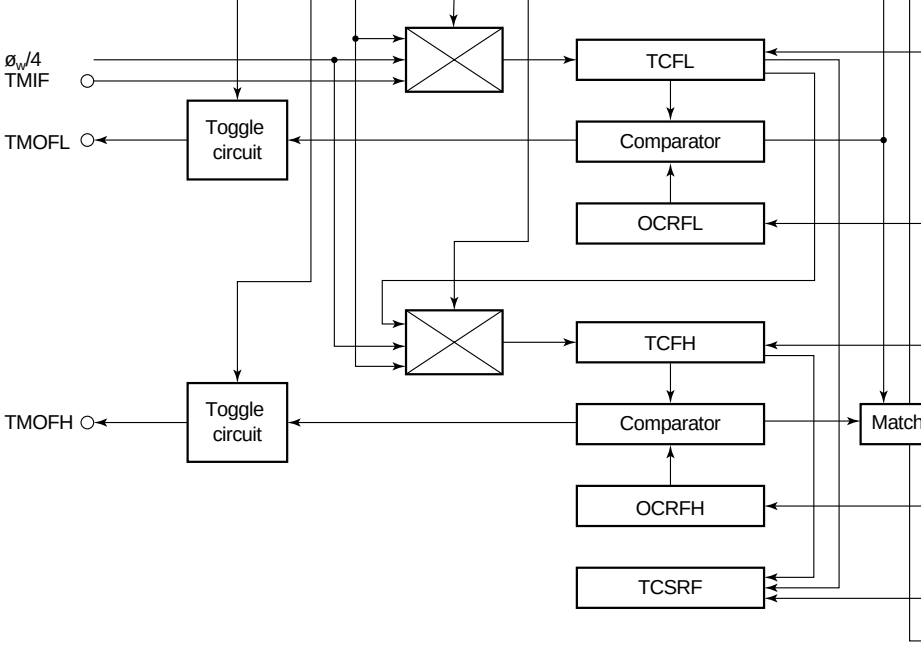
- Choice of four internal clock sources (ø/32, ø/16, ø/4, øw/4) or an external clock (ca
  as an external event counter)
- TMOFH pin toggle output provided using a single compare match signal (toggle ou
  value can be set)
- Counter resetting by a compare match signal
- Two interrupt sources: one compare match, one overflow
- Can operate as two independent 8-bit timers (timer FH and timer FL) (in 8-bit mode

	Timer FH 8-Bit Timer*	**Timer FL 8-Bit Timer/Event Counter**
Internal clock	Choice of 4 (ø/32, ø/16, ø/4, øw/4)	
Event input	—	TMIF pin
Toggle output	One compare match signal, output to TMOFH pin(initial value settable)	One compare match signal, TMOFL pin (initial value set
Counter reset	Counter can be reset by compare match signal	
Interrupt sources	One compare match One overflow	

Note: * When timer F operates as a 16-bit timer, it operates on the timer FL overflow sig

- Operation in watch mode, subactive mode, and subsleep mode

  When øw/4 is selected as the internal clock, timer F can operate in watch mode, sub
  mode, and subsleep mode.
- Use of module standby mode enables this module to be placed in standby mode ind
  when not used.



Notation:

TCRF:	Timer control register F
TCSRF:	Timer control/status register F
TCFH:	8-bit timer counter FH
TCFL:	8-bit timer counter FL
OCRFH:	Output compare register FH
OCRFL:	Output compare register FL
IRRTFH:	Timer FH interrupt request flag
IRRTFL:	Timer FL interrupt request flag
PSS:	Prescaler S

Figure 9.3 Block Diagram of Timer F

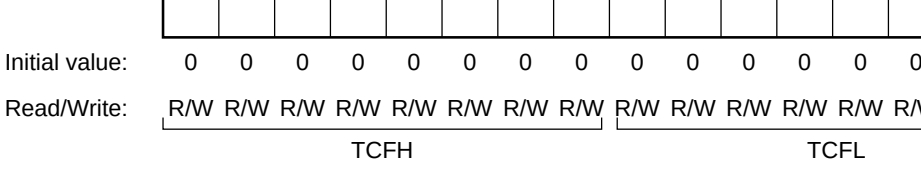
Timer FL output	TMOFL	Output	Timer FL toggle output p

4. Register configuration

Table 9.9 shows the register configuration of timer F.

Table 9.9 Timer F Registers

Name	Abbrev.	R/W	Initial Value
Timer control register F	TCRF	W	H'00
Timer control/status register F	TCSRF	R/W	H'00
8-bit timer counter FH	TCFH	R/W	H'00
8-bit timer counter FL	TCFL	R/W	H'00
Output compare register FH	OCRFH	R/W	H'FF
Output compare register FL	OCRFL	R/W	H'FF
Clock stop register 1	CKSTPR1	R/W	H'FF



TCF is a 16-bit read/write up-counter configured by cascaded connection of 8-bit time
TCFH and TCFL. In addition to the use of TCF as a 16-bit counter with TCFH as the
and TCFL as the lower 8 bits, TCFH and TCFL can also be used as independent 8-bit

TCFH and TCFL can be read and written by the CPU, but when they are used in 16-bi
transfer to and from the CPU is performed via a temporary register (TEMP). For deta
see 9.4.3, CPU Interface.

TCFH and TCFL are each initialized to H'00 upon reset.

a. 16-bit mode (TCF)

When CKSH2 is cleared to 0 in TCRF, TCF operates as a 16-bit counter. The TCI
is selected by bits CKSL2 to CKSL0 in TCRF.

TCF can be cleared in the event of a compare match by means of CCLRH in TCSR

When TCF overflows from H'FFFF to H'0000, OVFH is set to 1 in TCSRF. If OV
TCSRF is 1 at this time, IRRTFH is set to 1 in IRR2, and if IENTFH in IENR2 is
interrupt request is sent to the CPU.

b. 8-bit mode (TCFL/TCFH)

When CKSH2 is set to 1 in TCRF, TCFH and TCFL operate as two independent 8
counters. The TCFH (TCFL) input clock is selected by bits CKSH2 to CKSH0 (C
CKSL0) in TCRF.

TCFH (TCFL) can be cleared in the event of a compare match by means of CCLR
in TCSRF.

When TCFH (TCFL) overflows from H'FF to H'00, OVFH (OVFL) is set to 1 in T
OVIEH (OVIEL) in TCSRF is 1 at this time, IRRTFH (IRRTFL) is set to 1 in IRR
IENTFH (IENTFL) in IENR2 is 1, an interrupt request is sent to the CPU.

OCRF is a 16-bit read/write register composed of the two registers OCRFH and OCRF addition to the use of OCRF as a 16-bit register with OCRFH as the upper 8 bits and O the lower 8 bits, OCRFH and OCRFL can also be used as independent 8-bit registers.

OCRFH and OCRFL can be read and written by the CPU, but when they are used in 16 data transfer to and from the CPU is performed via a temporary register (TEMP). For d TEMP, see 9.4.3, CPU Interface.

OCRFH and OCRFL are each initialized to H'FF upon reset.

a. 16-bit mode (OCRF)

When CKSH2 is cleared to 0 in TCRF, OCRF operates as a 16-bit register. OCRF constantly compared with TCF, and when both values match, CMFH is set to 1 in T the same time, IRRTFH is set to 1 in IRR2. If IENTFH in IENR2 is 1 at this time, a request is sent to the CPU.

Toggle output can be provided from the TMOFH pin by means of compare matches output level can be set (high or low) by means of TOLH in TCRF.

b. 8-bit mode (OCRFH/OCRFL)

When CKSH2 is set to 1 in TCRF, OCRFH and OCRFL operate as two independen registers. OCRFH contents are compared with TCFH, and OCRFL contents are wit When the OCRFH (OCRFL) and TCFH (TCFL) values match, CMFH (CMFL) is s TCSRF. At the same time, IRRTFH (IRRTFL) is set to 1 in IRR2. If IENTFH (IE IENR2 is 1 at this time, an interrupt request is sent to the CPU.

Toggle output can be provided from the TMOFH pin (TMOFL pin) by means of co matches, and the output level can be set (high or low) by means of TOLH (TOLL) i

input clock from among four internal clock sources or external event input, and sets th
level of the TMOFH and TMOFL pins.

TCRF is initialized to H'00 upon reset.

Bit 7: Toggle output level H (TOLH)

Bit 7 sets the TMOFH pin output level. The output level is effective immediately afte
written.

Bit 7 TOLH	Description	
0	Low level	(
1	High level	

Bits 6 to 4: Clock select H (CKSH2 to CKSH0)

Bits 6 to 4 select the clock input to TCFH from among four internal clock sources or T
overflow.

Bit 6 CKSH2	Bit 5 CKSH1	Bit 4 CKSH0	Description	
0	0	0	16-bit mode, counting on TCFL overflow signal	(
0	0	1		
0	1	0		
0	1	1	Use prohibited	
1	0	0	Internal clock: counting on ø/32	
1	0	1	Internal clock: counting on ø/16	
1	1	0	Internal clock: counting on ø/4	
1	1	1	Internal clock: counting on øw/4	

Bits 2 to 0: Clock select L (CKSL2 to CKSL0)

Bits 2 to 0 select the clock input to TCFL from among four internal clock sources or ex
input.

Bit 2 CKSL2	Bit 1 CKSL1	Bit 0 CKSL0	Description
0	0	0	Counting on external event (TMIF) rising/ (ir
0	0	1	falling edge*
0	1	0	
0	1	1	Use prohibited
1	0	0	Internal clock: counting on ø/32
1	0	1	Internal clock: counting on ø/16
1	1	0	Internal clock: counting on ø/4
1	1	1	Internal clock: counting on øw/4

Note: * External event edge selection is set by IEG3 in the IRQ edge select register (IEG
details, see 1. IRQ edge select register (IEGR) in section 3.3.2.

Note that the timer F counter may increment if the setting of IRQ3 in port mode r
(PMR1) is changed from 0 to 1 while the TMIF pin is low in order to change the T
function.

TCSRF is an 8-bit read/write register that performs counter clear selection, overflow f
and compare match flag setting, and controls enabling of overflow interrupt requests.

TCSRF is initialized to H'00 upon reset.

Bit 7: Timer overflow flag H (OVFH)

Bit 7 is a status flag indicating that TCFH has overflowed from H'FF to H'00. This fla
hardware and cleared by software. It cannot be set by software.

Bit 7 OVFH	Description
0	Clearing conditions: After reading OVFH = 1, cleared by writing 0 to OVFH
1	Setting conditions: Set when TCFH overflows from H'FF to H'00

Bit 6: Compare match flag H (CMFH)

Bit 6 is a status flag indicating that TCFH has matched OCRFH. This flag is set by ha
cleared by software. It cannot be set by software.

Bit 6 CMFH	Description
0	Clearing conditions: After reading CMFH = 1, cleared by writing 0 to CMFH
1	Setting conditions: Set when the TCFH value matches the OCRFH value

Bit 4: Counter clear H (CCLRH)

In 8-bit mode, bit 4 selects whether TCF is cleared when TCF and OCRF match.

In 8-bit mode, bit 4 selects whether TCFH is cleared when TCFH and OCRFH match.

Bit 4 CCLRH	Description
0	16-bit mode: TCF clearing by compare match is disabled 8-bit mode: TCFH clearing by compare match is disabled (in
1	16-bit mode: TCF clearing by compare match is enabled 8-bit mode: TCFH clearing by compare match is enabled

Bit 3: Timer overflow flag L (OVFL)

Bit 3 is a status flag indicating that TCFL has overflowed from H'FF to H'00. This flag hardware and cleared by software. It cannot be set by software.

Bit 3 OVFL	Description
0	Clearing conditions: (in After reading OVFL = 1, cleared by writing 0 to OVFL
1	Setting conditions: Set when TCFL overflows from H’FF to H’00

1	Setting conditions: Set when the TCFL value matches the OCRFL value

Bit 1: Timer overflow interrupt enable L (OVIEL)

Bit 1 selects enabling or disabling of interrupt generation when TCFL overflows.

Bit 1 OVIEL	Description
0	TCFL overflow interrupt request is disabled
1	TCFL overflow interrupt request is enabled

Bit 0: Counter clear L (CCLRL)

Bit 0 selects whether TCFL is cleared when TCFL and OCRFL match.

Bit 0 CCLRL	Description
0	TCFL clearing by compare match is disabled
1	TCFL clearing by compare match is enabled

modules. Only the bit relating to timer F is described here. For details of the other bits
sections on the relevant modules.

Bit 2: Timer F module standby mode control (TFCKSTP)

Bit 2 controls setting and clearing of module standby mode for timer F.

TFCKSTP	Description
0	Timer F is set to module standby mode
1	Timer F module standby mode is cleared (in

is accessed.

In 8-bit mode, there are no restrictions on the order of access.

1. Write access

Write access to the upper byte results in transfer of the upper-byte write data to TEMP
write access to the lower byte results in transfer of the data in TEMP to the upper regis
and direct transfer of the lower-byte write data to the lower register byte.

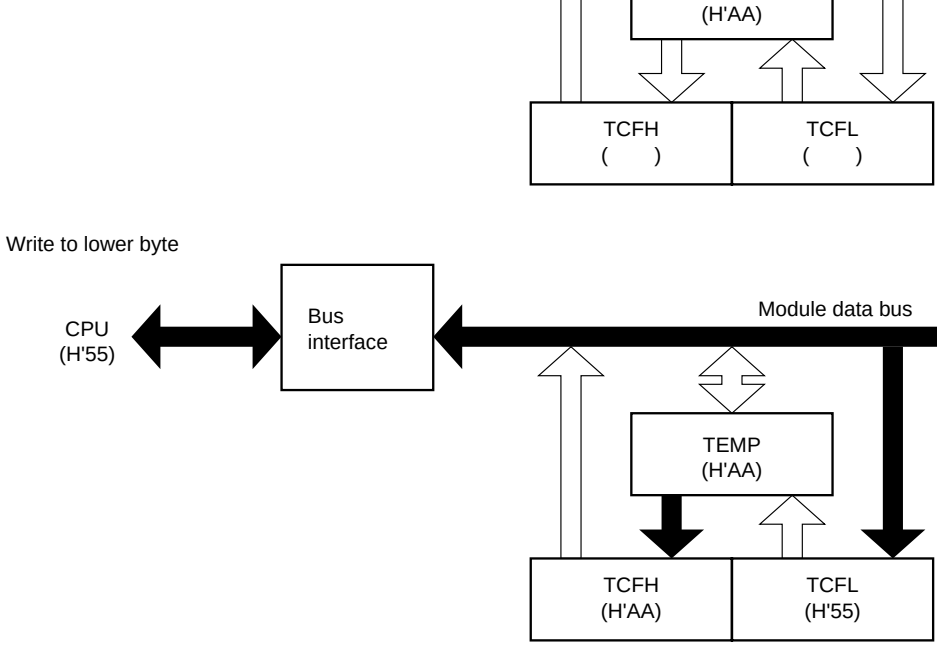


Figure 9.4 Write Access to TCR (CPU Æ TCF)

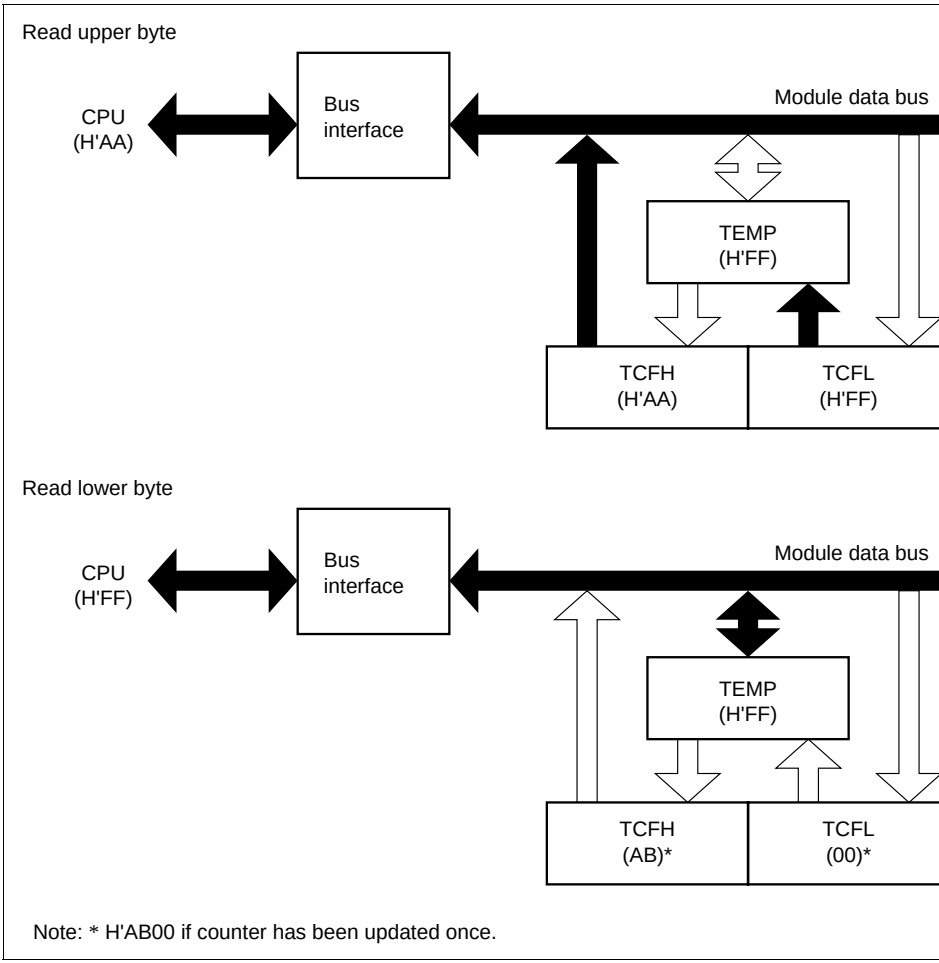


Figure 9.5 Read Access to TCF (TCF Æ CPU)

Timer F has two operating modes, 16-bit timer mode and 8-bit timer mode. The operat
of these modes is described below.

a. Operation in 16-bit timer mode

When CKSH2 is cleared to 0 in timer control register F (TCRF), timer F operates as
timer.

Following a reset, timer counter F (TCF) is initialized to H'0000, output compare re
(OCRF) to H'FFFF, and timer control register F (TCRF) and timer control/status reg
(TCSRF) to H'00. The counter starts incrementing on external event (TMIF) input.
external event edge selection is set by IEG3 in the IRQ edge select register (IEGR).

The timer F operating clock can be selected from four internal clocks output by pres
an external clock by means of bits CKSL2 to CKSL0 in TCRF.

OCRF contents are constantly compared with TCF, and when both values match, C
to 1 in TCSRF. If IENTFH in IENR2 is 1 at this time, an interrupt request is sent to
and at the same time, TMOFH pin output is toggled. If CCLRH in TCSRF is 1, TC
cleared. TMOFH pin output can also be set by TOLH in TCRF.

When TCF overflows from H'FFFF to H'0000, OVFH is set to 1 in TCSRF. If OVI
TCSRF and IENTFH in IENR2 are both 1, an interrupt request is sent to the CPU.

b. Operation in 8-bit timer mode

When CKSH2 is set to 1 in TCRF, TCF operates as two independent 8-bit timers, T
TCFL. The TCFH/TCFL input clock is selected by CKSH2 to CKSH0/CKSL2 to C
TCRF.

When the OCRFH/OCRFL and TCFH/TCFL values match, CMFH/CMFL is set to
TCSRF. If IENTFH/IENTFL in IENR2 is 1, an interrupt request is sent to the CPU
same time, TMOFH pin/TMOFL pin output is toggled. If CCLRH/CCLRL in TCS
TCFH/TCFL is cleared. TMOFH pin/TMOFL pin output can also be set by TOLH/
TCRF.

When TCFH/TCFL overflows from H'FF to H'00, OVFH/OVFL is set to 1 in TCSR
OVIEH/OVIEL in TCSRF and IENTFH/IENTFL in IENR2 are both 1, an interrupt
sent to the CPU.

either the rising or falling edge of external event input. External event edge selecti
IEG3 in the interrupt controller's IEGR register. An external event pulse width of
system clocks (ø) is necessary. Shorter pulses will not be counted correctly.

3. TMOFH/TMOFL output timing

In TMOFH/TMOFL output, the value set in TOLH/TOLL in TCRF is output. The ou
toggled by the occurrence of a compare match. Figure 9.6 shows the output timing.

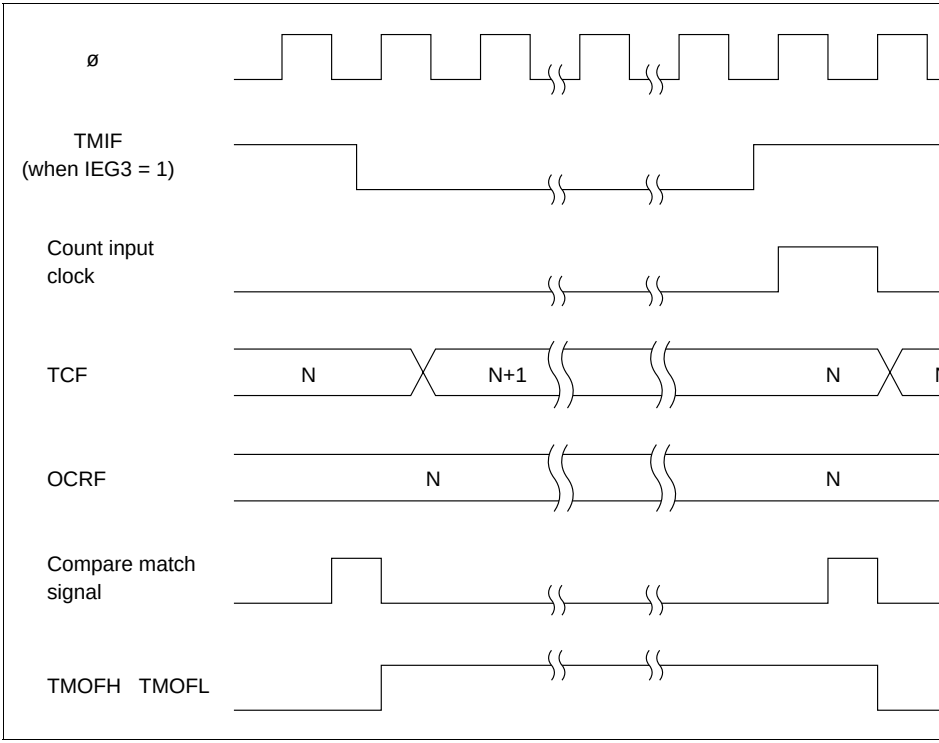


Figure 9.6 TMOFH/TMOFL Output Timing

The compare match flag (CMFH or CMFL) is set to 1 when the TCF and OCRF values
The compare match signal is generated in the last state during which the values match (
is updated from the matching value to a new value). When TCF matches OCRF, the co
match signal is not generated until the next counter clock.

7. Timer F operation modes

Timer F operation modes are shown in table 9.10.

Table 9.10 Timer F Operation Modes

Operation Mode	Reset	Active	Sleep	Watch	Sub-active	Sub-sleep	Standby
TCF	Reset	Functions	Functions	Functions/ Halted*	Functions/ Halted*	Functions/ Halted*	Halted
OCRF	Reset	Functions	Held	Held	Functions	Held	Held
TCRF	Reset	Functions	Held	Held	Functions	Held	Held
TCSRF	Reset	Functions	Held	Held	Functions	Held	Held

Note: * When $ø_W/4$ is selected as the TCF internal clock in active mode or sleep mode, s
system clock and internal clock are mutually asynchronous, synchronization is m
by a synchronization circuit. This results in a maximum count cycle error of 1/ø (
the counter is operated in subactive mode, watch mode, or subsleep mode, $ø_W/4$
selected as the internal clock. The counter will not operate if any other internal c
selected.

write. TMOFL pin output is unstable in 16-bit mode, and should not be used; the TM
should be used as a port pin.

If an OCRFL write and compare match signal generation occur simultaneously, the co
match signal is invalid. However, if the written data and the counter value match, a co
match signal will be generated at that point. As the compare match signal is output in
synchronization with the TCFL clock, a compare match will not result in compare ma
generation if the clock is stopped.

Compare match flag CMFH is set when all 16 bits match and a compare match signal
Compare match flag CMFL is set if the setting conditions for the lower 8 bits are satis

When TCF overflows, OVFH is set. OVFL is set if the setting conditions are satisfied
lower 8 bits overflow. If a TCFL write and overflow signal output occur simultaneous
overflow signal is not output.

2. 8-bit timer mode

a. TCFH, OCRFH

In toggle output, TMOFH pin output is toggled when a compare match occurs. If
write by a MOV instruction and generation of the compare match signal occur sim
TOLH data is output to the TMOFH pin as a result of the TCRF write.

If an OCRFH write and compare match signal generation occur simultaneously, th
match signal is invalid. However, if the written data and the counter value match,
match signal will be generated at that point. The compare match signal is output i
synchronization with the TCFH clock.

If a TCFH write and overflow signal output occur simultaneously, the overflow sig
output.

b. TCFL, OCRFL

In toggle output, TMOFL pin output is toggled when a compare match occurs. If a
write by a MOV instruction and generation of the compare match signal occur sim
TOLL data is output to the TMOFL pin as a result of the TCRF write.

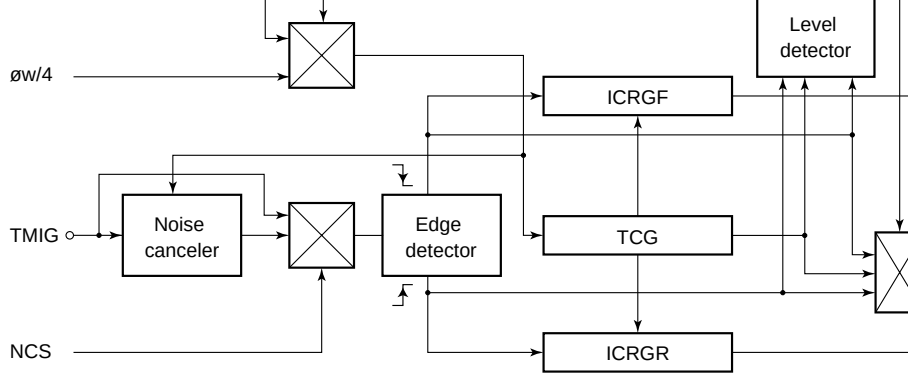
timer G functions as an 8-bit interval timer.

1. Features

Features of timer G are given below.

- Choice of four internal clock sources (ø/64, ø/32, ø/2, øw/2)
- Dedicated input capture functions for rising and falling edges
- Level detection at counter overflow

  It is possible to detect whether overflow occurred when the input capture input sig
  or when it was low.
- Selection of whether or not the counter value is to be cleared at the input capture i
  rising edge, falling edge, or both edges
- Two interrupt sources: one input capture, one overflow. The input capture input si
  or falling edge can be selected as the interrupt source.
- A built-in noise canceler eliminates high-frequency component noise in the input c
  signal.
- Watch mode, subactive mode and subsleep mode operation is possible when øw/2
  as the internal clock.
- Use of module standby mode enables this module to be placed in standby mode in
  when not used.



Notation:

TMG	:	Timer mode register G
TCG	:	Timer counter G
ICRGF	:	Input capture register GF
ICRGR	:	Input capture register GR
IRRTG	:	Timer G interrupt request flag
NCS	:	Noise canceler select
PSS	:	Prescaler S

Figure 9.7 Block Diagram of Timer G

4. Register configuration

Table 9.12 shows the register configuration of timer G.

Table 9.12 Timer G Registers

Name	Abbrev.	R/W	Initial Value
Timer control register G	TMG	R/W	H'00
Timer counter G	TCG	—	H'00
Input capture register GF	ICRGF	R	H'00
Input capture register GR	ICRGR	R	H'00
Clock stop register 1	CKSTPR1	R/W	H'FF

TCG is an 8-bit up-counter which is incremented by clock input. The input clock is sel
bits CKS1 and CKS0 in TMG.

TMIG in PMR1 is set to 1 to operate TCG as an input capture timer, or cleared to 0 to c
TCG as an interval timer*. In input capture timer operation, the TCG value can be clea
rising edge, falling edge, or both edges of the input capture input signal, according to th
made in TMG.

When TCG overflows from H'FF to H'00, if OVIE in TMG is 1, IRRTG is set to 1 in IR
IENTG in IENR2 is 1, an interrupt request is sent to the CPU.

For details of the interrupt, see 3.3, Interrupts.

TCG cannot be read or written by the CPU. It is initialized to H'00 upon reset.

Note: * An input capture signal may be generated when TMIG is modified.

2. Input capture register GF (ICRGF)

Bit:	7	6	5	4	3	2	1
	ICRGF7	ICRGF6	ICRGF5	ICRGF4	ICRGF3	ICRGF2	ICRGF1
Initial value:	0	0	0	0	0	0	0
Read/Write:	R	R	R	R	R	R	R

ICRGF is an 8-bit read-only register. When a falling edge of the input capture input sig
detected, the current TCG value is transferred to ICRGF. If IIEGS in TMG is 1 at this
IRRTG is set to 1 in IRR2, and if IENTG in IENR2 is 1, an interrupt request is sent to t

For details of the interrupt, see 3.3, Interrupts.

To ensure dependable input capture operation, the pulse width of the input capture inpu
must be at least 2ø or $2ø_{SUB}$ (when the noise canceler is not used).

ICRGF is initialized to H'00 upon reset.

detected, the current TCG value is transferred to ICRGR. If IIEGS in TMG is 1 at this
IRRTG is set to 1 in IRR2, and if IENTG in IENR2 is 1, an interrupt request is sent to

For details of the interrupt, see 3.3, Interrupts.

To ensure dependable input capture operation, the pulse width of the input capture inp
must be at least 2ø or $2ø_{SUB}$ (when the noise canceler is not used).

ICRGR is initialized to H'00 upon reset.

4. Timer mode register G (TMG)

Bit:	7	6	5	4	3	2	1
	OVFH	OVFL	OVIE	IIEGS	CCLR1	CCLR0	CKS1
Initial value:	0	0	0	0	0	0	0
Read/Write:	R/W*	R/W*	R/W	R/W	R/W	R/W	R/W

Note: * Bits 7 and 6 can only be written with 0, for flag clearing.

TMG is an 8-bit read/write register that performs TCG clock selection from four inter
sources, counter clear selection, and edge selection for the input capture input signal i
request, controls enabling of overflow interrupt requests, and also contains the overflo

TMG is initialized to H'00 upon reset.

	After reading OVFH = 1, cleared by writing 0 to OVFH
1	Setting conditions: Set when TCG overflows from H'FF to H'00

Bit 6: Timer overflow flag L (OVFL)

Bit 6 is a status flag indicating that TCG has overflowed from H'FF to H'00 when the in
input signal is low, or in interval operation. This flag is set by hardware and cleared by
It cannot be set by software.

Bit 6 **OVFL**	**Description**
0	Clearing conditions: After reading OVFL = 1, cleared by writing 0 to OVFL
1	Setting conditions: Set when TCG overflows from H'FF to H'00

Bit 5: Timer overflow interrupt enable (OVIE)

Bit 5 selects enabling or disabling of interrupt generation when TCG overflows.

Bit 5 **OVIE**	**Description**
0	TCG overflow interrupt request is disabled
1	TCG overflow interrupt request is enabled

Bits 3 and 2: Counter clear 1 and 0 (CCLR1, CCLR0)

Bits 3 and 2 specify whether or not TCG is cleared by the rising edge, falling edge, or of the input capture input signal.

Bit 3 CCLR1	Bit 2 CCLR0	Description
0	0	TCG clearing is disabled
0	1	TCG cleared by falling edge of input capture input signal
1	0	TCG cleared by rising edge of input capture input signal
1	1	TCG cleared by both edges of input capture input signal

Bits 1 and 0: Clock select (CKS1, CKS0)

Bits 1 and 0 select the clock input to TCG from among four internal clock sources.

Bit 1CKS1	Bit 0CKS0	Description
0	0	Internal clock: counting on ø/64
0	1	Internal clock: counting on ø/32
1	0	Internal clock: counting on ø/2
1	1	Internal clock: counting on øw/4

modules. Only the bit relating to timer G is described here. For details of the other bits
sections on the relevant modules.

Bit 3: Timer G module standby mode control (TGCKSTP)

Bit 3 controls setting and clearing of module standby mode for timer G.

TGCKSTP	Description
0	Timer G is set to module standby mode
1	Timer G module standby mode is cleared (in

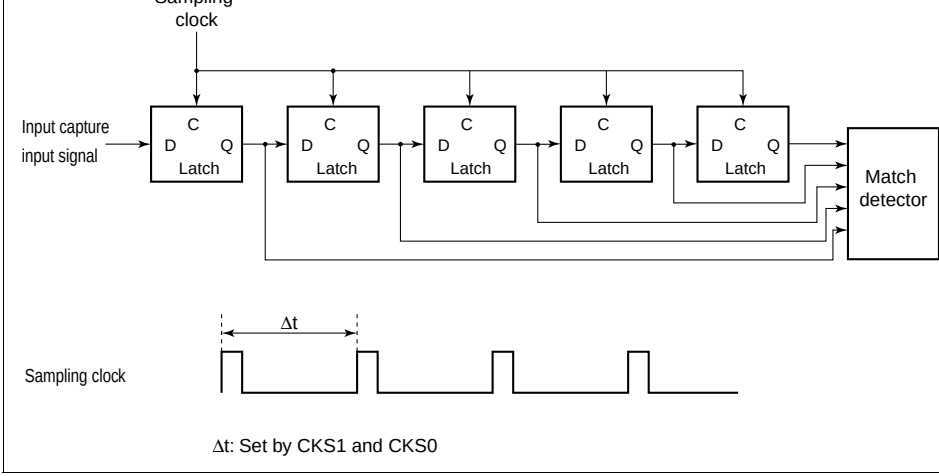


Figure 9.8 Noise Canceler Block Diagram

The noise canceler consists of five latch circuits connected in series and a match detec
When the noise cancellation function is not used (NCS = 0), the system clock is select
sampling clock When the noise cancellation function is used (NCS = 1), the sampling
internal clock selected by CKS1 and CKS0 in TMG, the input capture input is sample
rising edge of this clock, and the data is judged to be correct when all the latch outputs
all the outputs do not match, the previous value is retained. After a reset, the noise ca
is initialized when the falling edge of the input capture input signal has been sampled
Therefore, after making a setting for use of the noise cancellation function, a pulse wit
five times the width of the sampling clock is a dependable input capture signal. Even
cancellation is not used, an input capture input signal pulse width of at least 2ø or 2ø$_S$
necessary to ensure that input capture operations are performed properly

Note: * An input capture signal may be generated when the NCS bit is modified.

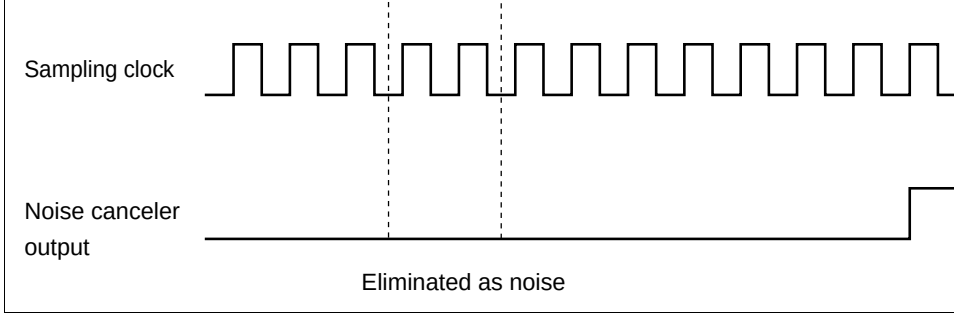


Figure 9.9 Noise Canceler Timing (Example)

The operation of these two functions is described below.

a. Input capture timer operation

When the TMIG bit is set to 1 in port mode register 1 (PMR1), timer G functions a capture timer*.

In a reset, timer mode register G (TMG), timer counter G (TCG), input capture reg (ICRGF), and input capture register GR (ICRGR) are all initialized to H'00.

Following a reset, TCG starts incrementing on the ø/64 internal clock.

The input clock can be selected from four internal clock sources by bits CKS1 and TMG.

When a rising edge/falling edge is detected in the input capture signal input from t pin, the TCG value at that time is transferred to ICRGR/ICRGF. When the edge s IIEGS in TMG is input, IRRTG is set to 1 in IRR2, and if the IENTG bit in IENR2 time, an interrupt request is sent to the CPU. For details of the interrupt, see 3.3.,

TCG can be cleared by a rising edge, falling edge, or both edges of the input captu according to the setting of bits CCLR1 and CCLR0 in TMG. If TCG overflows w capture signal is high, the OVFH bit is set in TMG; if TCG overflows when the inp signal is low, the OVFL bit is set in TMG. If the OVIE bit in TMG is 1 when thes set, IRRTG is set to 1 in IRR2, and if the IENTG bit in IENR2 is 1, timer G sends request to the CPU. For details of the interrupt, see 3.3., Interrupts.

Timer G has a built-in noise canceler that enables high-frequency component noise eliminated from pulses input from the TMIG pin. For details, see 9.5.3, Noise Can

Note: * An input capture signal may be generated when TMIG is modified.

b. Interval timer operation

When the TMIG bit is cleared to 0 in PMR1, timer G functions as an interval timer a reset, TCG starts incrementing on the ø/64 internal clock. The input clock can be from four internal clock sources by bits CKS1 and CKS0 in TMG. TCG incremen selected clock, and when it overflows from H'FF to H'00, the OVFL bit is set to 1 the OVIE bit in TMG is 1 at this time, IRRTG is set to 1 in IRR2, and if the IENT IENR2 is 1, timer G sends an interrupt request to the CPU. For details of the inter 3.3., Interrupts.

For input capture input, dedicated input capture functions are provided for rising and
edges.

Figure 9.10 shows the timing for rising/falling edge input capture input.

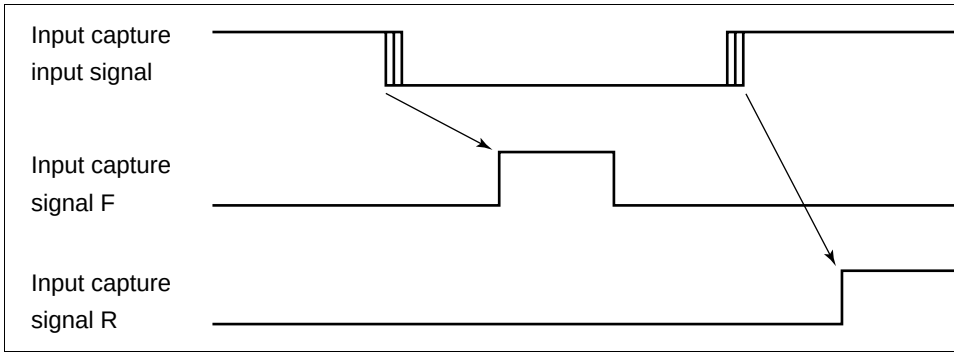


Figure 9.10 Input Capture Input Timing (without Noise Cancellation Func

b. With noise cancellation function

When noise cancellation is performed on the input capture input, the passage of the
capture signal through the noise canceler results in a delay of five sampling clock cy
the input capture input signal edge.

Figure 9.11 shows the timing in this case.

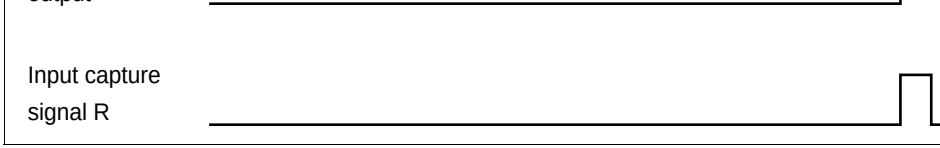


Figure 9.11 Input Capture Input Timing (with Noise Cancellation Funct

4. Timing of input capture by input capture input

Figure 9.12 shows the timing of input capture by input capture input

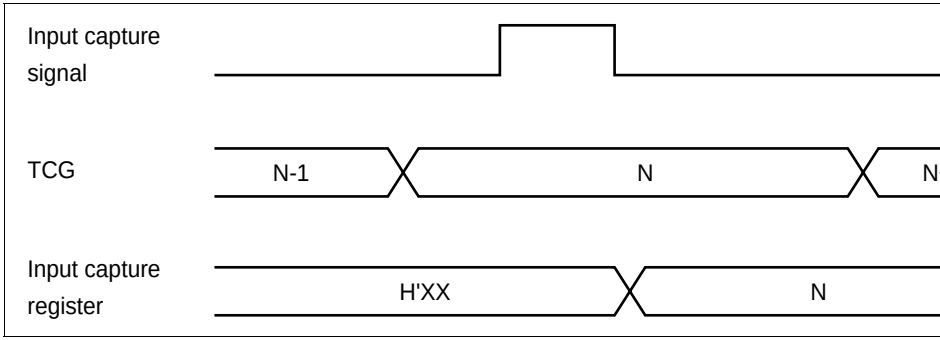


Figure 9.12 Timing of Input Capture by Input Capture Input

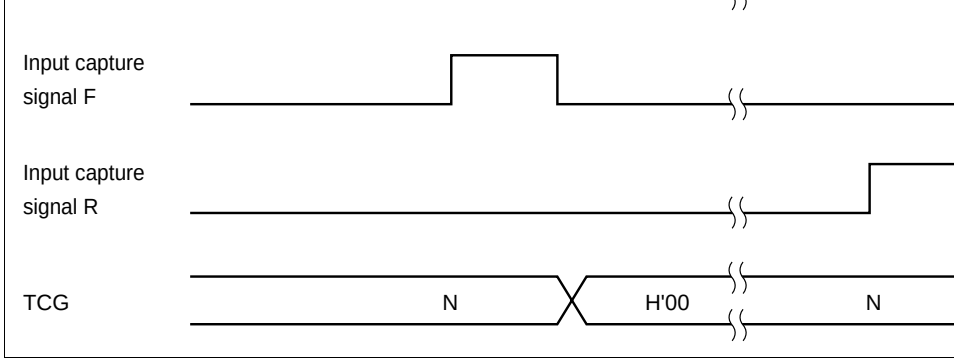


Figure 9.13 TCG Clear Timing

Interval	Reset	Functions*	Functions*	Functions/ halted*	Functions/ halted*	Functions/ halted*	Halted
ICRGF	Reset	Functions*	Functions*	Functions/ halted*	Functions/ halted*	Functions/ halted*	Held
ICRGR	Reset	Functions*	Functions*	Functions halted*	Functions/ halted*	Functions/ halted*	Held
TMG	Reset	Functions	Held	Held	Functions	Held	Held

Note: * When øw/4 is selected as the TCG internal clock in active mode or sleep mode system clock and internal clock are mutually asynchronous, synchronization is by a synchronization circuit. This results in a maximum count cycle error of 1/ø øw/4 is selected as the TCG internal clock in watch mode, TCG and the noise c operate on the øw/4 internal clock without regard to the $ø_{SUB}$ subclock (øw/8, ø Note that when another internal clock is selected, TCG and the noise canceler operate, and input of the input capture input signal does not result in input captu

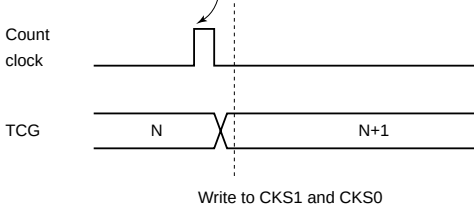
To be operated Timer G in subactive mode or subsleep mode, select øw/4 for i of TCG and also select øw/2 for sub clock $ø_{SUB}$. When another internal clock is and when another sub clock (øw/8, øw/4) is selected, TCG and noise canceler operate.

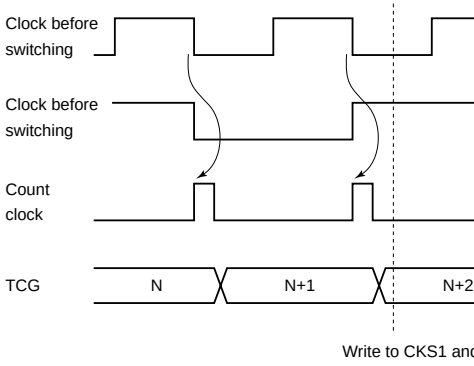
9.5.5 Application Notes

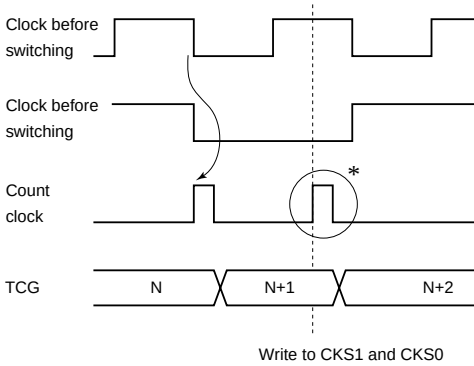
1. Internal clock switching and TCG operation

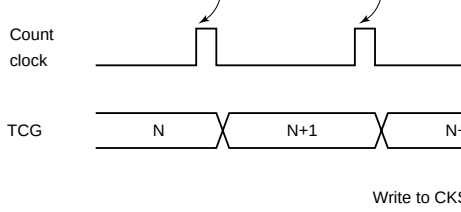
Depending on the timing, TCG may be incremented by a switch between difference in sources. Table 9.14 shows the relation between internal clock switchover timing (by v CKS1 and CKS0) and TCG operation.

When TCG is internally clocked, an increment pulse is generated on detection of the f of an internal clock signal, which is divided from the system clock (ø) or subclock (øw reason, in a case like No. 3 in table 9.14 where the switch is from a high clock signal t clock signal, the switchover is seen as a falling edge, causing TCG to increment.



2	Goes from low level to high level	

3	Goes from high level to low level	



Note: * The switchover is seen as a falling edge, and TCG is incremented.

2. Notes on port mode register modification

The following points should be noted when a port mode register is modified to switch capture function or the input capture input noise canceler function.

- Switching input capture input pin function

Note that when the pin function is switched by modifying TMIG in port mode register which performs input capture input pin control, an edge will be regarded as having be the pin even though no valid edge has actually been input. Input capture input signal i and the conditions for their occurrence, are summarized in table 9.15.

Generation of falling edge	When TMIG is modified from 1 to 0 while the TMIG pi
	When NCS is modified from 0 to 1 while the TMIG pin then TMIG is modified from 0 to 1 before the signal is five times by the noise canceler
	When NCS is modified from 0 to 1 while the TMIG pin then TMIG is modified from 1 to 0 after the signal is s times by the noise canceler

Note: When the $P1_3$ pin is not set as an input capture input pin, the timer G input captu signal is low.

- Switching input capture input noise canceler function

When performing noise canceler function switching by modifying NCS in port mode re (PMR3), which controls the input capture input noise canceler, TMIG should first be cl Note that if NCS is modified without first clearing TMIG, an edge will be regarded as h input at the pin even though no valid edge has actually been input. Input capture input edges, and the conditions for their occurrence, are summarized in table 9.16.

Table 9.16 Input Capture Input Signal Input Edges Due to Noise Canceler Funct Switching, and Conditions for Their Occurrence

Input Capture Input Signal Input Edge	Conditions
Generation of rising edge	When the TMIG pin level is switched from low to high TMIG is set to 1, then NCS is modified from 0 to 1 bef signal is sampled five times by the noise canceler
Generation of falling edge	When the TMIG pin level is switched from high to low TMIG is set to 1, then NCS is modified from 1 to 0 bef signal is sampled five times by the noise canceler

interrupt request flag setting when the pin function is switched: by controlling the pin
the conditions shown in tables 9.15 and 9.16 are not satisfied, or by setting the opposit
generated edge in the IIEGS bit in TMG.

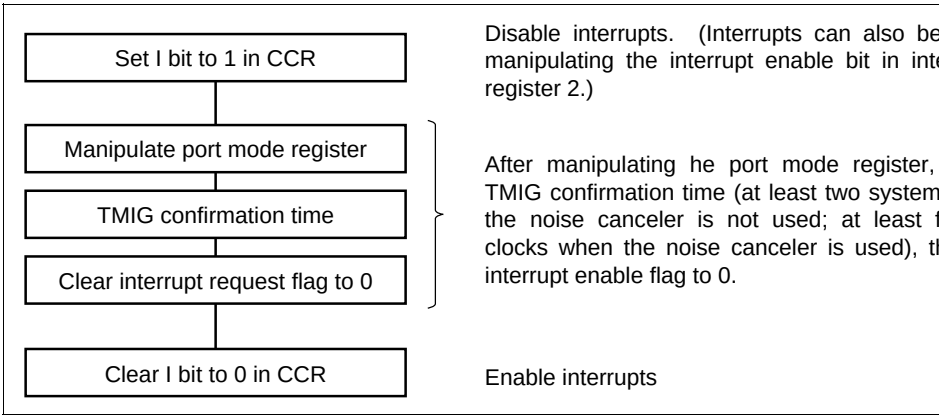


**Figure 9.14 Port Mode Register Manipulation and Interrupt Enable Flag C
Procedure**

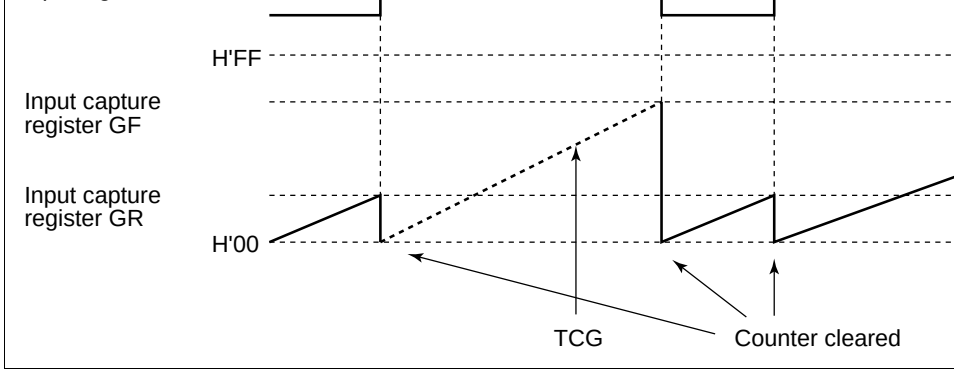


Figure 9.15 Timer G Application Example

1. Features

Features of the watchdog timer are given below.

- Incremented by internal clock source (ø/8192 or øw/32).
- A reset signal is generated when the counter overflows. The overflow period can b
  from 1 to 256 times 8192/ø or 32/øw (from approximately 4 ms to 1000 ms when ø
  MHz).
- Use of module standby mode enables this module to be placed in standby mode ind
  when not used.

2. Block diagram

Figure 9.16 shows a block diagram of the watchdog timer.

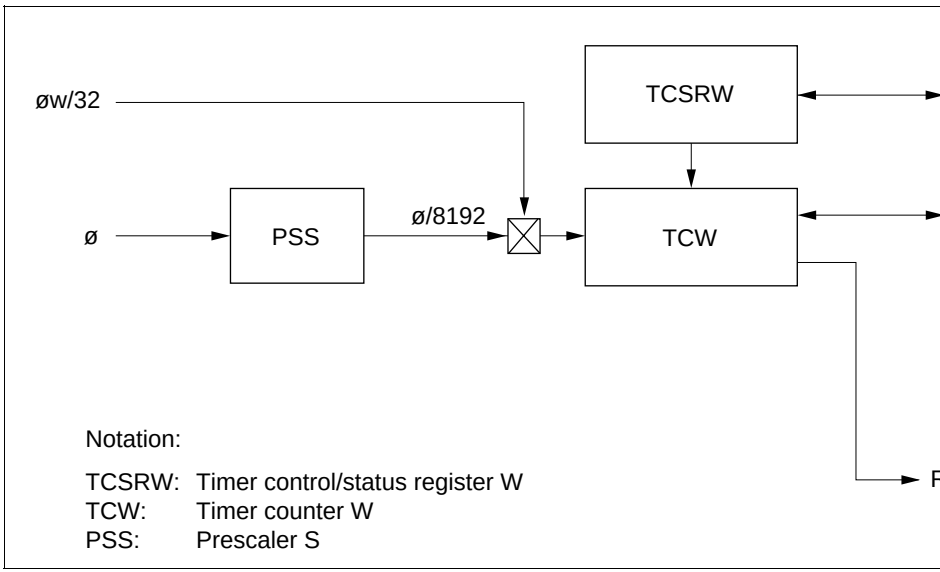


Figure 9.16 Block Diagram of Watchdog Timer

Clock stop register 2	CKSTP2	R/W	H'FF
Port mode register 3	PMR3	R/W	H'00

9.6.2 Register Descriptions

1. Timer control/status register W (TCSRW)

Bit	7	6	5	4	3	2	1
	B6WI	TCWE	B4WI	TCSRWE	B2WI	WDON	B0WI
Initial value	1	0	1	0	1	0	1
Read/Write	R	R/W*	R	R/W*	R	R/W*	R

Note: * Write is permitted only under certain conditions, which are given in the descr the individual bits.

TCSRW is an 8-bit read/write register that controls write access to TCW and TCSRW i controls watchdog timer operations, and indicates operating status.

Bit 7: Bit 6 write inhibit (B6WI)

Bit 7 controls the writing of data to bit 6 in TCSRW.

Bit 7 B6WI	Description	
0	Bit 6 is write-enabled	
1	Bit 6 is write-protected	(ir

This bit is always read as 1. Data written to this bit is not stored.

Bit 5: Bit 4 write inhibit (B4WI)

Bit 5 controls the writing of data to bit 4 in TCSRW.

Bit 5 B4WI	Description
0	Bit 4 is write-enabled
1	Bit 4 is write-protected

This bit is always read as 1. Data written to this bit is not stored.

Bit 4: Timer control/status register W write enable (TCSRWE)

Bit 4 controls the writing of data to TCSRW bits 2 and 0.

Bit 4 TCSRWE	Description
0	Data cannot be written to bits 2 and 0
1	Data can be written to bits 2 and 0

Bit 3: Bit 2 write inhibit (B2WI)

Bit 3 controls the writing of data to bit 2 in TCSRW.

Bit 3 B2WI	Description
0	Bit 2 is write-enabled
1	Bit 2 is write-protected

This bit is always read as 1. Data written to this bit is not stored.

1	Watchdog timer operation is enabled Setting conditions: When TCSRWE = 1 and 0 is written in B2WI and 1 is written in WDON

Counting starts when this bit is set to 1, and stops when this bit is cleared to 0.

Bit 1: Bit 0 write inhibit (B0WI)

Bit 1 controls the writing of data to bit 0 in TCSRW.

Bit 1 B0WI	Description
0	Bit 0 is write-enabled
1	Bit 0 is write-protected (i

This bit is always read as 1. Data written to this bit is not stored.

Bit 0: Watchdog timer reset (WRST)

Bit 0 indicates that TCW has overflowed, generating an internal reset signal. The intern signal generated by the overflow resets the entire chip. WRST is cleared to 0 by a reset $\overline{\text{RES}}$ pin, or when software writes 0.

Bit 0 WRST	Description
0	Clearing conditions: Reset by $\overline{\text{RES}}$ pin When TCSRWE = 1, and 0 is written in both B0WI and WRST
1	Setting conditions: When TCW overflows and an internal reset signal is generated

clock is ø/8192 or øw/32. The TCW value can always be written or read by the CPU.

When TCW overflows from H'FF to H'00, an internal reset signal is generated and WI
1 in TCSRW. Upon reset, TCW is initialized to H'00.

3. Clock stop register 2 (CKSTPR2)

Bit	7	6	5	4	3	2	1
	—	—	—	—	AECKSTP	WDCKSTP	PWCKST
Initial value	1	1	1	1	1	1	1
Read/Write	—	—	—	—	R/W	R/W	R/W

CKSTPR2 is an 8-bit read/write register that performs module standby mode control f
modules. Only the bit relating to the watchdog timer is described here. For details of
bits, see the sections on the relevant modules.

Bit 2: Watchdog timer module standby mode control (WDCKSTP)

Bit 2 controls setting and clearing of module standby mode for the watchdog timer.

WDCKSTP	Description
0	Watchdog timer is set to module standby mode
1	Watchdog timer module standby mode is cleared

Note: WDCKSTP is valid when the WDON bit is cleared to 0 in timer control/status re
(TCSRW). If WDCKSTP is set to 0 while WDON is set to 1 (during watchdog ti
operation), 0 will be set in WDCKSTP but the watchdog timer will continue its w
function and will not enter module standby mode. When the watchdog function
WDON is cleared to 0 by software, the WDCKSTP setting will become valid and
watchdog timer will enter module standby mode.

0	ø/8192 selected	(in
1	øw/32 selected	

9.6.3 Timer Operation

The watchdog timer has an 8-bit counter (TCW) that is incremented by clock input (ø/8 øw/32). The input clock is selected by bit WDCKS in port mode register 3 (PMR3): ø/8 selected when WDCKS is cleared to 0, and øw/32 when set to 1. When TCSRWE = 1 i if 0 is written in B2WI and 1 is simultaneously written in WDON, TCW starts counting the TCW count reaches H'FF, the next clock input causes the watchdog timer to overflo internal reset signal is generated one reference clock (ø or $ø_{SUB}$) cycle later. The intern signal is output for 512 clock cycles of the $ø_{OSC}$ clock. It is possible to write to TCW, c TCW to count up from the written value. The overflow period can be set in the range fr 256 input clocks, depending on the value written in TCW.

Figure 9.17 shows an example of watchdog timer operations.

Example: ø = 2 MHz and the desired overflow period is 30 ms.

$$\frac{2 \times 10^6}{8192} \times 30 \times 10^{-3} = 7.3$$

The value set in TCW should therefore be 256 – 8 = 248 (H'F8).

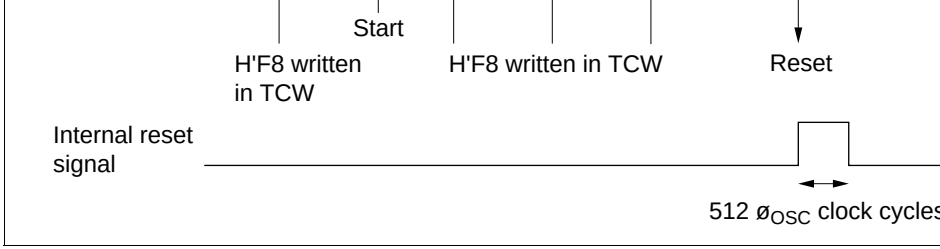


Figure 9.17 Typical Watchdog Timer Operations (Example)

9.6.4 Watchdog Timer Operation States

Table 9.18 summarizes the watchdog timer operation states.

Table 9.18 Watchdog Timer Operation States

Operation Mode	Reset	Active	Sleep	Watch	Sub-active	Sub-sleep	Standb
TCW	Reset	Functions	Functions	Halted	Functions/ Halted*	Halted	Halted
TCSRW	Reset	Functions	Functions	Retained	Functions/ Halted*	Retained	Retaine

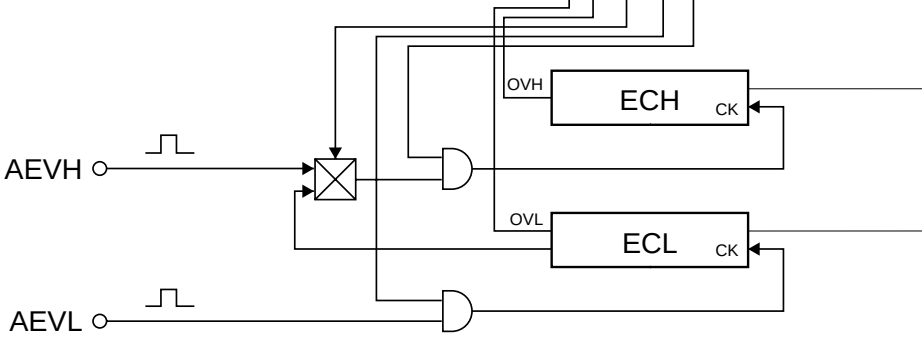
Note: * Functions when øw/32 is selected as the input clock.

- Can count asynchronous events

Can count external events input asynchronously without regard to the operation of base and $ø_{SUB}$.

The counter has a 16-bit configuration, enabling it to count up to 65536 (2^{16}) events.

- Can also be used as two independent 8-bit event counter channels.
- Counter resetting and halting of the count-up function controllable by software
- Automatic interrupt generation on detection of event counter overflow
- Use of module standby mode enables this module to be placed in standby mode inde when not used.



Notation:

ECCSR : Event counter control/status register
ECH : Event counter H
ECL : Event counter L
AEVH : Asynchronous event input H
AEVL : Asynchronous event input L
IRREC : Event counter overflow interrupt request flag

Figure 9.18 Block Diagram of Asynchronous Event Counter

4. Register configuration

Table 9.20 shows the register configuration of the asynchronous event counter.

Table 9.20 Asynchronous Event Counter Registers

Name	Abbrev.	R/W	Initial Value
Event counter control/status register	ECCSR	R/W	H'00
Event counter H	ECH	R	H'00
Event counter L	ECL	R	H'00
Clock stop register 2	CKSTP2	R/W	H'FF

9.7.2 Register Descriptions

1. Event counter control/status register (ECCSR)

Bit	7	6	5	4	3	2	1
	OVH	OVL	—	CH2	CUEH	CUEL	CRCH
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 7 and 6 can only be written with 0, for flag clearing.

ECCSR is an 8-bit read/write register that controls counter overflow detection, counter and halting of the count-up function.

ECCSR is initialized to H'00 upon reset.

OVH	Description
0	ECH has not overflowed Clearing conditions: After reading OVH = 1, cleared by writing 0 to OVH
1	ECH has overflowed Setting conditions: Set when ECH overflows from H'FF to H'00

Bit 6: Counter overflow flag L (OVL)

Bit 6 is a status flag indicating that ECL has overflowed from H'FF to H'00. This flag ECL overflows. It is cleared by software but cannot be set by software. OVL is clear reading it when set to 1, then writing 0.

Bit 6 OVL	Description
0	ECL has not overflowed Clearing conditions: After reading OVL = 1, cleared by writing 0 to OVL
1	ECL has overflowed Setting conditions: Set when ECL overflows from H'FF to H'00 while CH2 is set to 1

Bit 5: Reserved bit

Bit 5 is reserved; it can be read and written, and is initialized to 0 upon reset.

Bit 4 CH2	Description
0	ECH and ECL are used together as a single-channel 16-bit event counter (in
1	ECH and ECL are used as two independent 8-bit event counter channels

Bit 3: Count-up enable H (CUEH)

Bit 3 enables event clock input to ECH. When 1 is written to this bit, event clock input and increments the counter. When 0 is written to this bit, event clock input is disabled ECH value is held. The AEVH pin or the ECL overflow signal can be selected as the e source by bit CH2.

Bit 3 CUEH	Description
0	ECH event clock input is disabled (in ECH value is held
1	ECH event clock input is enabled

Bit 2: Count-up enable L (CUEL)

Bit 3 enables event clock input to ECL. When 1 is written to this bit, event clock input and increments the counter. When 0 is written to this bit, event clock input is disabled ECL value is held.

Bit 2 CUEL	Description
0	ECL event clock input is disabled (in ECL value is held
1	ECL event clock input is enabled

Bit 0: Counter reset control L (CRCL)

Bit 0 controls resetting of ECL. When this bit is cleared to 0, ECL is reset. When 1 is
this bit, the counter reset is cleared and the ECL count-up function is enabled.

Bit 0 CRCL	Description
0	ECL is reset
1	ECL reset is cleared and count-up function is enabled

2. Event counter H (ECH)

Bit	7	6	5	4	3	2	1
	ECH7	ECH6	ECH5	ECH4	ECH3	ECH2	ECH1
Initial Value	0	0	0	0	0	0	0
Read/Write	R	R	R	R	R	R	R

ECH is an 8-bit read-only up-counter that operates either as an independent 8-bit even
as the upper 8-bit up-counter of a 16-bit event counter configured in combination with
Either the external asynchronous event AEVH pin or the overflow signal from lower 8
ECL can be selected as the input clock source by bit CH2. ECH can be cleared to H'0
software, and is also initialized to H'00 upon reset.

Initial Value	0	0	0	0	0	0	0
Read/Write	R	R	R	R	R	R	R

4. Clock stop register 2 (CKSTPR2)

Bit	7	6	5	4	3	2	1
	—	—	—	—	AECKSTP	WDCKSTP	PWCKSTP
Initial value	1	1	1	1	1	1	1
Read/Write	—	—	—	—	R/W	R/W	R/W

CKSTPR2 is an 8-bit read/write register that performs module standby mode control fo
modules. Only the bit relating to the asynchronous event counter is described here. Fo
the other bits, see the sections on the relevant modules.

Bit 3: Asynchronous event counter module standby mode control (AECKSTP)

Bit 3 controls setting and clearing of module standby mode for the asynchronous event

AECKSTP	Description
0	Asynchronous event counter is set to module standby mode
1	Asynchronous event counter module standby mode is cleared (ir

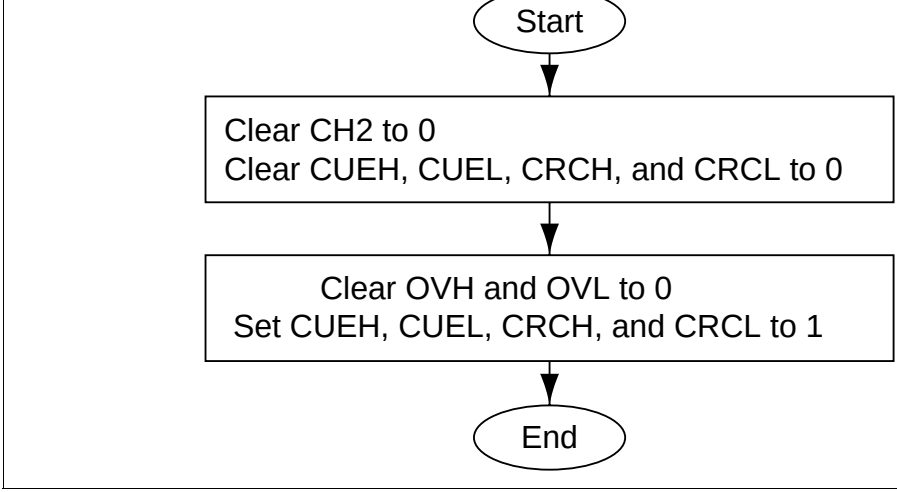


Figure 9.19 Example of Software Processing when Using ECH and ECL as 16-Counter

As CH2 is cleared to 0 by a reset, ECH and ECL operate as a 16-bit event counter afte
They can also be used as a 16-bit event counter by carrying out the software processin
the example in figure 9.19. The operating clock source is asynchronous event input fr
AEVL pin. When the next clock is input after the count value reaches H'FF in both E
ECL, ECH and ECL overflow from H'FFFF to H'0000, the OVH flag is set to 1 in EC
ECH and ECL count values each return to H'00, and counting up is restarted. When o
occurs, the IRREC bit is set to 1 in IRR2. If the IENEC bit in IENR2 is 1 at this time,
request is sent to the CPU.

2. 8-bit event counter operation

When bit CH2 is set to 1 in ECCSR, ECH and ECL operate as independent 8-bit even
Figure 9.20 shows an example of the software processing when ECH and ECL are use
event counters.

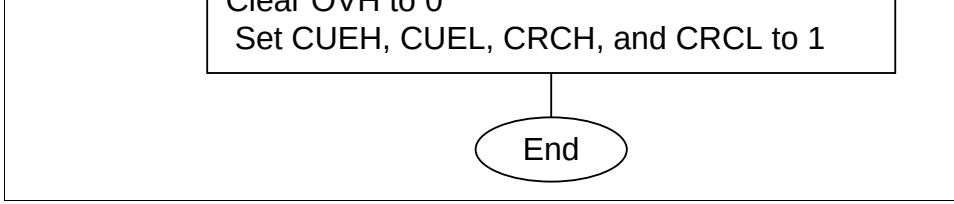


Figure 9.20 Example of Software Processing when Using ECH and ECL as 8-B Counters

ECH and ECL can be used as 8-bit event counters by carrying out the software processi in the example in figure 9.20. The 8-bit event counter operating clock source is asynch event input from the AEVH pin for ECH, and asynchronous event input from the AEV ECL. When the next clock is input after the ECH count value reaches H'FF, ECH over OVH flag is set to 1 in ECCSR, the ECH count value returns to H'00, and counting up i Similarly, when the next clock is input after the ECL count value reaches H'FF, ECL ov the OVL flag is set to 1 in ECCSR, the ECL count value returns to H'00, and counting u restarted. When overflow occurs, the IRREC bit is set to 1 in IRR2. If the IENEC bit i 1 at this time, an interrupt request is sent to the CPU.

9.7.4 Asynchronous Event Counter Operation Modes

Asynchronous event counter operation modes are shown in table 9.21.

Table 9.21 Asynchronous Event Counter Operation Modes

Operation Mode	Reset	Active	Sleep	Watch	Sub-active	Sub-sleep	Standby
ECCSR	Reset	Functions	Functions	Held*	Functions	Functions	Held*
ECH	Reset	Functions	Functions*	Functions*	Functions	Functions	Functions
ECL	Reset	Functions	Functions*	Functions*	Functions	Functions	Functions

Note: * When an asynchronous external event is input, the counter increments but the c overflow H/L flags are not affected.

V), up to 1 MHz (others) for input to the AEVH and AEVL pins, and ensure that th
low widths of the clock are at least 83 ns. The duty cycle is immaterial.

Mode			Maximum AEV Input Clock Fr
16-bit mode			Internal step-do
8-bit mode	Active (high-speed), sleep (high-speed)		not used:
			V_{CC} = 4.5 to 5.5
			V_{CC} = 3.0 to 5.5
			V_{CC} = 2.6 to 5.5
			V_{CC} = 2.2 to 5.5
			Other than abov
			Internal step-do used:
			V_{CC} = 2.2 to 5.5
			Other than abov
8-bit mode	Active (medium-speed), sleep (medium-speed)	(ø/16)	$2 \cdot f_{OSC}$
		(ø/32)	f_{OSC}
		(ø/64)	$1/2 \cdot f_{OSC}$
	f_{OSC} = 400 kHz to 4 MHz	(ø/128)	$1/4 \cdot f_{OSC}$
8-bit mode	Watch, subactive, subsleep, standby	(øw/2)	1000 kHz
		(øw/4)	500 kHz
	øw = 32.768 kHz or 38.4 kHz	(øw/8)	250 kHz

3. When AEC uses with 16-bit mode, set CUEH in ECCSR to “1” first, set CRCH in
“1” second, or set both CUEH and CRCH to “1” at same time before clock entry.
is operating on 16-bit mode, do not change CUEH. Otherwise, ECH will be misco

Serial communication interface 3 (SCI3) can carry out serial data communication in ei
asynchronous or synchronous mode. It is also provided with a multiprocessor commu
function that enables serial data to be transferred among processors.

10.1.1 Features

Features of SCI3 are listed below.

- Choice of asynchronous or synchronous mode for serial data communication
  - — Asynchronous mode

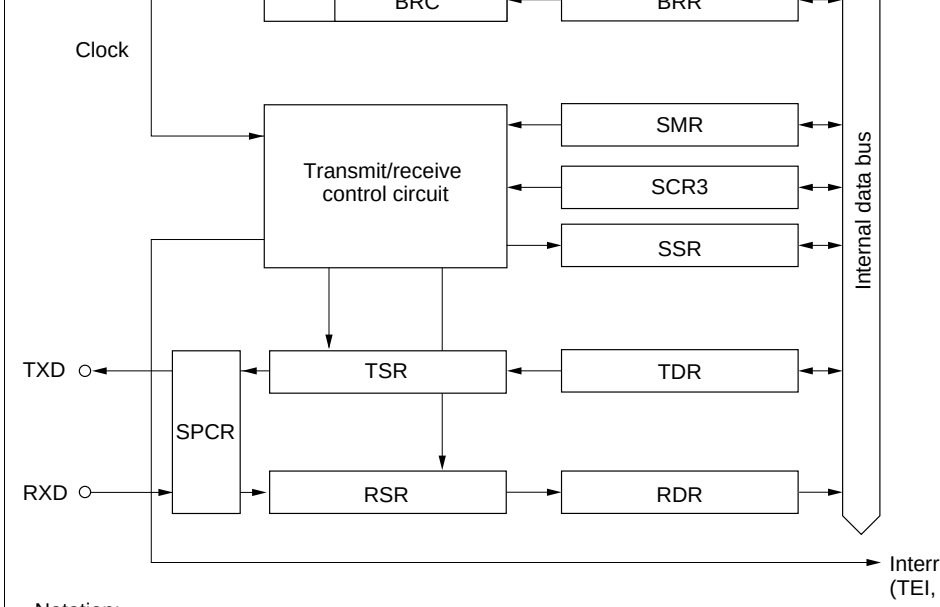
    Serial data communication is performed asynchronously, with synchronization
    character by character. In this mode, serial data can be exchanged with standar
    asynchronous communication LSIs such as a Universal Asynchronous
    Receiver/Transmitter (UART) or Asynchronous Communication Interface Ada
    (ACIA). A multiprocessor communication function is also provided, enabling
    communication among processors.

    There is a choice of 16 data transfer formats.

Data length	7, 8, 5 bits
Stop bit length	1 or 2 bits
Parity	Even, odd, or none
Multiprocessor bit	“1” or “0”
Receive error detection	Parity, overrun, and framing errors
Break detection	Break detected by reading the RXD_{3x} pin level directly framing error occurs

Separate transmission and reception units are provided, enabling transmission and r
be carried out simultaneously. The transmission and reception units are both double
allowing continuous transmission and reception.

- On-chip baud rate generator, allowing any desired bit rate to be selected
- Choice of an internal or external clock as the transmit/receive clock source
- Six interrupt sources: transmit end, transmit data empty, receive data full, overrun e
framing error, and parity error



Notation:

RSR:	Receive shift register
RDR:	Receive data register
TSR:	Transmit shift register
TDR:	Transmit data register
SMR:	Serial mode register
SCR3:	Serial control register 3
SSR:	Serial status register
BRR:	Bit rate register
BRC:	Bit rate counter
SPCR:	Serial port control register

Figure 10.1 SCI3 Block Diagram

SCI3 transmit data output	TXD_{3x}	Output	SCI3 transmit data o

10.1.4 Register configuration

Table 10.2 shows the SCI3 register configuration.

Table 10.2 Registers

Name	Abbrev.	R/W	Initial Value	Add
Serial mode register	SMR	R/W	H'00	H'F
Bit rate register	BRR	R/W	H'FF	H'F
Serial control register 3	SCR3	R/W	H'00	H'F
Transmit data register	TDR	R/W	H'FF	H'F
Serial data register	SSR	R/W	H'84	H'F
Receive data register	RDR	R	H'00	H'F
Transmit shift register	TSR	Protected	—	—
Receive shift register	RSR	Protected	—	—
Bit rate counter	BRC	Protected	—	—
Clock stop register 1	CKSTPR1	R/W	H'FF	H'F
Serial port control register	SPCR	R/W	H'C0	H'F

Read/Write

RSR is a register used to receive serial data. Serial data input to RSR from the RXD_{3} the order in which it is received, starting from the LSB (bit 0), and converted to parall When one byte of data is received, it is transferred to RDR automatically.

RSR cannot be read or written directly by the CPU.

10.2.2 Receive data register (RDR)

Bit	7	6	5	4	3	2	1
	RDR7	RDR6	RDR5	RDR4	RDR3	RDR2	RDR1
Initial value	0	0	0	0	0	0	0
Read/Write	R	R	R	R	R	R	R

RDR is an 8-bit register that stores received serial data.

When reception of one byte of data is finished, the received data is transferred from R and the receive operation is completed. RSR is then able to receive data. RSR and RI double-buffered, allowing consecutive receive operations.

RDR is a read-only register, and cannot be written by the CPU.

RDR is initialized to H'00 upon reset, and in standby, module standby or watch mode.

and serial data transmission is carried out by sending the data to the TXD_{3x} pin in order from the LSB (bit 0). When one byte of data is transmitted, the next byte of transmit da transferred to TDR, and transmission started, automatically. Data transfer from TDR to not performed if no data has been written to TDR (if bit TDRE is set to 1 in the serial st register (SSR)).

TSR cannot be read or written directly by the CPU.

10.2.4 Transmit data register (TDR)

Bit	7	6	5	4	3	2	1
	TDR7	TDR6	TDR5	TDR4	TDR3	TDR2	TDR1
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

TDR is an 8-bit register that stores transmit data. When TSR is found to be empty, the data written in TDR is transferred to TSR, and serial data transmission is started. Conti transmission is possible by writing the next transmit data to TDR during TSR serial dat transmission.

TDR can be read or written by the CPU at any time.

TDR is initialized to H'FF upon reset, and in standby, module standby, or watch mode.

the baud rate generator.

SMR can be read or written by the CPU at any time.

SMR is initialized to H'00 upon reset, and in standby, module standby, or watch mode

Bit 7: Communication mode (COM)

Bit 7 selects whether SCI3 operates in asynchronous mode or synchronous mode.

Bit 7 COM	Description	
0	Asynchronous mode	(
1	Synchronous mode	

Bit 6: Character length (CHR)

Bit 6 selects either 7 or 8 bits as the data length to be used in asynchronous mode. In s
mode the data length is always 8 bits, irrespective of the bit 6 setting.

Bit 6 CHR	Description	
0	8-bit data/5-bit data[*2]	(
1	7-bit data[*1]/5-bit data[*2]	

Notes: 1. When 7-bit data is selected, the MSB (bit 7) of TDR is not transmitted.
2. When 5-bit data is selected, set both PE and MP to 1. The three most sign
(bits 7, 6, and 5) of TDR are not transmitted.

1	Parity bit addition and checking enabled [1, 2]

Notes: 1. When PE is set to 1, even or odd parity, as designated by bit PM, is added to data before it is sent, and the received parity bit is checked against the parity designated by bit PM.
2. For the case where 5-bit data is selected, see table 10.11.

Bit 4: Parity mode (PM)

Bit 4 selects whether even or odd parity is to be used for parity addition and checking. setting is only valid in asynchronous mode when bit PE is set to 1, enabling parity bit a checking. The PM bit setting is invalid in synchronous mode, and in asynchronous mo bit addition and checking is disabled.

Bit 4 PM	Description
0	Even parity[*1] (i
1	Odd parity[*2]

Notes: 1. When even parity is selected, a parity bit is added in transmission so that the number of 1 bits in the transmit data plus the parity bit is an even number; in a check is carried out to confirm that the number of 1 bits in the receive data parity bit is an even number.
2. When odd parity is selected, a parity bit is added in transmission so that the number of 1 bits in the transmit data plus the parity bit is an odd number; in r check is carried out to confirm that the number of 1 bits in the receive data p parity bit is an odd number.

1	2 stop bits [2]

Notes: 1. In transmission, a single 1 bit (stop bit) is added at the end of a transmit cha
2. In transmission, two 1 bits (stop bits) are added at the end of a transmit cha

In reception, only the first of the received stop bits is checked, irrespective of the STO
If the second stop bit is 1 it is treated as a stop bit, but if 0, it is treated as the start bit o
transmit character.

Bit 2: Multiprocessor mode (MP)

Bit 2 enables or disables the multiprocessor communication function. When the multi
communication function is disabled, the parity settings in the PE and PM bits are inva
bit setting is only valid in asynchronous mode. When synchronous mode is selected th
should be set to 0. For details on the multiprocessor communication function, see 10.1
Multiprocessor Communication Function.

Bit 2 MP	Description	
0	Multiprocessor communication function disabled*	(
1	Multiprocessor communication function enabled*	

Note: * For the case where 5-bit data is selected, see table 10.11.

0	1	ø w/2 clock[*1]/ø w clock[*2]
1	0	ø/16 clock
1	1	ø/64 clock

Notes: 1. ø w/2 clock in active (medium-speed/high-speed) mode and sleep mode
2. ø w clock in subactive mode and subsleep mode
3. In subactive or subsleep mode, SCI3 can be operated when CPU clock is øw

10.2.6 Serial control register 3 (SCR3)

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

SCR3 is an 8-bit register for selecting transmit or receive operation, the asynchronous m output, interrupt request enabling or disabling, and the transmit/receive clock source.

SCR3 can be read or written by the CPU at any time.

SCR3 is initialized to H'00 upon reset, and in standby, module standby or watch mode.

0	Transmit data empty interrupt request (TXI) disabled	(
1	Transmit data empty interrupt request (TXI) enabled	

Bit 6: Receive interrupt enable (RIE)

Bit 6 selects enabling or disabling of the receive data full interrupt request (RXI) and t
error interrupt request (ERI) when receive data is transferred from the receive shift reg
to the receive data register (RDR), and bit RDRF in the serial status register (SSR) is s
There are three kinds of receive error: overrun, framing, and parity.

RXI can be released by clearing bit RDRF or the FER, PER, or OER error flag to 0, o
bit RIE to 0.

Bit 6 RIE	Description
0	Receive data full interrupt request (RXI) and receive error interrupt request (ERI) disabled
1	Receive data full interrupt request (RXI) and receive error interrupt request (ERI) enabled

Bit 5: Transmit enable (TE)

Bit 5 selects enabling or disabling of the start of transmit operation.

Bit 5 TE	Description	
0	Transmit operation disabled[*1] (TXD pin is I/O port)	(
1	Transmit operation enabled[*2] (TXD pin is transmit data pin)	

Notes: 1. Bit TDRE in SSR is fixed at 1.

2. When transmit data is written to TDR in this state, bit TDR in SSR is cleare
serial data transmission is started. Be sure to carry out serial mode registe
settings, and setting of bit SPC31 or SPC32 in SPCR, to decide the transm
before setting bit TE to 1.

cleared to 0, and retain their previous state.

2. In this state, serial data reception is started when a start bit is detected in asy mode or serial clock input is detected in synchronous mode. Be sure to carr mode register (SMR) settings to decide the reception format before setting b

Bit 3: Multiprocessor interrupt enable (MPIE)

Bit 3 selects enabling or disabling of the multiprocessor interrupt request. The MPIE b only valid when asynchronous mode is selected and reception is carried out with bit MP set to 1. The MPIE bit setting is invalid when bit COM is set to 1 or bit MP is cleared t

Bit 3 MPIE	Description
0	Multiprocessor interrupt request disabled (normal receive operation) (in Clearing conditions: When data is received in which the multiprocessor bit is set to 1
1	Multiprocessor interrupt request enabled*

Note: * Receive data transfer from RSR to RDR, receive error detection, and setting of th FER, and OER status flags in SSR is not performed. RXI, ERI, and setting of the FER, and OER flags in SSR, are disabled until data with the multiprocessor bit s received. When a receive character with the multiprocessor bit set to 1 is receiv MPBR in SSR is set to 1, bit MPIE is automatically cleared to 0, and RXI and ER (when bits TIE and RIE in serial control register 3 (SCR3) are set to 1) and settin RDRF, FER, and OER flags are enabled.

Note: * TEI can be released by clearing bit TDRE to 0 and clearing bit TEND to 0 in SS
clearing bit TEIE to 0.

Bits 1 and 0: Clock enable 1 and 0 (CKE1, CKE0)

Bits 1 and 0 select the clock source and enabling or disabling of clock output from the
The combination of CKE1 and CKE0 determines whether the SCK_{3x} pin functions as
a clock output pin, or a clock input pin.

The CKE0 bit setting is only valid in case of internal clock operation (CKE1 = 0) in as
mode. In synchronous mode, or when external clock operation is used (CKE1 = 1), bi
should be cleared to 0.

After setting bits CKE1 and CKE0, set the operating mode in the serial mode register

For details on clock source selection, see table 10.4 in 10.1.3, Operation.

Bit 1	Bit 0	Description		
CKE1	CKE0	Communication Mode	Clock Source	SCK_{3x} Pin Fu
0	0	Asynchronous	Internal clock	I/O port[*1]
		Synchronous	Internal clock	Serial clock ou
0	1	Asynchronous	Internal clock	Clock output[*2]
		Synchronous	Reserved	
1	0	Asynchronous	External clock	Clock input[*3]
		Synchronous	External clock	Serial clock inp
1	1	Asynchronous	Reserved	
		Synchronous	Reserved	

Notes: 1. Initial value
2. A clock with the same frequency as the bit rate is output.
3. Input a clock with a frequency 16 times the bit rate.

SSR is an 8-bit register containing status flags that indicate the operational status of SC
multiprocessor bits.

SSR can be read or written by the CPU at any time, but only a write of 1 is possible to b
RDRF, OER, PER, and FER. In order to clear these bits by writing 0, 1 must first be re

Bits TEND and MPBR are read-only bits, and cannot be modified.

SSR is initialized to H'84 upon reset, and in standby, module standby, or watch mode.

Bit 7: Transmit data register empty (TDRE)

Bit 7 indicates that transmit data has been transferred from TDR to TSR.

Bit 7 TDRE	Description
0	Transmit data written in TDR has not been transferred to TSR Clearing conditions: After reading TDRE = 1, cleared by writing 0 to TDRE When data is written to TDR by an instruction
1	Transmit data has not been written to TDR, or transmit data written in TDR has been transferred to TSR Setting conditions: When bit TE in SCR3 is cleared to 0 When data is transferred from TDR to TSR (in

1	There is receive data in RDR Setting conditions: When reception ends normally and receive data is transferred from RSR

Note: If an error is detected in the receive data, or if the RE bit in SCR3 has been clea
RDR and bit RDRF are not affected and retain their previous state.
Note that if data reception is completed while bit RDRF is still set to 1, an overru
(OER) will result and the receive data will be lost.

Bit 5: Overrun error (OER)

Bit 5 indicates that an overrun error has occurred during reception.

Bit 5 OER	Description
0	Reception in progress or completed[*1] Clearing conditions: After reading OER = 1, cleared by writing 0 to OER
1	An overrun error has occurred during reception[*2] Setting conditions: When reception is completed with RDRF set to 1

Notes: 1. When bit RE in SCR3 is cleared to 0, bit OER is not affected and retains its state.
2. RDR retains the receive data it held before the overrun error occurred, and received after the error is lost. Reception cannot be continued with bit OER and in synchronous mode, transmission cannot be continued either.

1	A framing error has occurred during reception Setting conditions: When the stop bit at the end of the receive data is checked for a value of 1 at the end of reception, and the stop bit is 0[*2]

Notes: 1. When bit RE in SCR3 is cleared to 0, bit FER is not affected and retains its p state.

2. Note that, in 2-stop-bit mode, only the first stop bit is checked for a value of 1 second stop bit is not checked. When a framing error occurs the receive dat transferred to RDR but bit RDRF is not set. Reception cannot be continued FER set to 1. In synchronous mode, neither transmission nor reception is po when bit FER is set to 1.

Bit 3: Parity error (PER)

Bit 3 indicates that a parity error has occurred during reception with parity added in asy mode.

Bit 3 PER	Description
0	Reception in progress or completed[*1] (in Clearing conditions: After reading PER = 1, cleared by writing 0 to PER
1	A parity error has occurred during reception[*2] Setting conditions: When the number of 1 bits in the receive data plus parity bit does not match the parity designated by bit PM in the serial mode register (SMR)

Notes: 1. When bit RE in SCR3 is cleared to 0, bit PER is not affected and retains its p state.

2. Receive data in which it a parity error has occurred is still transferred to RDR RDRF is not set. Reception cannot be continued with bit PER set to 1. In sy mode, neither transmission nor reception is possible when bit FER is set to 1

	After reading TDRE = 1, cleared by writing 0 to TDRE When data is written to TDR by an instruction	
1	Transmission ended Setting conditions: When bit TE in SCR3 is cleared to 0 When bit TDRE is set to 1 when the last bit of a transmit character is sen	(

Bit 1: Multiprocessor bit receive (MPBR)

Bit 1 stores the multiprocessor bit in a receive character during multiprocessor format asynchronous mode.

Bit 1 is a read-only bit and cannot be modified.

Bit 1 MPBR	**Description**	
0	Data in which the multiprocessor bit is 0 has been received*	(
1	Data in which the multiprocessor bit is 1 has been received	

Note: * When bit RE is cleared to 0 in SCR3 with the multiprocessor format, bit MPBR i affected and retains its previous state.

Bit 0: Multiprocessor bit transfer (MPBT)

Bit 0 stores the multiprocessor bit added to transmit data when transmitting in asynchr mode. The bit MPBT setting is invalid when synchronous mode is selected, when the multiprocessor communication function is disabled, and when not transmitting.

Bit 0 MPBT	**Description**	
0	A 0 multiprocessor bit is transmitted	(
1	A 1 multiprocessor bit is transmitted	

rate generator operating clock selected by bits CKS1 and CKS0 of the serial mode regis

BRR can be read or written by the CPU at any time.

BRR is initialized to H'FF upon reset, and in standby, module standby, or watch mode.

Table 10.3 shows examples of BRR settings in asynchronous mode. The values shown active (high-speed) mode.

Table 10.3 Examples of BRR Settings for Various Bit Rates (Asynchronous Mod

	OSC											
	32.8 kHz			38.4 kHz			2 MHz			2.4576 MHz		
B Bit Rate (bit/s)	n	N	Error (%)	n	N	Error (%)	n	N	Error (%)	n	N	Error (%)	n
110	Cannot be used,			—	—	—	—	—	—	2	21	–0.83	—
150	as error			0	3	0	2	12	0.16	3	3	0	2
200	exceeds 3%			0	2	0	0	155	0.16	3	2	0	—
250				—	—	—	0	124	0	0	153	–0.26	0
300				0	1	0	0	103	0.16	3	1	0	2
600				0	0	0	0	51	0.16	3	0	0	0
1200				—	—	—	0	25	0.16	2	1	0	0
2400				—	—	—	0	12	0.16	2	0	0	0
4800				—	—	—	—	—	—	0	7	0	0
9600				—	—	—	—	—	—	0	3	0	—
19200				—	—	—	—	—	—	0	1	0	—
31250				—	—	—	—	—	—	—	—	—	0
38400				—	—	—	—	—	—	0	0	0	—

n: Baud rate generator input clock number (n = 0, 2, or 3)

(The relation between n and the clock is shown in table 10.4.)

3. The error in table 10.3 is the value obtained from the following equation, r
two decimal places.

$$\text{Error (\%)} = \frac{\text{B (rate obtained from n, N, OSC)} - \text{R(bit rate in left-hand column in table 10.3.)}}{\text{R (bit rate in left-hand column in table 10.3.)}}$$

Table 10.4 Relation between n and Clock

		SMR Setting	
n	**Clock**	**CKS1**	**CKS0**
0	ø	0	0
0	$ø_W/2^{*1}/ø_W{}^{*2}$	0	1
2	ø/16	1	0
3	ø/64	1	1

Notes: 1. ø w/2 clock in active (medium-speed/high-speed) mode and sleep mode
2. ø w clock in subactive mode and subsleep mode
3. In subactive or subsleep mode, SCI3 can be operated when CPU clock is ø

Table 10.5 shows the maximum bit rate for each frequency. The values shown are for
(high-speed) mode.

* : When SMR is set up to CKS1 = “0”, CKS0 = “1”.

Table 10.6 shows examples of BRR settings in synchronous mode. The values shown a
active (high-speed) mode.

Table 10.6 Examples of BRR Settings for Various Bit Rates (Synchronous Mode)

	OSC							
B Bit Rate	**38.4 kHz**			**2 MHz**			**4 MH**	
(bit/s)	**n**	**N**	**Error**	**n**	**N**	**Error**	**n**	**N**
200	0	23	0	—	—	—	—	—
250	—	—	—	—	—	—	2	124
300	2	0	0	—	—	—	—	—
500				—	—	—	—	—
1k				0	249	0	—	—
2.5k				0	99	0	0	199
5k				0	49	0	0	99
10k				0	24	0	0	49
25k				0	9	0	0	19
50k				0	4	0	0	9
100k				—	—	—	0	4
250k				0	0	0	0	1
500k							0	0
1M								

Blank: Cannot be set.
— : A setting can be made, but an error will result.

(The relation between n and the clock is shown in table 10.7.)

Table 10.7 Relation between n and Clock

		SMR Setting	
n	**Clock**	**CKS1**	**CKS0**
0	ø	0	0
0	$ø_W/2^{*1}/ø_W{}^{*2}$	0	1
2	ø/16	1	0
3	ø/64	1	1

Notes: 1. ø w/2 clock in active (medium-speed/high-speed) mode and sleep mode
2. ø w clock in subactive mode and subsleep mode
3. In subactive or subsleep mode, SCI3 can be operated when CPU clock is ø

modules. Only the bits relating to SCI3 are described here. For details of the other bits
sections on the relevant modules.

Bit 6: SCI3-1 module standby mode control (S31CKSTP)

Bit 6 controls setting and clearing of module standby mode for SCI31.

S31CKSTP	Description
0	SCI3-1 is set to module standby mode
1	SCI3-1 module standby mode is cleared (in

Note: All SCI31 register is initialized in module standby mode.

Bit 5: SCI3-2 module standby mode control (S32CKSTP)

Bit 5 controls setting and clearing of module standby mode for SCI32.

S32CKSTP	Description
0	SCI3-2 is set to module standby mode
1	SCI3-2 module standby mode is cleared (in

Note: All SCI32 register is initialized in module standby mode.

input/output data inversion switching. SPCR is initialized to H'C0 by a reset.

Bit 0: RXD_{31} pin input data inversion switch

Bit 0 specifies whether or not RXD_{31} pin input data is to be inverted.

Bit 0 SCINV0	Description
0	RXD_{31} input data is not inverted (I
1	RXD_{31} input data is inverted

Bit 1: TXD_{31} pin output data inversion switch

Bit 1 specifies whether or not TXD_{31} pin output data is to be inverted.

Bit 1 SCINV1	Description
0	TXD_{31} output data is not inverted (I
1	TXD_{31} output data is inverted

Bit 2: RXD_{32} pin input data inversion switch

Bit 2 specifies whether or not RXD_{32} pin input data is to be inverted.

Bit 2 SCINV2	Description
0	RXD_{32} input data is not inverted (I
1	RXD_{32} input data is inverted

Bit 4: $P3_5/TXD_{31}$ pin function switch (SPC31)

This bit selects whether pin $P3_5/TXD_{31}$ is used as $P3_5$ or as TXD_{31}.

Bit 4 SPC31	Description	
0	Functions as $P3_5$ I/O pin	(in
1	Functions as TXD_{31} output pin*	

Note: * Set the TE bit in SCR3 after setting this bit to 1.

Bit 5: $P4_2/TXD_{32}$ pin function switch (SPC32)

This bit selects whether pin $P4_2/TXD_{32}$ is used as $P4_2$ or as TXD_{32}.

Bit 5 SPC32	Description	
0	Functions as $P4_2$ I/O pin	(in
1	Functions as TXD_{32} output pin*	

Note: * Set the TE bit in SCR3 after setting this bit to 1.

Bits 7 to 6: Reserved bits

Bits 7 to 6 are reserved; they are always read as 1 and cannot be modified.

The clock source for SCI3 is determined by bit COM in SMR and bits CKE1 and CKE
as shown in table 10.9.

1. Synchronous mode

- Choice of 5-, 7-, or 8-bit data length
- Choice of parity addition, multiprocessor bit addition, and addition of 1 or 2 stop b
  combination of these parameters determines the data transfer format and the charac
- Framing error (FER), parity error (PER), overrun error (OER), and break detection
  reception
- Choice of internal or external clock as the clock source

  When internal clock is selected: SCI3 operates on the baud rate generator clock, an
  with the same frequency as the bit rate can be output.

  When external clock is selected: A clock with a frequency 16 times the bit rate mu
  (The on-chip baud rate generator is not used.)

2. Synchronous mode

- Data transfer format: Fixed 8-bit data length
- Overrun error (OER) detection during reception
- Choice of internal or external clock as the clock source

  When internal clock is selected: SCI3 operates on the baud rate generator clock, an
  clock is output.

  When external clock is selected: The on-chip baud rate generator is not used, and S
  operates on the input serial clock.

				1				
	1		0	0		7-bit data		No
				1				
			1	0				Yes
				1				
	0	1	0	0		8-bit data	Yes	No
				1				
			1	0		5-bit data	No	
				1				
	1		0	0		7-bit data	Yes	
				1				
			1	0		5-bit data	No	Yes
				1				
1	*	0	*	*	Synchronous mode	8-bit data	No	No

*

1	0	0	Synchronous	Internal	Outputs serial clock
	1	0	mode	External	Inputs serial clock
0	1	1	Reserved (Do not specify these combinations)		
1	0	1			
1	1	1			

3. Interrupts and continuous transmission/reception

SCI3 can carry out continuous reception using RXI and continuous transmission using
These interrupts are shown in table 10.10.

Table 10.10 Transmit/Receive Interrupts

Interrupt	Flags	Interrupt Request Conditions	Notes
RXI	RDRF RIE	When serial reception is performed normally and receive data is transferred from RSR to RDR, bit RDRF is set to 1, and if bit RIE is set to 1 at this time, RXI is enabled and an interrupt is requested. (See figure 10.2 (a).)	The RXI interrupt routine receive data transferred clears bit RDRF to 0. C reception can be perfor repeating the above ope reception of the next RS completed.
TXI	TDRE TIE	When TSR is found to be empty (on completion of the previous transmission) and the transmit data placed in TDR is transferred to TSR, bit TDRE is set to 1. If bit TIE is set to 1 at this time, TXI is enabled and an interrupt is requested. (See figure 10.2 (b).)	The TXI interrupt routine next transmit data to TD bit TDRE to 0. Continuo transmission can be per repeating the above ope the data transferred to T been transmitted.
TEI	TEND TEIE	When the last bit of the character in TSR is transmitted, if bit TDRE is set to 1, bit TEND is set to 1. If bit TEIE is set to 1 at this time, TEI is enabled and an interrupt is requested. (See figure 10.2 (c).)	TEI indicates that the ne data has not been writte when the last bit of the t character in TSR is sent

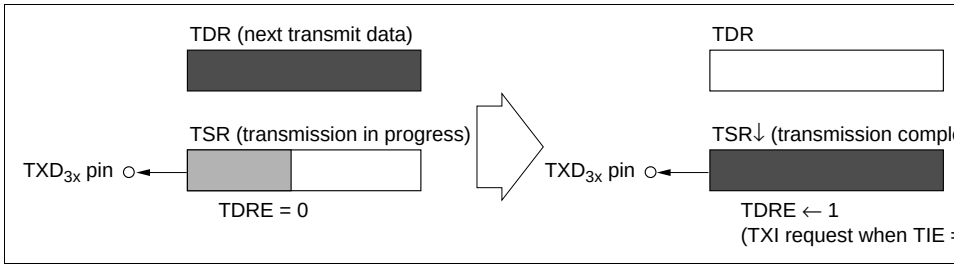


Figure 10.2 (b) TDRE Setting and TXI Interrupt

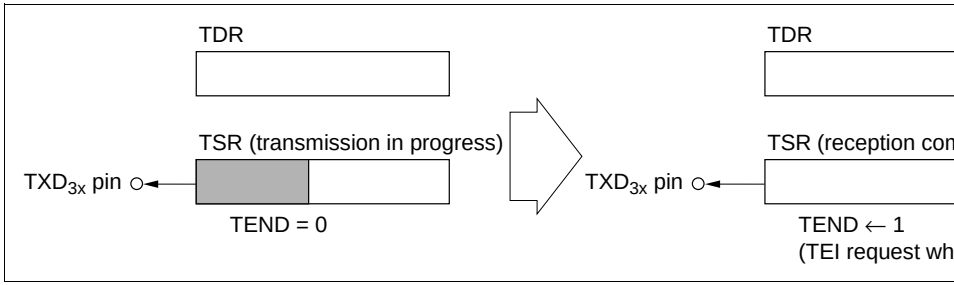


Figure 10.2 (c) TEND Setting and TEI Interrupt

1. Data transfer format

The general data transfer format in asynchronous communication is shown in figure 1

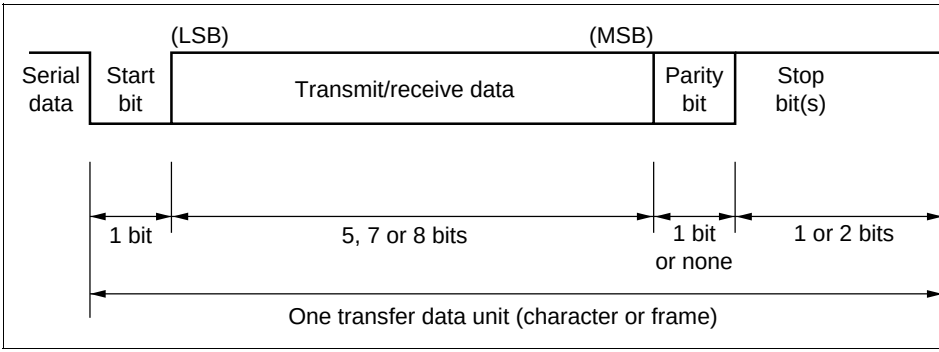
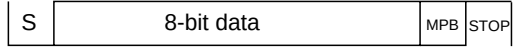
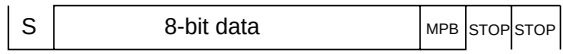
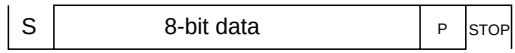
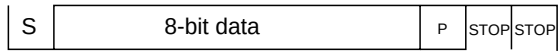
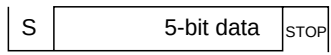
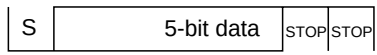
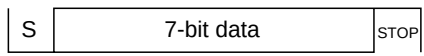
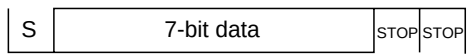
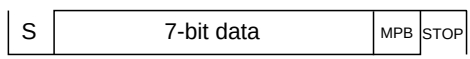
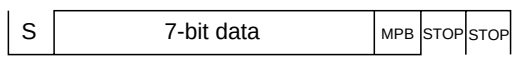
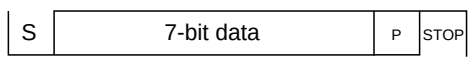
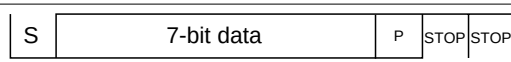
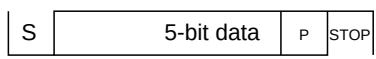
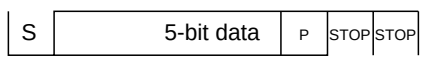


Figure 10.3 Data Format in Asynchronous Communication

In asynchronous communication, the communication line is normally in the mark state level). SCI3 monitors the communication line and when it detects a space (low level), this as a start bit and begins serial data communication.

One transfer data character consists of a start bit (low level), followed by transmit/rece (LSB-first format, starting from the least significant bit), a parity bit (high or low level finally one or two stop bits (high level).

In asynchronous mode, synchronization is performed by the falling edge of the start bi reception. The data is sampled on the 8th pulse of a clock with a frequency 16 times t period, so that the transfer data is latched at the center of each bit.

0	0	1	0	
0	0	1	1	
0	1	0	0	
0	1	0	1	
0	1	1	0	
0	1	1	1	
1	0	0	0	
1	0	0	1	
1	0	1	0	
1	0	1	1	
1	1	0	0	
1	1	0	1	
1	1	1	0	
1	1	1	1	

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

RENESAS

When SCI3 operates on an internal clock, the clock can be output at the SCK_{3x} pin. In the frequency of the output clock is the same as the bit rate, and the phase is such that t rises at the center of each bit of transmit/receive data, as shown in figure 10.4.

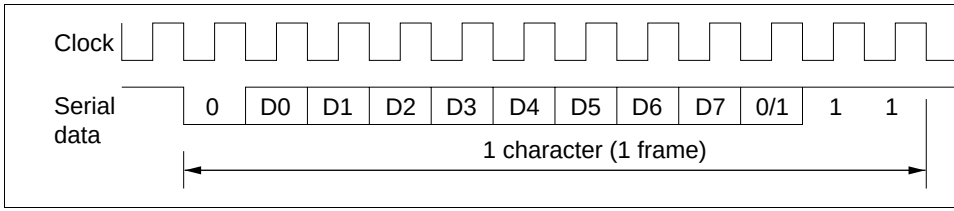


Figure 10.4 Phase Relationship between Output Clock and Transfer Da (Asynchronous Mode) (8-bit data, parity, 2 stop bits)

3. Data transfer operations

• SCI3 initialization

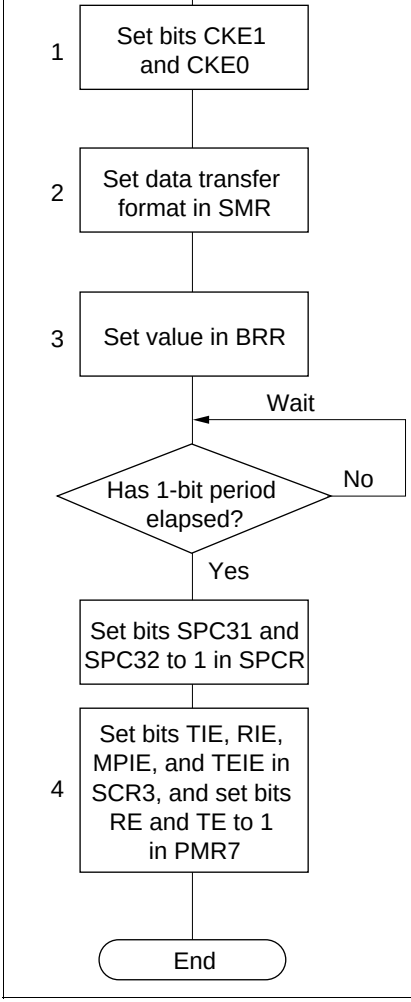
Before data is transferred on SCI3, bits TE and RE in SCR3 must first be cleared to 0, a SCI3 must be initialized as follows.

Note: If the operation mode or data transfer format is changed, bits TE and RE must f cleared to 0.

When bit TE is cleared to 0, bit TDRE is set to 1.

Note that the RDRF, PER, FER, and OER flags and the contents of RDR are re when RE is cleared to 0.

When an external clock is used in asynchronous mode, the clock should not be during operation, including initialization. When an external clock is used in sy mode, the clock should not be supplied during operation, including initializatio



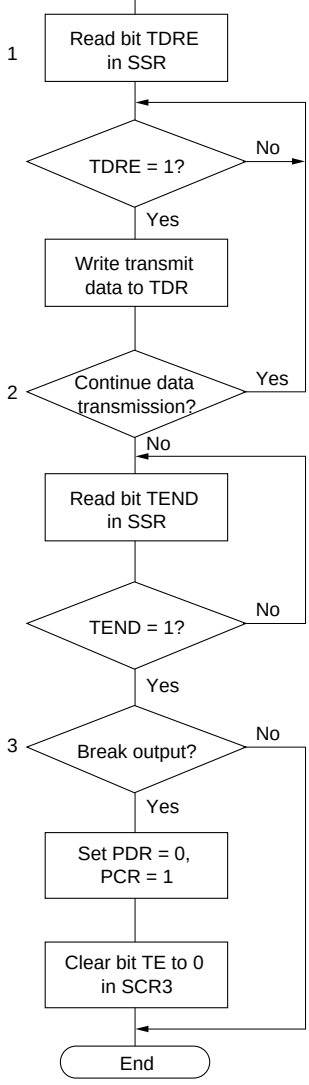
1. Set clock selection in SCR3. Be sure clear the other bits to 0. If clock outpu is selected in asynchronous mode, th clock is output immediately after setti bits CKE1 and CKE0. If clock output selected for reception in synchronous mode, the clock is output immediately after bits CKE1, CKE0, and RE are set to 1.

2. Set the data transfer format in the ser mode register (SMR).

3. Write the value corresponding to the transfer rate in BRR. This operation i not necessary when an external clock is selected.

4. Wait for at least one bit period, then s bits TIE, RIE, MPIE, and TEIE in SCR and set bits RE and TE to 1 in PMR7. Setting bits TE and RE enables the T and RXD3x pins to be used. In asyn mode the mark state is established w transmitting, and the idle state waiting a start bit when receiving.

Figure 10.5 Example of SCI3 Initialization Flowchart



1. Read the serial status register (SSR) and check that bit TDRE is set to 1, then write transmit data to the transmit data register (TDR). When data is written to TDR, bit TDRE is cleared to 0 automatically.
   (After the TE bit is set to 1, one frame of 1s is output, then transmission is possible.)

2. When continuing data transmission, be sure to read TDRE = 1 to confirm that a write can be performed before writing data to TDR. When data is written to TDR, bit TDRE is cleared to 0 automatically.

3. If a break is to be output when data transmission ends, set the port PCR to 1 and clear the port PDR to 0, then clear bit TE in SCR3 to 0.

Figure 10.6 Example of Data Transmission Flowchart (Asynchronous Mo

next frame. If bit TDRE is set to 1, bit TEND in SSR bit is set to 1the mark state, in w
transmitted, is established after the stop bit has been sent. If bit TEIE in SCR3 is set t
time, a TEI request is made.

Figure 10.12 shows an example of the operation when transmitting in asynchronous m

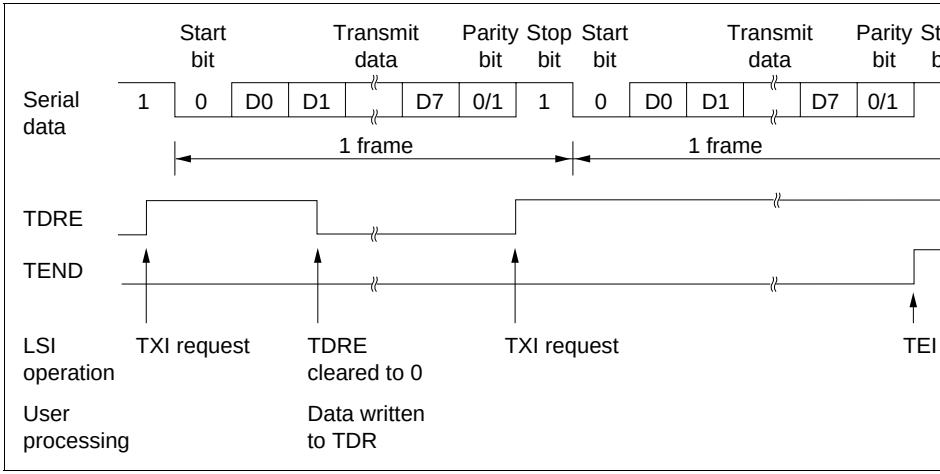
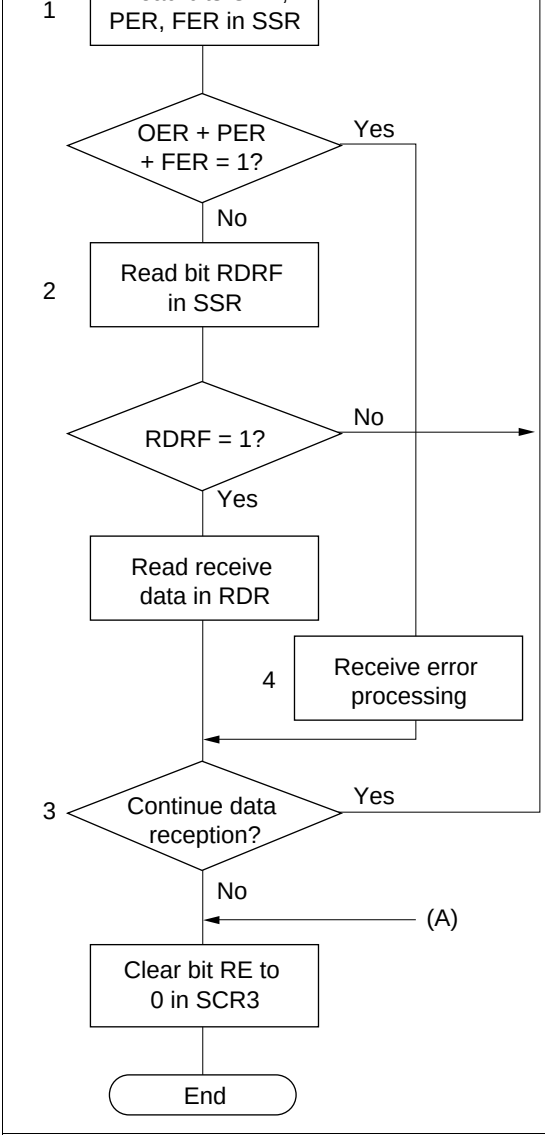


Figure 10.7 Example of Operation when Transmitting in Asynchronous M
(8-bit data, parity, 1 stop bit)



1. Read bits OER, PER, and FER
serial status register (SSR) to de
if there is an error. If a receive e
occurred, execute receive error
processing.

2. Read SSR and check that bit RD
set to 1. If it is, read the receive
in RDR. When the RDR data is
bit RDRF is cleared to 0 automa

3. When continuing data reception
reading of bit RDRF and RDR b
receiving the stop bit of the curr
frame. When the data in RDR is
bit RDRF is cleared to 0 automa

Figure 10.8 Example of Data Reception Flowchart (Asynchronous Mod

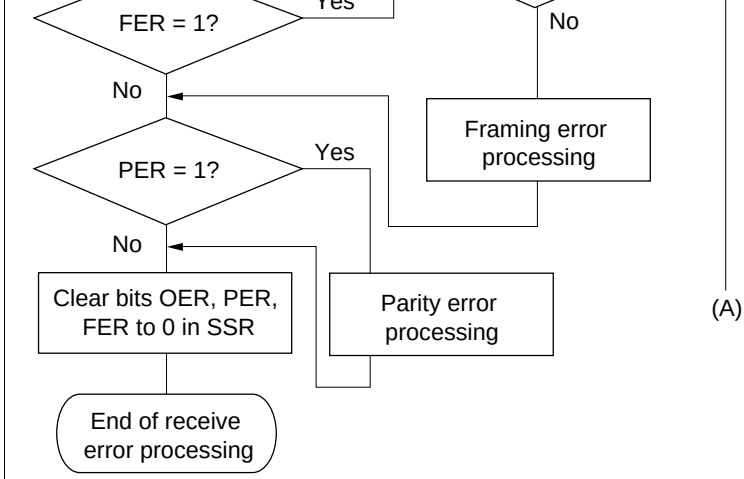


Figure 10.8 Example of Data Reception Flowchart (Asynchronous Mode)

set in bit PM in the serial mode register (SMR).

- Stop bit check

  SCI3 checks that the stop bit is 1. If two stop bits are used, only the first is checked

- Status check

  SCI3 checks that bit RDRF is set to 0, indicating that the receive data can be transfe RSR to RDR.

If no receive error is found in the above checks, bit RDRF is set to 1, and the receive da in RDR. If bit RIE is set to 1 in SCR3, an RXI interrupt is requested. If the error check receive error, bit OER, PER, or FER is set to 1 depending on the kind of error. Bit RDI its state prior to receiving the data. If bit RIE is set to 1 in SCR3, an ERI interrupt is re

Table 10.12 shows the conditions for detecting a receive error, and receive data process

Note: No further receive operations are possible while a receive error flag is set. Bits FER, PER, and RDRF must therefore be cleared to 0 before resuming reception

Table 10.12 Receive Error Detection Conditions and Receive Data Processing

Receive Error	Abbreviation	Detection Conditions	Receive Data Proce
Overrun error	OER	When the next date receive operation is completed while bit RDRF is still set to 1 in SSR	Receive data is not t from RSR to RDR
Framing error	FER	When the stop bit is 0	Receive data is trans from RSR to RDR
Parity error	PER	When the parity (odd or even) set in SMR is different from that of the received data	Receive data is trans from RSR to RDR

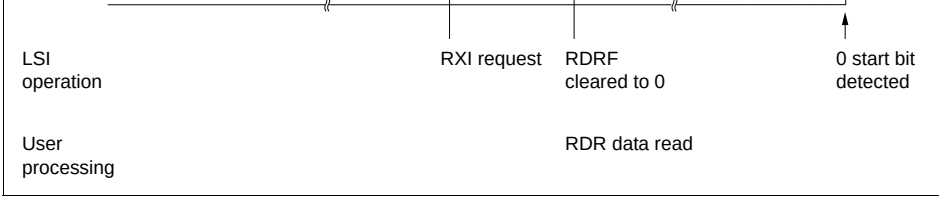


**Figure 10.9 Example of Operation when Receiving in Asynchronous M
(8-bit data, parity, 1 stop bit)**

10.3.3 Operation in Synchronous Mode

In synchronous mode, SCI3 transmits and receives data in synchronization with clock
mode is suitable for high-speed serial communication.

SCI3 has separate transmission and reception units, allowing full-duplex communicati
shared clock.

As the transmission and reception units are both double-buffered, data can be written
transmission and read during reception, making possible continuous transmission and

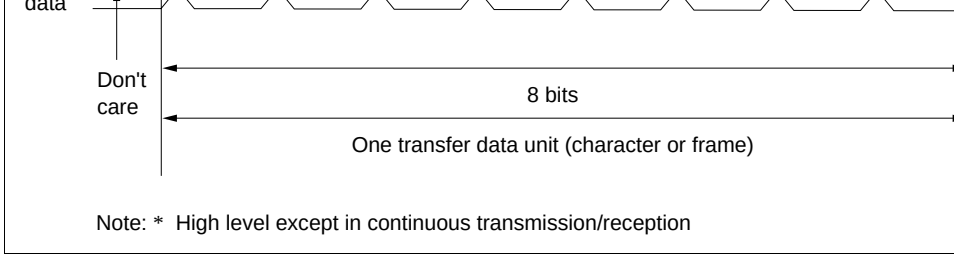


Figure 10.10 Data Format in Synchronous Communication

In synchronous communication, data on the communication line is output from one fall
the serial clock until the next falling edge. Data confirmation is guaranteed at the rising
the serial clock.

One transfer data character begins with the LSB and ends with the MSB. After output
MSB, the communication line retains the MSB state.

When receiving in synchronous mode, SCI3 latches receive data at the rising edge of th
clock.

The data transfer format uses a fixed 8-bit data length.

Parity and multiprocessor bits cannot be added.

2. Clock

Either an internal clock generated by the baud rate generator or an external clock input
SCK_{3x} pin can be selected as the SCI3 serial clock. The selection is made by means of
in SMR and bits CKE1 and CKE0 in SCR3. See table 10.9 for details on clock source

When SCI3 operates on an internal clock, the serial clock is output at the SCK_{3x} pin. E
of the serial clock are output in transmission or reception of one character, and when SC
transmitting or receiving, the clock is fixed at the high level.

followed for data transmission after initializing SCI3.

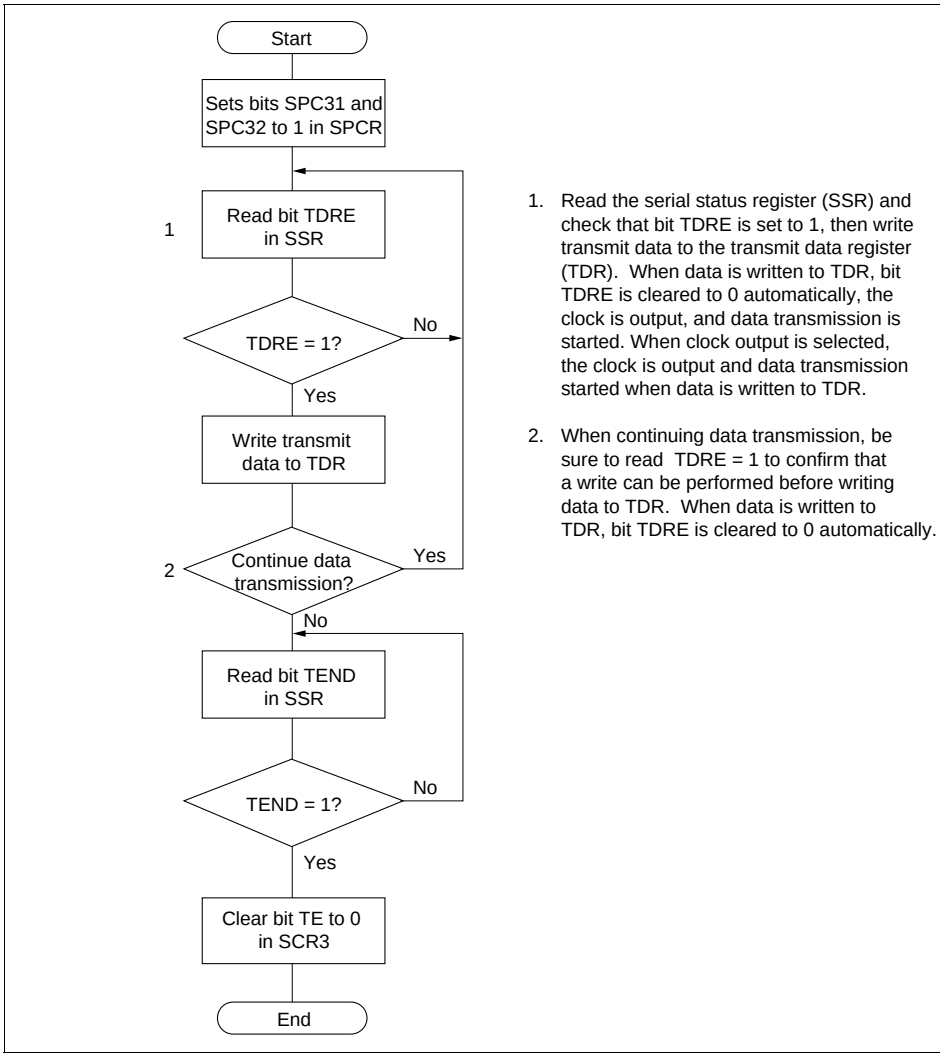


Figure 10.11 Example of Data Transmission Flowchart (Synchronous M

When the MSB (bit 7) is sent, checks bit TDRE. If bit TDRE is cleared to 0, SCI3 tran from TDR to TSR, and starts transmission of the next frame. If bit TDRE is set to 1, S TEND to 1 in SSR, and after sending the MSB (bit 7), retains the MSB state. If bit TE is set to 1 at this time, a TEI request is made.

After transmission ends, the SCK pin is fixed at the high level.

Note: Transmission is not possible if an error flag (OER, FER, or PER) that indicates reception status is set to 1. Check that these error flags are all cleared to 0 befo transmit operation.

Figure 10.12 shows an example of the operation when transmitting in synchronous mo

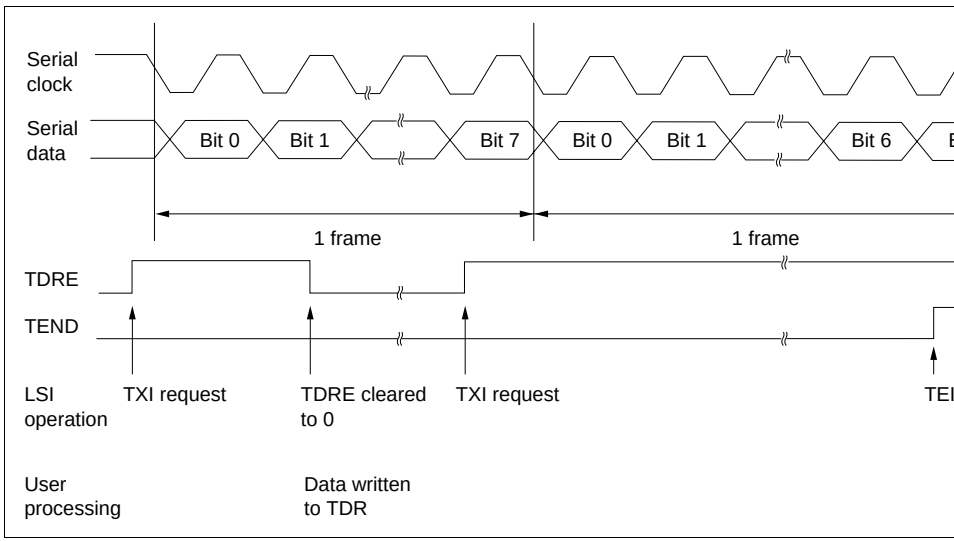
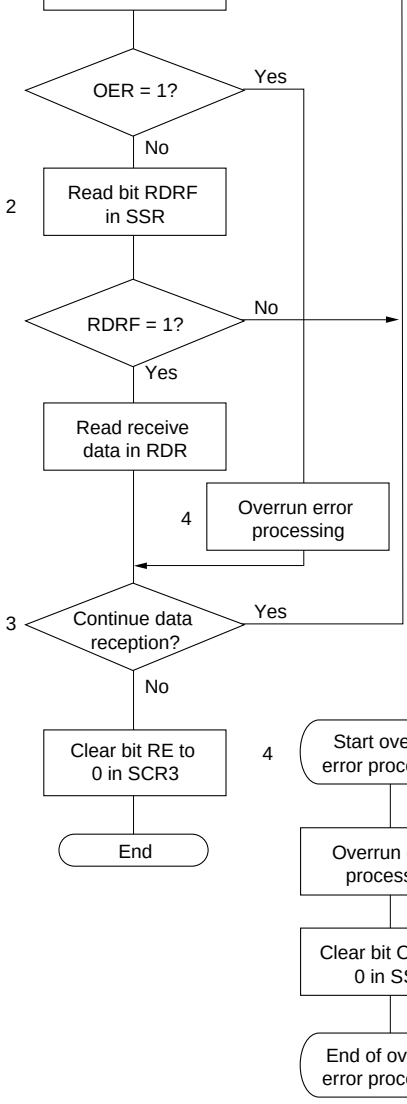


Figure 10.12 Example of Operation when Transmitting in Synchronous M



(SSR) to determine if there is an error. If an overrun error has occurred, execute overrun error processing.

2. Read SSR and check that bit RDRF is set to 1. If it is, read the receive data in RDR. When the RDR data is read, bit RDRF is cleared to 0 automatically.

3. When continuing data reception, finish reading of bit RDRF and RDR before receiving the MSB (bit 7) of the current frame. When the data in RDR is read, bit RDRF is cleared to 0 automatically.

4. If an overrun error has occurred, read bit OER in SSR, and after carrying out the necessary error processing, clear bit OE to 0. Reception cannot be resumed if bit OER is set to 1.

Figure 10.13 Example of Data Reception Flowchart (Synchronous Mo

If this check shows that there is no overrun error, bit RDRF is set to 1, and the receive d
stored in RDR. If bit RIE is set to 1 in SCR3, an RXI interrupt is requested. If the chec
identifies an overrun error, bit OER is set to 1.

Bit RDRF remains set to 1. If bit RIE is set to 1 in SCR3, an ERI interrupt is requested

See table 10.12 for the conditions for detecting a receive error, and receive data process

Note: No further receive operations are possible while a receive error flag is set. Bits
FER, PER, and RDRF must therefore be cleared to 0 before resuming reception

Figure 10.14 shows an example of the operation when receiving in synchronous mode.

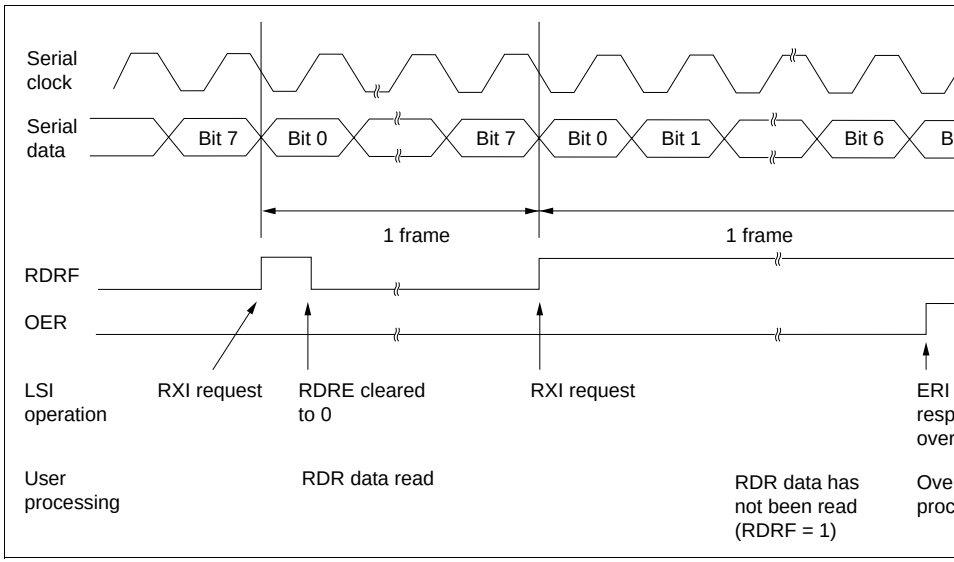
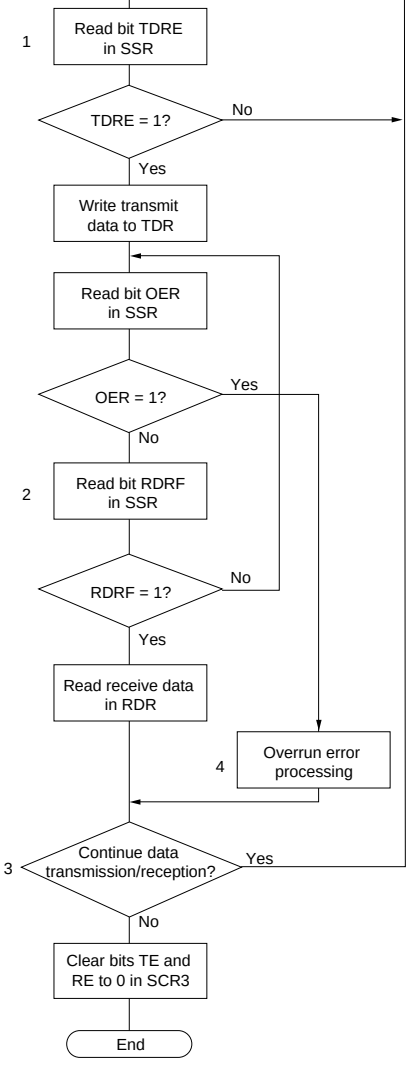


Figure 10.14 Example of Operation when Receiving in Synchronous Mo



1. Read the serial status register (SSR) and check that bit TDRE is set to 1, then write transmit data to the transmit data register (TDR). When data is written to TDR, bit TDRE is cleared to 0 automatically.

2. Read SSR and check that bit RDRF is set to 1. If it is, read the receive data in RDR. When the RDR data is read, bit RDRF is cleared to 0 automatically.

3. When continuing data transmission/reception, finish reading of bit RDRF and RDR before receiving the MSB (bit 7) of the current frame. Before receiving the MSB (bit 7) of the current frame, also read TDRE = 1 to confirm that a write can be performed, then write data to TDR. When data is written to TDR, bit TDRE is cleared to 0 automatically, and when the data in RDR is read, bit RDRF is cleared to 0 automatically.

4. If an overrun error has occurred, read bit OER in SSR, and after carrying out the necessary error processing, clear bit OER to 0. Transmission and reception cannot be resumed if bit OER is set to 1.
   See figure 10-13 for details on overrun error processing.

Figure 10.15 Example of Simultaneous Data Transmission/Reception Flow (Synchronous Mode)

The multiprocessor communication function enables data to be exchanged among a nur processors on a shared communication line. Serial data communication is performed in asynchronous mode using the multiprocessor format (in which a multiprocessor bit is a transfer data).

In multiprocessor communication, each receiver is assigned its own ID code. The seria communication cycle consists of two cycles, an ID transmission cycle in which the rece specified, and a data transmission cycle in which the transfer data is sent to the specifie These two cycles are differentiated by means of the multiprocessor bit, 1 indicating an I transmission cycle, and 0, a data transmission cycle.

The sender first sends transfer data with a 1 multiprocessor bit added to the ID code of it wants to communicate with, and then sends transfer data with a 0 multiprocessor bit a transmit data. When a receiver receives transfer data with the multiprocessor bit set to compares the ID code with its own ID code, and if they are the same, receives the trans sent next. If the ID codes do not match, it skips the transfer data until data with the mu bit set to 1 is sent again.

In this way, a number of processors can exchange data among themselves.

Figure 10.16 shows an example of communication between processors using the multip format.

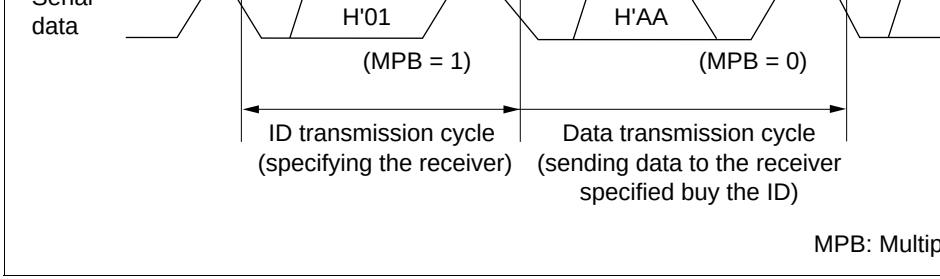


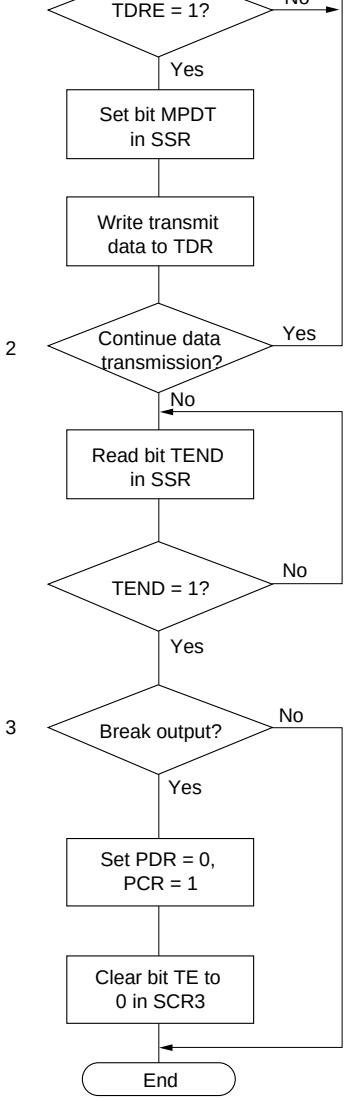
Figure 10.16 Example of Inter-Processor Communication Using Multiprocesso (Sending data H'AA to receiver A)

There is a choice of four data transfer formats. If a multiprocessor format is specified, bit specification is invalid. See table 10.11 for details.

For details on the clock used in multiprocessor communication, see 10.1.4, Operation Synchronous Mode.

- Multiprocessor transmitting

Figure 10.17 shows an example of a flowchart for multiprocessor data transmission. T procedure should be followed for multiprocessor data transmission after initializing SC



TDR, bit TDRE is cleared to 0 automatically.

2. When continuing data transmission, be sure to read TDRE = 1 to confirm that a write can be performed before writing data to TDR. When data is written to TDR, bit TDRE is cleared to 0 automatically.

3. If a break is to be output when data transmission ends, set the port PCR to 1 and clear the port PDR to 0, then clear bit TE in SCR3 to 0.

Figure 10.17 Example of Multiprocessor Data Transmission Flowchart

bit TDRE is set to 1 bit TEND in SSR bit is set to 1, the mark state, in which 1s are tra
established after the stop bit has been sent. If bit TEIE in SCR3 is set to 1 at this time
request is made.

Figure 10.18 shows an example of the operation when transmitting using the multipro
format.

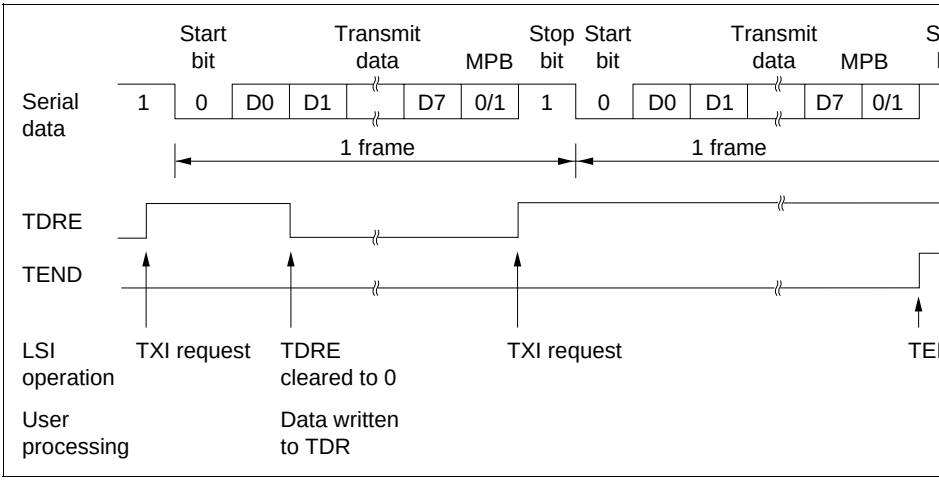
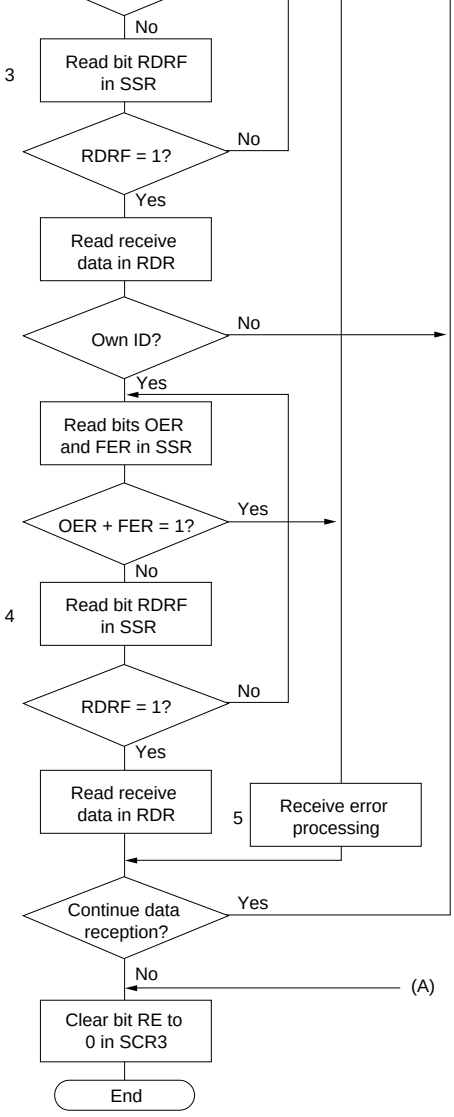


**Figure 10.18 Example of Operation when Transmitting using Multiprocesso
(8-bit data, multiprocessor bit, 1 stop bit)**

- Multiprocessor receiving

Figure 10.19 shows an example of a flowchart for multiprocessor data reception. This
should be followed for multiprocessor data reception after initializing SCI3.



own ID. If the ID is not this receiver's, set bit MPIE to 1 again. When the RDR data is read, bit RDRF is cleared to 0 automatically.

4. Read SSR and check that bit RDRF is set to 1, then read the data in RDR.

5. If a receive error has occurred, read bits OER and FER in SSR to identify the error and after carrying out the necessary error processing, ensure that bits OER and FER are both cleared to 0. Reception cannot be resumed if either of these bits is set to 1. In the case of a framing error, a break can be detected by reading the value of the RXD_{3x} pin.

Figure 10.19 Example of Multiprocessor Data Reception Flowchart

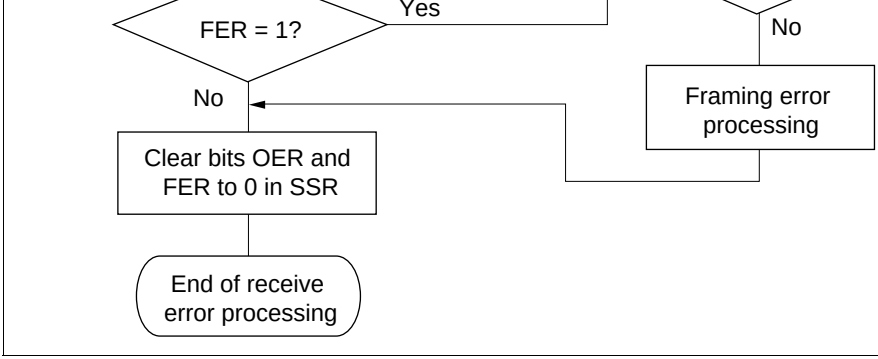
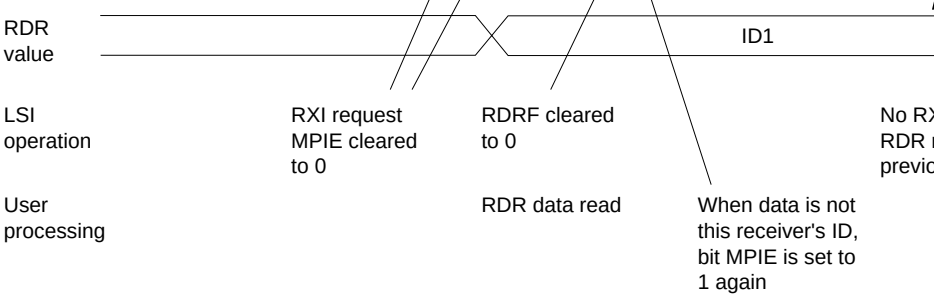
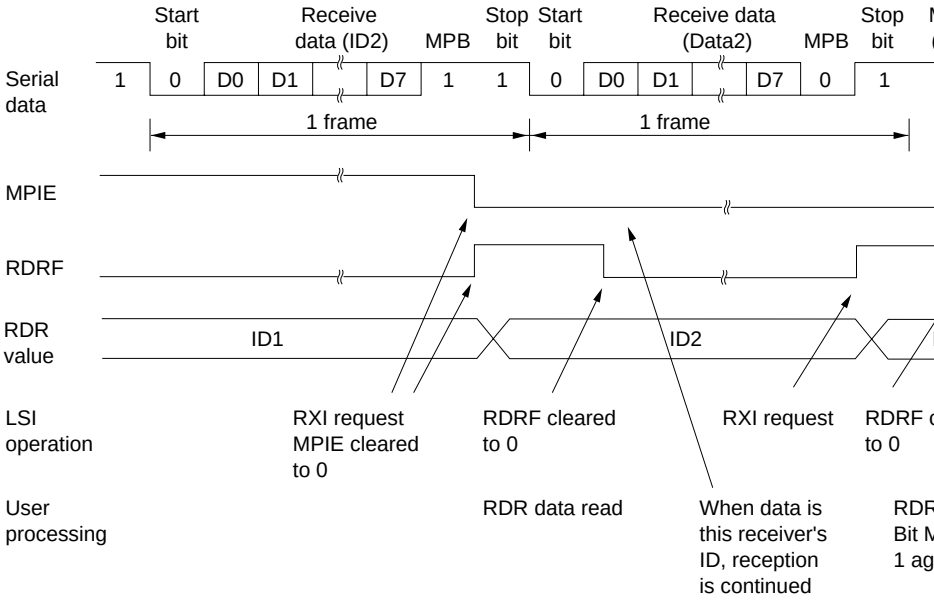


Figure 10.19 Example of Multiprocessor Data Reception Flowchart (co

Figure 10.20 shows an example of the operation when receiving using the multiproces



(a) When data does not match this receiver's ID



(b) When data matches this receiver's ID

Figure 10.20 Example of Operation when Receiving using Multiprocessor Fo (8-bit data, multiprocessor bit, 1 stop bit)

Interrupt Abbreviation	Interrupt Request	Ve Ad
RXI	Interrupt request initiated by receive data full flag (RDRF)	H'0
TXI	Interrupt request initiated by transmit data empty flag (TDRE)	
TEI	Interrupt request initiated by transmit end flag (TEND)	
ERI	Interrupt request initiated by receive error flag (OER, FER, PER)	

Each interrupt request can be enabled or disabled by means of bits TIE and RIE in SC

When bit TDRE is set to 1 in SSR, a TXI interrupt is requested. When bit TEND is se SSR, a TEI interrupt is requested. These two interrupts are generated during transmiss

The initial value of bit TDRE in SSR is 1. Therefore, if the transmit data empty interr (TXI) is enabled by setting bit TIE to 1 in SCR3 before transmit data is transferred to interrupt will be requested even if the transmit data is not ready.

Also, the initial value of bit TEND in SSR is 1. Therefore, if the transmit end interrup (TEI) is enabled by setting bit TEIE to 1 in SCR3 before transmit data is transferred to interrupt will be requested even if the transmit data has not been sent.

Effective use of these interrupt requests can be made by having processing that transfe data to TDR carried out in the interrupt service routine.

To prevent the generation of these interrupt requests (TXI and TEI), on the other hand bits for these interrupt requests (bits TIE and TEIE) should be set to 1 after transmit da transferred to TDR.

When bit RDRF is set to 1 in SSR, an RXI interrupt is requested, and if any of bits OE FER is set to 1, an ERI interrupt is requested. These two interrupt requests are genera reception.

For further details, see 3.3, Interrupts.

Data can be written to TDR irrespective of the state of bit TDRE, but if new data is wri
TDR while bit TDRE is cleared to 0, the data previously stored in TDR will be lost of i
yet been transferred to TSR. Accordingly, to ensure that serial transmission is perform
dependably, you should first check that bit TDRE is set to 1, then write the transmit dat
once only (not two or more times).

2. Operation when a number of receive errors occur simultaneously

If a number of receive errors are detected simultaneously, the status flags in SSR will b
states shown in table 10.14. If an overrun error is detected, data transfer from RSR to R
not be performed, and the receive data will be lost.

Table 10.14 SSR Status Flag States and Receive Data Transfer

SSR Status Flags				Receive Data Transfer	
RDRF*	**OER**	**FER**	**PER**	**RSR → RDR**	**Receive Error Status**
1	1	0	0	X	Overrun error
0	0	1	0	O	Framing error
0	0	0	1	O	Parity error
1	1	1	0	X	Overrun error + framing error
1	1	0	1	X	Overrun error + parity error
0	0	1	1	O	Framing error + parity error
1	1	1	1	X	Overrun error + framing error +

O : Receive data is transferred from RSR to RDR.

X : Receive data is not transferred from RSR to RDR.

Note: * Bit RDRF retains its state prior to data reception. However, note that if RDR is r
overrun error has occurred in a frame because reading of the receive data in the
frame was delayed, RDRF will be cleared to 0.

When bit TE is cleared to 0, the TXD_{3x} pin functions as an I/O port whose input/outpu and level are determined by PDR and PCR. This fact can be used to set the TXD_{3x} pi mark state, or to detect a break during transmission.

To keep the communication line in the mark state (1 state) until bit TE is set to 1, set P PDR = 1. Since bit TE is cleared to 0 at this time, the TXD_{3x} pin functions as an I/O p output.

To detect a break, clear bit TE to 0 after setting PCR = 1 and PDR = 0.

When bit TE is cleared to 0, the transmission unit is initialized regardless of the curren transmission state, the TXD_{3x} pin functions as an I/O port, and 0 is output from the TX

5. Receive error flags and transmit operation (synchronous mode only)

When a receive error flag (OER, PER, or FER) is set to 1, transmission cannot be start bit TDRE is cleared to 0. The receive error flags must be cleared to 0 before starting t

Note also that receive error flags cannot be cleared to 0 even if bit RE is cleared to 0.

6. Receive data sampling timing and receive margin in asynchronous mode

In asynchronous mode, SCI3 operates on a basic clock with a frequency 16 times the t When receiving, SCI3 performs internal synchronization by sampling the falling edge bit with the basic clock. Receive data is latched internally at the 8th rising edge of the This is illustrated in figure 10.21.

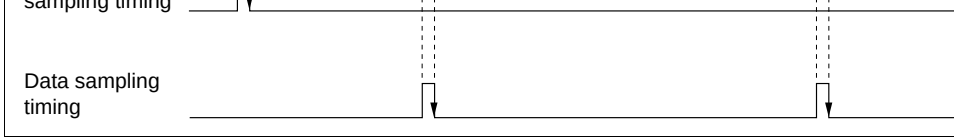


Figure 10.21 Receive Data Sampling Timing in Asynchronous Mode

Consequently, the receive margin in asynchronous mode can be expressed as shown in (1).

$$M = \{(0.5 - \frac{1}{2N}) - \frac{D - 0.5}{N} - (L - 0.5)\,F\} \times 100\ [\%] \quad \text{..... Equation (1)}$$

where

M: Receive margin (%)
N: Ratio of bit rate to clock (N = 16)
D: Clock duty (D = 0.5 to 1.0)
L: Frame length (L = 9 to 12)
F: Absolute value of clock frequency deviation

Substituting 0 for F (absolute value of clock frequency deviation) and 0.5 for D (clock d equation (1), a receive margin of 46.875% is given by equation (2).

When D = 0.5 and F = 0,
$M = \{0.5 — 1/(2 \times 16)\} \times 100\ [\%]$
$= 46.875\%$ ..... Equation (2)

However, this is only a computed value, and a margin of 20% to 30% should be allowe carrying out system design.

0, if the read operation coincides with completion of reception of a frame, the next fra
may be read. This is illustrated in figure 10.22.

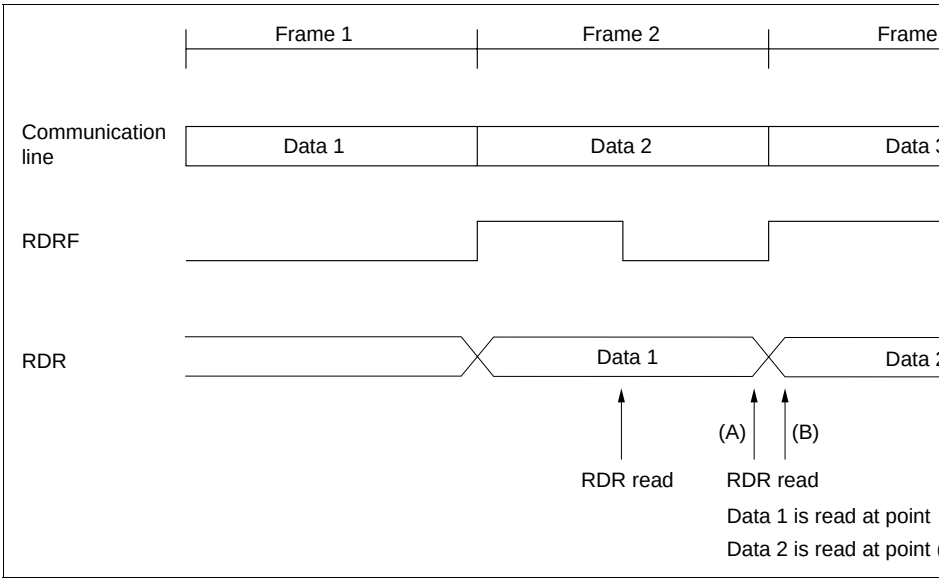


Figure 10.22 Relation between RDR Read Timing and Data

In this case, only a single RDR read operation (not two or more) should be performed checking that bit RDRF is set to 1. If two or more reads are performed, the data read t should be transferred to RAM, etc., and the RAM contents used. Also, ensure that the sufficient margin in an RDR read operation before reception of the next frame is comp be precise in terms of timing, the RDR read should be completed before bit 7 is transf synchronous mode, or before the STOP bit is transferred in asynchronous mode.

8. Transmit and receive operations when making a state transition

Make sure that transmit and receive operations have completely finished before carryi transition processing.

When stopping data transfer, issue one instruction to clear bits TE and RE to 0 and to s
CKE1 and CKE0 in SCR3 to 1 and 0, respectively. In this case, bit COM in SMR shou
The above prevents SCK_3 from being used as a general input/output pin. To avoid an i
level of voltage from being applied to SCK_3, the line connected to SCK_3 should be pull
the V_{CC} level via a resistor, or supplied with output from an external device.

b. When an SCK_3 function is switched from clock output to general input/output

When stopping data transfer,

(i) Issue one instruction to clear bits TE and RE to 0 and to set bits CKE1 and CKE0 i
1 and 0, respectively.

(ii) Clear bit COM in SMR to 0

(iii) Clear bits CKE1 and CKE0 in SCR3 to 0

Note that special care is also needed here to avoid an intermediate level of voltage from
applied to SCK_3.

10. Set up at subactive or subsleep mode

At subactive or subsleep mode, SCI3 becomes possible use only at CPU clock is øw/2.

Features of the 14-bit PWM are as follows.

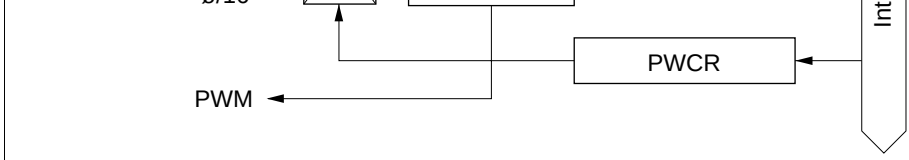
- Choice of two conversion periods

Any of the following four conversion periods can be chosen:
131,072/ø, with a minimum modulation width of 8/ø (PWCR1 = 1, PWCR0 = 1)
65,536/ø, with a minimum modulation width of 4/ø (PWCR1 = 1, PWCR0 = 0)
32,768/ø, with a minimum modulation width of 2/ø (PWCR1 = 0, PWCR0 = 1)
16,384/ø, with a minimum modulation width of 1/ø (PWCR1 = 0, PWCR0 = 0)

- Pulse division method for less ripple
- Use of module standby mode enables this module to be placed in standby mode inc
  when not used.

11.1.2 Block Diagram

Figure 11.1 shows a block diagram of the 14-bit PWM.



Notation:
PWDRL: PWM data register L
PWDRU: PWM data register U
PWCR: PWM control register

Figure 11.1 Block Diagram of the 14 bit PWM

11.1.3 Pin Configuration

Table 11.1 shows the output pin assigned to the 14-bit PWM.

Table 11.1 Pin Configuration

Name	Abbrev.	I/O	Function
PWM output pin	PWM	Output	Pulse-division PWM waveform

11.1.4 Register Configuration

Table 11.2 shows the register configuration of the 14-bit PWM.

Table 11.2 Register Configuration

Name	Abbrev.	R/W	Initial Value
PWM control register	PWCR	W	H'FC
PWM data register U	PWDRU	W	H'C0
PWM data register L	PWDRL	W	H'00
Clock stop register 2	CKSTPR2	R/W	H'FF

PWCR is an 8-bit write-only register for input clock selection.

Upon reset, PWCR is initialized to H'FC.

Bits 7 to 2: Reserved bits

Bits 7 to 2 are reserved; they are always read as 1, and cannot be modified.

Bits 1 and 0: Clock select 1 and 0 (PWCR1, PWCR0)

Bits 1 and 0 select the clock supplied to the 14-bit PWM. These bits are write-only bi
always read as 1.

Bit 1 PWCR1	Bit 0 PWCR0	Description
0	0	The input clock is ø/2 (tø* = 2/ø) The conversion period is 16,384/ø, with a minimum modulation width of 1/ø
0	1	The input clock is ø/4 (tø* = 4/ø) The conversion period is 32,768/ø, with a minimum modulation width of 2/ø
1	0	The input clock is ø/8 (tø* = 8/ø) The conversion period is 65,536/ø, with a minimum modulation width of 4/ø
1	1	The input clock is ø/16 (tø* = 16/ø) The conversion period is 131,072/ø, with a minimum modulation width of 8/ø

Note: * Period of PWM input clock.

PWDRL

Bit	7	6	5	4	3	2	1
	PWDRL7	PWDRL6	PWDRL5	PWDRL4	PWDRL3	PWDRL2	PWDRL1
Initial value	0	0	0	0	0	0	0
Read/Write	W	W	W	W	W	W	W

PWDRU and PWDRL form a 14-bit write-only register, with the upper 6 bits assigned and the lower 8 bits to PWDRL. The value written to PWDRU and PWDRL gives the level width of one PWM waveform cycle.

When 14-bit data is written to PWDRU and PWDRL, the register contents are latched i waveform generator, updating the PWM waveform generation data. The 14-bit data sh always be written in the following sequence:

1. Write the lower 8 bits to PWDRL.
2. Write the upper 6 bits to PWDRU.

PWDRU and PWDRL are write-only registers. If they are read, all bits are read as 1.

Upon reset, PWDRU and PWDRL are initialized to H'C000.

11.2.3 Clock Stop Register 2 (CKSTPR2)

Bit	7	6	5	4	3	2	1
	—	—	—	—	AECKSTP	WDCKSTP	PWCKSTP
Initial value	1	1	1	1	1	1	1
Read/Write	—	—	—	—	R/W	R/W	R/W

CKSTPR2 is an 8-bit read/write register that performs module standby mode control fo modules. Only the bit relating to the PWM is described here. For details of the other b sections on the relevant modules.

2. Set bits PWCR1 and PWCR0 in the PWM control register (PWCR) to select a conv
period of 131,072/ø (PWCR1 = 1, PWCR0 = 1), 65,536/ø (PWCR1 = 1, PWCR0 =
32,768/ø (PWCR1 = 0, PWCR0 = 1), or 16,384/ø (PWCR1 = 0, PWCR0 = 0).
3. Set the output waveform data in PWM data registers U and L (PWDRU/L). Be sure
the correct sequence, first PWDRL then PWDRU. When data is written to PWDRU
in these registers will be latched in the PWM waveform generator, updating the PW
waveform generation in synchronization with internal signals.

   One conversion period consists of 64 pulses, as shown in figure 11.2. The total of t
level pulse widths during this period (T_H) corresponds to the data in PWDRU and P
This relation can be represented as follows.

$$T_H = (\text{data value in PWDRU and PWDRL} + 64) \times t_{ø}/2$$

where tø is the PWM input clock period: 2/ø (PWCR = H'0), 4/ø (PWCR = H'1), 8/ø (P
H'2), or 16/ø (PWCR = H'3).

Example: Settings in order to obtain a conversion period of 32,768 µs:
When PWCR1 = 0 and PWCR0 = 0, the conversion period is 16,384/ø, so ø
0.5 MHz. In this case, tfn = 512 µs, with 1/ø (resolution) = 2.0 µs.
When PWCR1 = 0 and PWCR0 = 1, the conversion period is 32,768/ø, so ø
MHz. In this case, tfn = 512 µs, with 2/ø (resolution) = 2.0 µs.
When PWCR1 = 1 and PWCR0 = 0, the conversion period is 65,536/ø , so ø
MHz. In this case, tfn = 512 µs, with 4/ø (resolution) = 2.0 µs.
Accordingly, for a conversion period of 32,768 µs, the system clock frequen
must be 0.5 MHz, 1 MHz, or 2 MHz.

$t_H = t_{H1} + t_{H2} + t_{H3} + \cdots + t_{H64}$

$t_{f1} = t_{f2} = t_{f3} \quad \cdots\cdots \quad = t_{f84}$

Figure 11.2 PWM Output Waveform

11.3.2 PWM Operation Modes

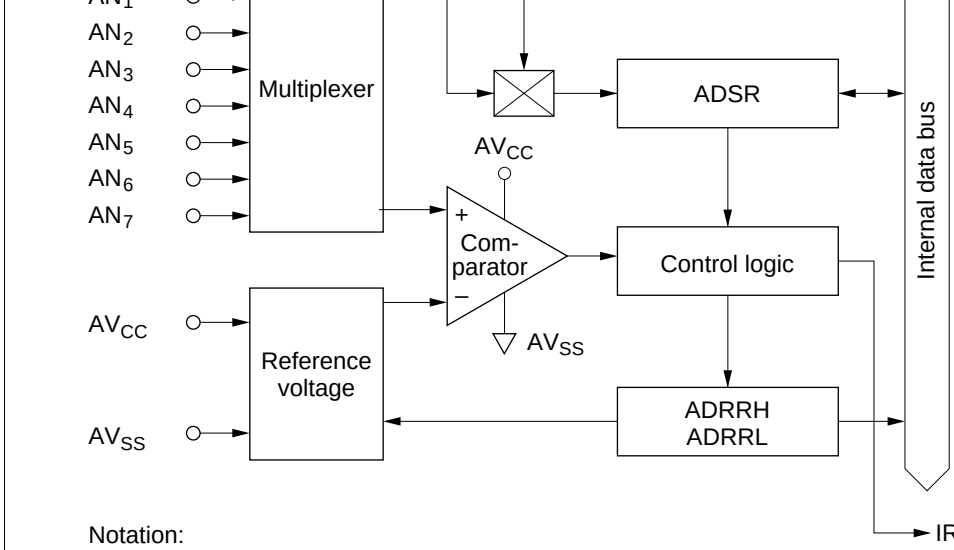
PWM operation modes are shown in table 11.3.

Table 11.3 PWM Operation Modes

Operation Mode	Reset	Active	Sleep	Watch	Subactive	Subsleep	Standby
PWCR	Reset	Functions	Functions	Held	Held	Held	Held
PWDRU	Reset	Functions	Functions	Held	Held	Held	Held
PWDRL	Reset	Functions	Functions	Held	Held	Held	Held

The A/D converter has the following features.

- 10-bit resolution
- Eight input channels
- Conversion time: approx. 15.5 µs per channel (at 2 MHz operation)
- Built-in sample-and-hold function
- Interrupt requested on completion of A/D conversion
- A/D conversion can be started by external trigger input
- Use of module standby mode enables this module to be placed in standby mode inc
  when not used.



Notation:
AMR: A/D mode register
ADSR: A/D start register
ADRR: A/D result register
IRRAD: A/D conversion end interrupt request flag

Figure 12.1 Block Diagram of the A/D Converter

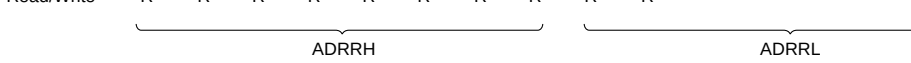
Analog input 0	AN0	Input	Analog input channel 0
Analog input 1	AN1	Input	Analog input channel 1
Analog input 2	AN2	Input	Analog input channel 2
Analog input 3	AN3	Input	Analog input channel 3
Analog input 4	AN4	Input	Analog input channel 4
Analog input 5	AN5	Input	Analog input channel 5
Analog input 6	AN6	Input	Analog input channel 6
Analog input 7	AN7	Input	Analog input channel 7
External trigger input	ADTRG	Input	External trigger input for starting A/D co

12.1.4 Register Configuration

Table 12.2 shows the A/D converter register configuration.

Table 12.2 Register Configuration

Name	Abbrev.	R/W	Initial Value
A/D mode register	AMR	R/W	H'30
A/D start register	ADSR	R/W	H'7F
A/D result register H	ADRRH	R	Not fixed
A/D result register L	ADRRL	R	Not fixed
Clock stop register 1	CKSTPRT1	R/W	H'FF



ADRRH and ADRRL together comprise a 16-bit read-only register for holding the resu
analog-to-digital conversion. The upper 8 bits of the data are held in ADRRH, and the
bits in ADRRL.

ADRRH and ADRRL can be read by the CPU at any time, but the ADRRH and ADRR
during A/D conversion are not fixed. After A/D conversion is complete, the conversion
stored as 10-bit data, and this data is held until the next conversion operation starts.

ADRRH and ADRRL are not cleared on reset.

12.2.2 A/D Mode Register (AMR)

Bit	7	6	5	4	3	2	1
	CKS	TRGE	—	—	CH3	CH2	CH1
Initial value	0	0	1	1	0	0	0
Read/Write	R/W	R/W	—	—	R/W	R/W	R/W

AMR is an 8-bit read/write register for specifying the A/D conversion speed, external t
option, and the analog input pins.

Upon reset, AMR is initialized to H'30.

of at least 15.5 µs.

Bit 6: External trigger select (TRGE)

Bit 6 enables or disables the start of A/D conversion by external trigger input.

Bit 6 TRGE	Description
0	Disables start of A/D conversion by external trigger (
1	Enables start of A/D conversion by rising or falling edge of external trigge $\overline{\text{ADTRG}}$*

Note: * The external trigger ($\overline{\text{ADTRG}}$) edge is selected by bit INTEG4 of IEGR. See 1. edge select register (IEGR) in 3.3.2 for details.

Bits 5 and 4: Reserved bits

Bits 5 and 4 are reserved; they are always read as 1, and cannot be modified.

0	1	0	0	AN0
0	1	0	1	AN1
0	1	1	0	AN2
0	1	1	1	AN3
1	0	0	0	AN4
1	0	0	1	AN5
1	0	1	0	AN6
1	0	1	1	AN7

Note: * Don't care

12.2.3 A/D Start Register (ADSR)

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

The A/D start register (ADSR) is an 8-bit read/write register for starting and stopping A conversion.

A/D conversion is started by writing 1 to the A/D start flag (ADSF) or by input of the d edge of the external trigger signal, which also sets ADSF to 1. When conversion is con converted data is set in ADRRH and ADRRL, and at the same time ADSF is cleared to

	Write: Starts A/D conversion

Bits 6 to 0: Reserved bits

Bits 6 to 0 are reserved; they are always read as 1, and cannot be modified.

12.2.4 Clock Stop Register 1 (CKSTPR1)

Bit	7	6	5	4	3	2	1
	—	S31CKSTP	S32CKSTP	ADCKSTP	TGCKSTP	TFCKSTP	TCCKST
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

CKSTPR1 is an 8-bit read/write register that performs module standby mode control f
modules. Only the bit relating to the A/D converter is described here. For details of t
see the sections on the relevant modules.

Bit 4: A/D converter module standby mode control (ADCKSTP)

Bit 4 controls setting and clearing of module standby mode for the A/D converter.

ADCKSTP	Description
0	A/D converter is set to module standby mode
1	A/D converter module standby mode is cleared

value of 1 during A/D conversion, and is cleared to 0 automatically when conversion is

The completion of conversion also sets bit IRRAD in interrupt request register 2 (IRR2
A/D conversion end interrupt is requested if bit IENAD in interrupt enable register 2 (I
set to 1.

If the conversion time or input channel needs to be changed in the A/D mode register (A
during A/D conversion, bit ADSF should first be cleared to 0, stopping the conversion
in order to avoid malfunction.

12.3.2 Start of A/D Conversion by External Trigger Input

The A/D converter can be made to start A/D conversion by input of an external trigger
External trigger input is enabled at pin $\overline{\text{ADTRG}}$ when bit IRQ4 in PMR1 is set to 1 and
in AMR is set to 1. Then when the input signal edge designated in bit IEG4 of interrup
select register (IEGR) is detected at pin $\overline{\text{ADTRG}}$, bit ADSF in ADSR will be set to 1, s
conversion.

Figure 12.2 shows the timing.

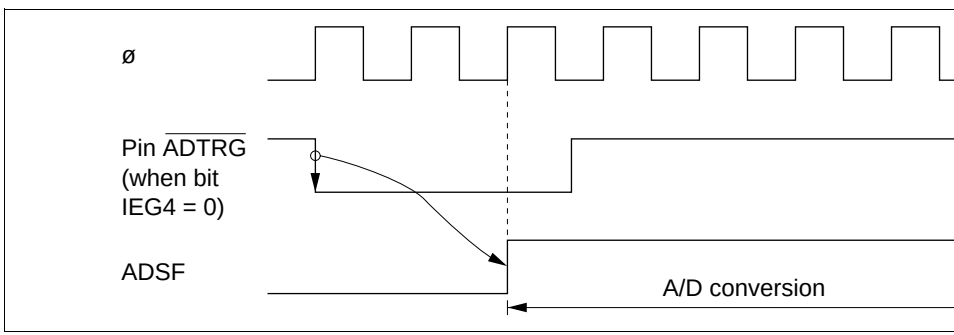


Figure 12.2 External Trigger Input Timing

ADSR	Reset	Functions	Functions	Held	Held	Held	Held
ADRRH	Held*	Functions	Functions	Held	Held	Held	Held
ADRRL	Held*	Functions	Functions	Held	Held	Held	Held

Note: * Undefined in a power-on reset.

12.4 Interrupts

When A/D conversion ends (ADSF changes from 1 to 0), bit IRRAD in interrupt requ
(IRR2) is set to 1.

A/D conversion end interrupts can be enabled or disabled by means of bit IENAD in i
enable register 2 (IENR2).

For further details see 3.3, Interrupts.

12.5 Typical Use

An example of how the A/D converter can be used is given below, using channel 1 (pi
the analog input channel. Figure 12.3 shows the operation timing.

1. Bits CH3 to CH0 of the A/D mode register (AMR) are set to 0101, making pin AN
   input channel. A/D interrupts are enabled by setting bit IENAD to 1, and A/D con
   started by setting bit ADSF to 1.
2. When A/D conversion is complete, bit IRRAD is set to 1, and the A/D conversion
   stored is stored in ADRRH and ADRRL. At the same time ADSF is cleared to 0, a
   converter goes to the idle state.
3. Bit IENAD = 1, so an A/D conversion end interrupt is requested.
4. The A/D interrupt handling routine starts.
5. The A/D conversion result is read and processed.
6. The A/D interrupt handling routine ends.

If ADSF is set to 1 again afterward, A/D conversion starts and steps 2 through 6 take p

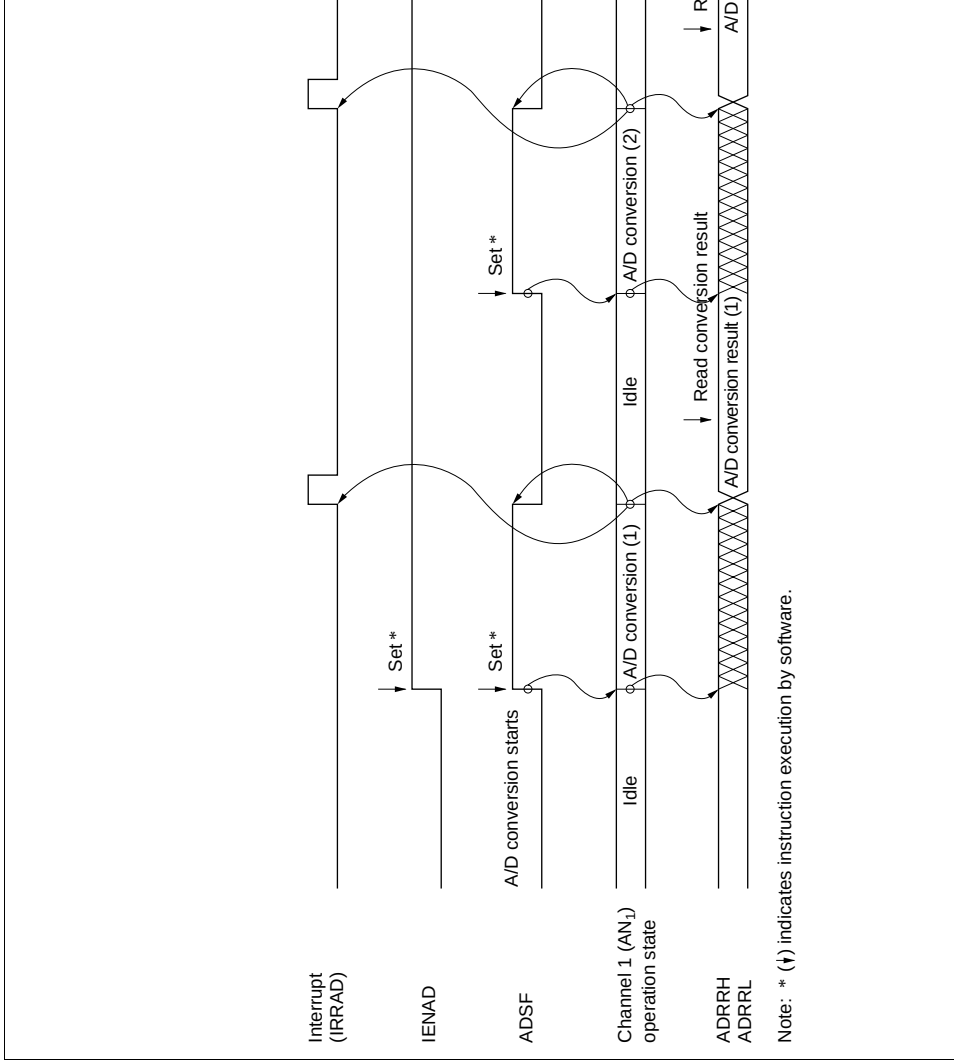


Figure 12.3 Typical A/D Converter Operation Timing

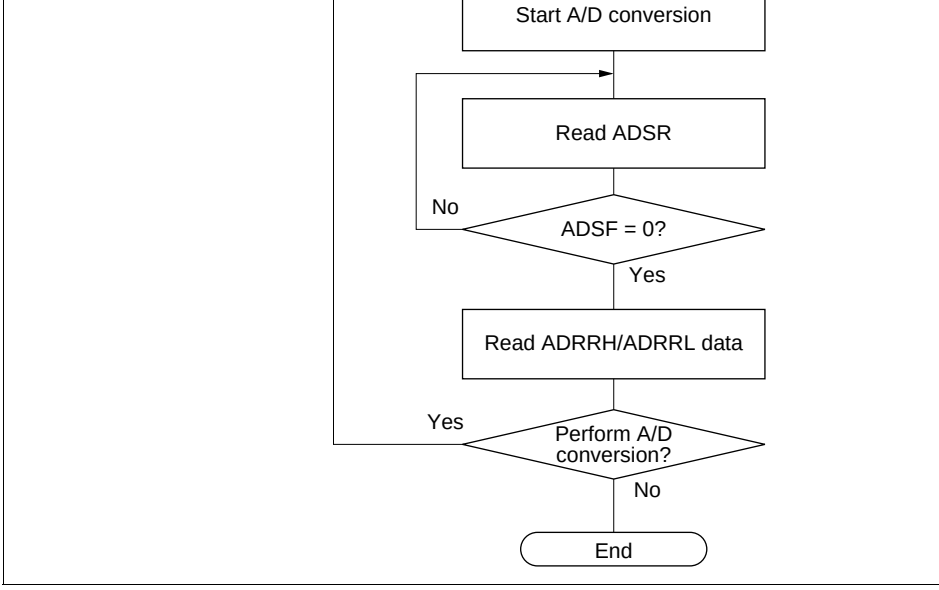


Figure 12.4 Flow Chart of Procedure for Using A/D Converter (Polling by S

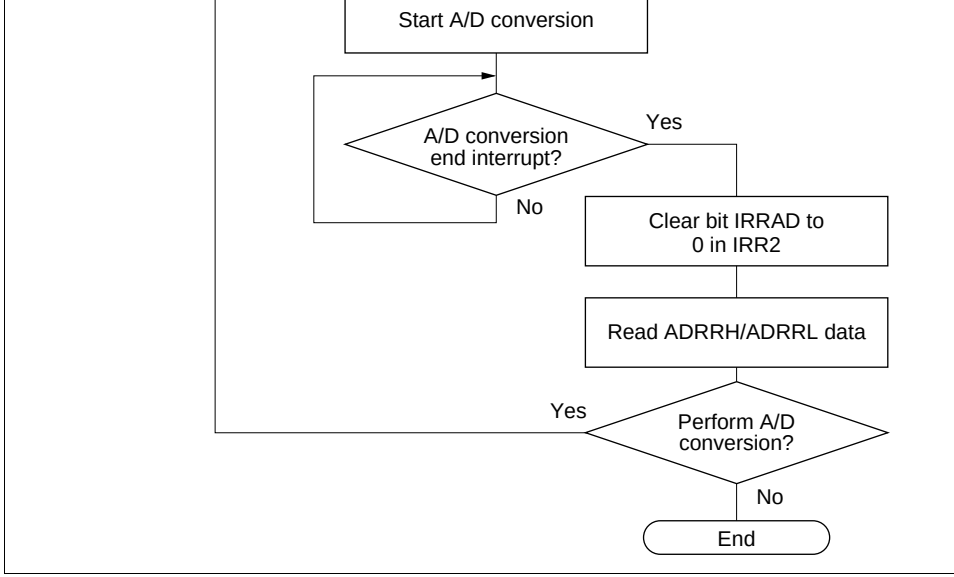


Figure 12.5 Flow Chart of Procedure for Using A/D Converter (Interrupts

12.6 Application Notes

- Data in ADRRH and ADRRL should be read only when the A/D start flag (ADSF) start register (ADSR) is cleared to 0.
- Changing the digital input signal at an adjacent pin during A/D conversion may adv affect conversion accuracy.
- When A/D conversion is started after clearing module standby mode, wait for 10 ø c cycles before starting.

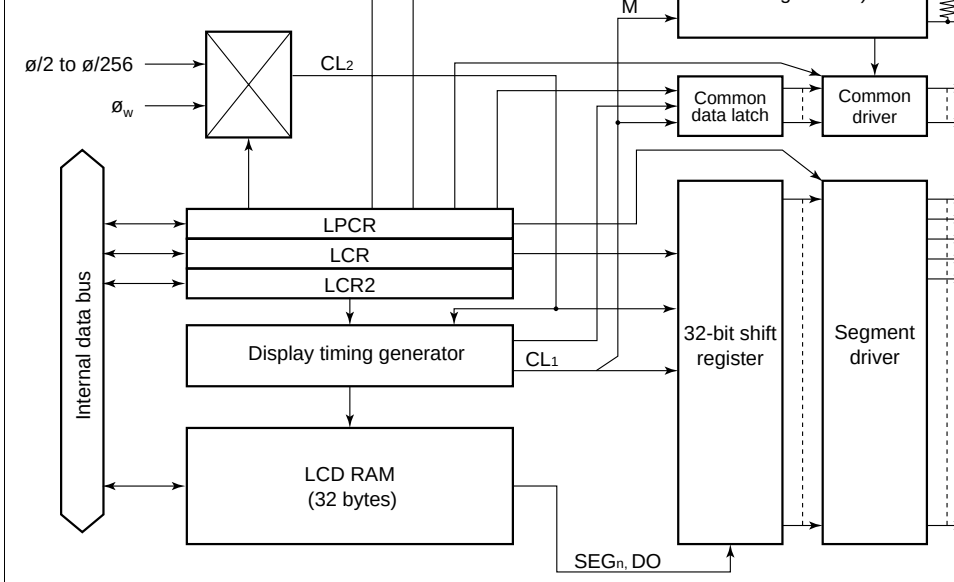
1. Features

Features of the LCD controller/driver are given below.

- Display capacity

Duty Cycle	Internal Driver	Segment External Expansion
Static	32 seg	256 seg
1/2	32 seg	128 seg
1/3	32 seg	64 seg
1/4	32 seg	64 seg

- LCD RAM capacity
  8 bits × 32 bytes (256 bits)
- Word access to LCD RAM
- All eight segment output pins can be used individually as port pins.
- Common output pins not used because of the duty cycle can be used for common
  buffering (parallel connection).
- Display possible in operating modes other than standby mode
- Choice of 11 frame frequencies
- Built-in power supply split-resistance, supplying LCD drive power
- Use of module standby mode enables this module to be placed in standby mode in
  when not used.
- Built-in step-up constant-voltage (5 V) power supply allows LCD display even at l
  (H8/3867 Series)
- A or B waveform selectable by software



Notation:
LPCR: LCD port control register
LCR: LCD control register
LCR2: LCD control register 2

Figure 13.1 Block Diagram of LCD Controller/Driver

			(setting programmable)
Common output pins	COM_4 to COM_1	Output	LCD common drive pins Pins can be used in parallel wi 1/2 duty
Segment external expansion signal pins	CL_1	Output	Display data latch clock, multip SEG_{32}
	CL_2	Output	Display data shift clock, multip SEG_{31}
	M	Output	LCD alternation signal, multipl SEG_{29}
	DO	Output	Serial display data, multiplexe
LCD power supply pins	V_0, V_1, V_2, V_3	—	Used when a bypass capacito connected externally, and whe external power supply circuit is

13.1.4 Register Configuration

Table 13.2 shows the register configuration of the LCD controller/driver.

Table 13.2 LCD Controller/Driver Registers

Name	Abbrev.	R/W	Initial Value	Add
LCD port control register	LPCR	R/W	H'00	H'FF
LCD control register	LCR	R/W	H'80	H'FF
LCD control register 2	LCR2	R/W	H'60	H'FF
LCD RAM	—	R/W	Undefined	H'F7
Clock stop register 2	CKSTPR2	R/W	H'FF	H'FF

LPCR is an 8-bit read/write register which selects the duty cycle and LCD driver pin fu

LPCR is initialized to H'00 upon reset.

Bits 7 to 5: Duty cycle select 1 and 0 (DTS1, DTS0), common function select (CMX)

The combination of DTS1 and DTS0 selects static, 1/2, 1/3, or 1/4 duty. CMX specifie
or not the same waveform is to be output from multiple pins to increase the common dr
when not all common pins are used because of the duty setting.

Bit 7 DTS1	Bit 6 DTS0	Bit 5 CMX	Duty Cycle	Common Drivers	Notes
0	0	0	Static	COM_1 (initial value)	Do not use COM_4, COM COM_2.
		1		COM_4 to COM_1	COM_4, COM_3, and COM the same waveform as C
0	1	0	1/2 duty	COM_2 to COM_1	Do not use COM_4 and C
		1		COM_4 to COM_1	COM_4 outputs the same as COM_3, and COM_2 ou same waveform as COM
1	0	0	1/3 duty	COM_3 to COM_1	Do not use COM_4.
		1		COM_4 to COM_1	Do not use COM_4.
1	1	0	1/4 duty	COM_4 to COM_1	—
		1			

Note: * These pins function as ports when the setting of SGS3 to SGS0 is 0000 or 000

Bit 3: Segment driver select 3 to 0 (SGS3 to SGS0)

Bits 3 to 0 select the segment drivers to be used.

					Function of Pins SEG_{32} to SEG_1			
Bit 4 SGX	**Bit 3 SGS3**	**Bit 2 SGS2**	**Bit 1 SGS1**	**Bit 0 SGS0**	**SEG_{32} to SEG_{25}**	**SEG_{24} to SEG_{17}**	**SEG_{16} to SEG_9**	**SEG_8 to SEG_1**
0	0	0	0	0	Port	Port	Port	Port
	0	0	0	1	Port	Port	Port	Port
	0	0	1	*	SEG	Port	Port	Port
	0	1	0	*	SEG	SEG	Port	Port
	0	1	1	*	SEG	SEG	SEG	Port
	1	*	*	*	SEG	SEG	SEG	SEG
1	0	0	0	0	Port*	Port	Port	Port
	*	*	*	*	Setting prohibited			

Note: * SEG_{32} to SEG_{29} are external expansion pins.

display data control, and selects the frame frequency.

LCR is initialized to H'80 upon reset.

Bit 7: Reserved bit

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

Bit 6: LCD drive power supply on/off control (PSW)

Bit 6 can be used to turn the LCD drive power supply off when LCD display is not requ
power-down mode, or when an external power supply is used. When the ACT bit is cle
or in standby mode, the LCD drive power supply is turned off regardless of the setting

Bit 6 PSW	Description	
0	LCD drive power supply off	(in
1	LCD drive power supply on	

Bit 5: Display function activate (ACT)

Bit 5 specifies whether or not the LCD controller/driver is used. Clearing this bit to 0 h
operation of the LCD controller/driver. The LCD drive power supply is also turned off
of the setting of the PSW bit. However, register contents are retained.

Bit 5 ACT	Description	
0	LCD controller/driver operation halted	(in
1	LCD controller/driver operates	

Bits 3 to 0: Frame frequency select 3 to 0 (CKS3 to CKS0)

Bits 3 to 0 select the operating clock and the frame frequency. In subactive mode, wa and subsleep mode, the system clock (ø) is halted, and therefore display operations are performed if one of the clocks from ø/2 to ø/256 is selected. If LCD display is require modes, øw, øw/2, or øw/4 must be selected as the operating clock.

Bit 3	Bit 2	Bit 1	Bit 0		Frame Frequen	
CKS3	CKS2	CKS1	CKS0	Operating Clock	ø = 2 MHz	ø = 25
0	*	0	0	øw	128 Hz[*3] (initial value)	
0	*	0	1	øw/2	64 Hz[*3]	
0	*	1	*	øw/4	32 Hz[*3]	
1	0	0	0	ø/2	—	244 H
1	0	0	1	ø/4	977 Hz	122 H
1	0	1	0	ø/8	488 Hz	61 Hz
1	0	1	1	ø/16	244 Hz	30.5 H
1	1	0	0	ø/32	122 Hz	—
1	1	0	1	ø/64	61 Hz	—
1	1	1	0	ø/128	30.5 Hz	—
1	1	1	1	ø/256	—	—

Notes: 1. This is the frame frequency in active (medium-speed, øosc/16) mode when
2. When 1/3 duty is selected, the frame frequency is 4/3 times the value show
3. This is the frame frequency when øw = 32.768 kHz.

waveform, selects the drive power supply, controls the step-up constant-voltage (5 V) p
supply, and selects the duty cycle of the charge/discharge pulses which control disconn
the power supply split-resistance from the power supply circuit.

LCR2 is initialized to H'60 upon reset.

Bit 7: A waveform/B waveform switching control (LCDAB)

Bit 7 specifies whether the A waveform or B waveform is used as the LCD drive wavef

Bit 7 LCDAB	Description
0	Drive using A waveform (i
1	Drive using B waveform

Bits 6 and 5: Reserved bits

Bits 6 and 5 are reserved; they are always read as 1 and cannot be modified.

Bit 4: Drive power supply select, step-up constant-voltage (5 V) power supply control
(Applies to the H8/3867 Series only)

When V_{CC} is selected as the drive power supply, the step-up constant-voltage (5 V) po
simultaneously stops operating; when 5 V is selected as the drive power supply, the ste
constant-voltage (5 V) power supply simultaneously operates.

Bit 4 SUPS	Description
0	Drive power supply is V_{CC}, step-up constant-voltage (5 V) power supply h (I
1	Drive power supply is 5 V, step-up constant-voltage (5 V) power supply op

0	1	0	0	4/8	
0	1	0	1	5/8	
0	1	1	0	6/8	
0	1	1	1	0	Fixed low
1	0	*	*	1/16	
1	1	*	*	1/32	

Bits 3 to 0 select the duty cycle while the power supply split-resistance is connected to
supply circuit.

When a 0 duty cycle is selected, the power supply split-resistance is permanently disc
from the power supply circuit, so power should be supplied to pins V1, V2, and V3 by
circuit.

Figure 13.2 shows the waveform of the charge/discharge pulses. The duty cycle is Tc

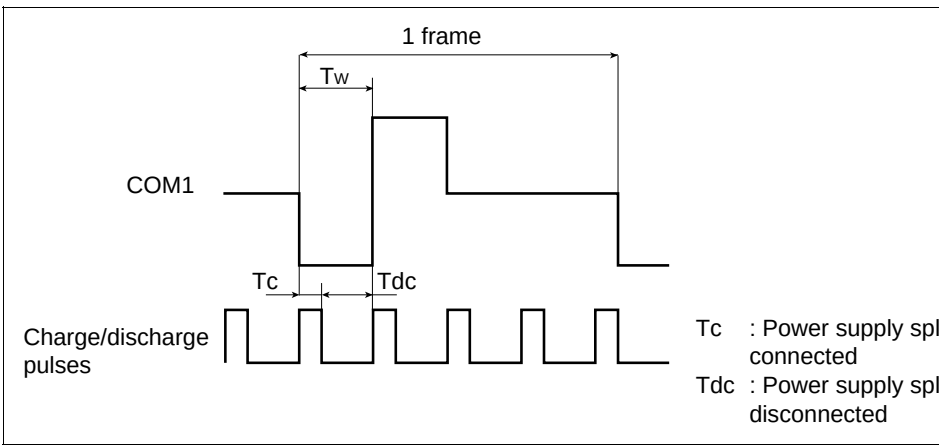


Figure 13.2 Example of A Waveform with 1/2 Duty and 1/2 Bias

modules. Only the bit relating to the LCD controller/driver is described here. For deta
other bits, see the sections on the relevant modules.

Bit 0: LCD controller/driver module standby mode control (LDCKSTP)

Bit 0 controls setting and clearing of module standby mode for the LCD controller/driv

Bit 0 LDCKSTP	Description	
0	LCD controller/driver is set to module standby mode	
1	LCD controller/driver module standby mode is cleared	(i

a. Using 1/2 duty

When 1/2 duty is used, interconnect pins V_2 and V_3 as shown in figure 13.3.

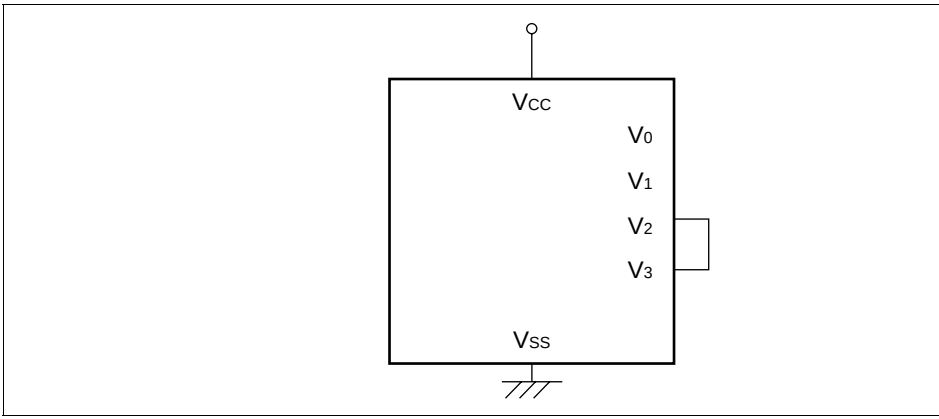


Figure 13.3 Handling of LCD Drive Power Supply when Using 1/2 Du

b. Large-panel display

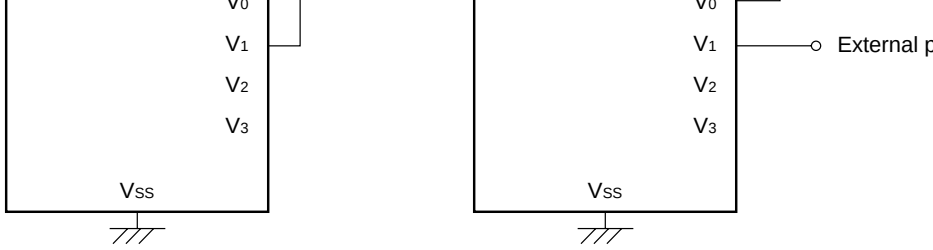
As the impedance of the built-in power supply split-resistance is large, it may n suitable for driving a large panel. If the display lacks sharpness when using a l refer to section 13.3.7, Boosting the LCD Drive Power Supply. When static or selected, the common output drive capability can be increased. Set CMX to 1 selecting the duty cycle. In this mode, with a static duty cycle pins COM_4 to C the same waveform, and with 1/2 duty the COM_1 waveform is output from pin COM_1, and the COM_2 waveform is output from pins COM_4 and COM_3.

c. Luminance adjustment function (V_0 pin)

Connecting a resistance between the V_0 and V_1 pins enables the luminance to l For details, see 13.3.3, Luminance Adjustment Function (V_0 Pin).

d. LCD drive power supply setting

With the H8/3867 Series, there are two ways of providing LCD power: by usin chip power supply circuit, or by using an external circuit. With the H8/3867 S chip power supply circuit allows the selection of either the power supply voltag a step-up constant voltage (5 V). For details of the step-up constant-voltage po see 13.3.4, Step-Up Constant-Voltage (5 V) Power Supply.



(a) Using on-chip power supply circuit (b) Using external power supply circuit

Figure 13.4 Examples of LCD Power Supply Pin Connections

e. Low-power-consumption LCD drive system

Use of a low-power-consumption LCD drive system enables the power consump required for LCD drive to be optimized. For details, see 13.3.5, Low-Power-Co LCD Drive System.

f. Segment external expansion

The number of segments can be increased by connecting an HD66100 externally details, see section 13.3.8, Connection to HD66100.

should be selected in accordance with the LCD panel specification. For the clo
method in watch mode, subactive mode, and subsleep mode, see 13.3.6, Opera
Power-Down Modes.

d. A or B waveform selection

   Either the A or B waveform can be selected as the LCD waveform to be used b
   LCDAB.

e. LCD drive power supply selection

   When the on-chip power supply circuit is used, the power supply to be used ca
   with the SUPS bit. When an external power supply circuit is used, select V_{CC}
   SUPS bit and turn the LCD drive power supply off with the PSW bit.

turned on. Word- or byte-access instructions can be used for RAM setting.

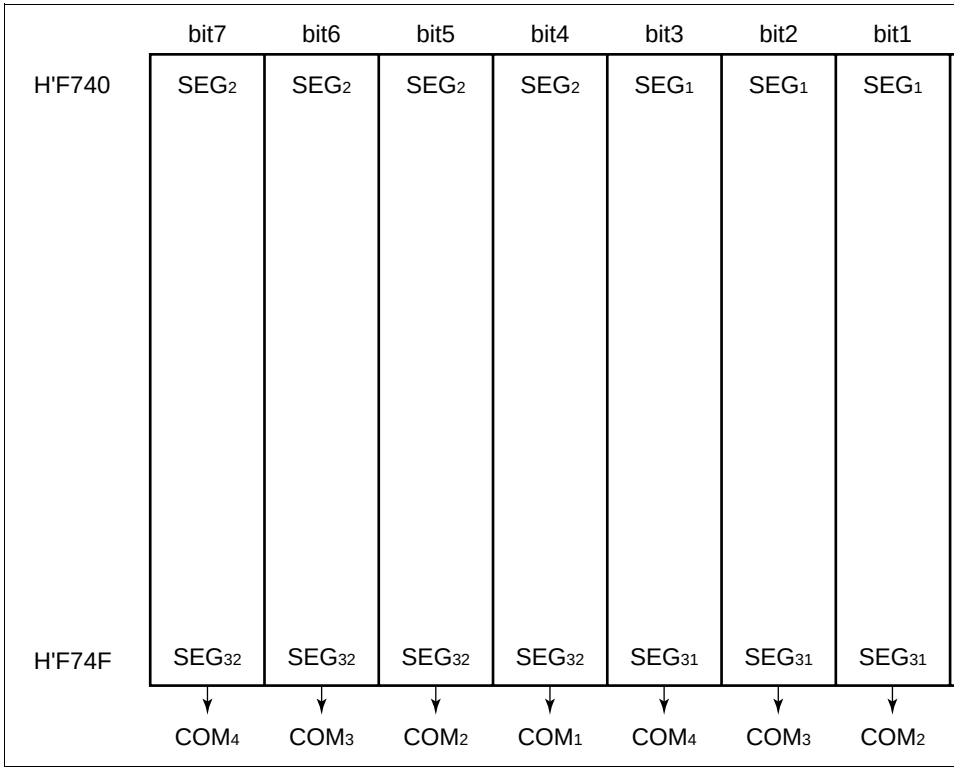


Figure 13.5 LCD RAM Map when Not Using Segment External Expansion (1/

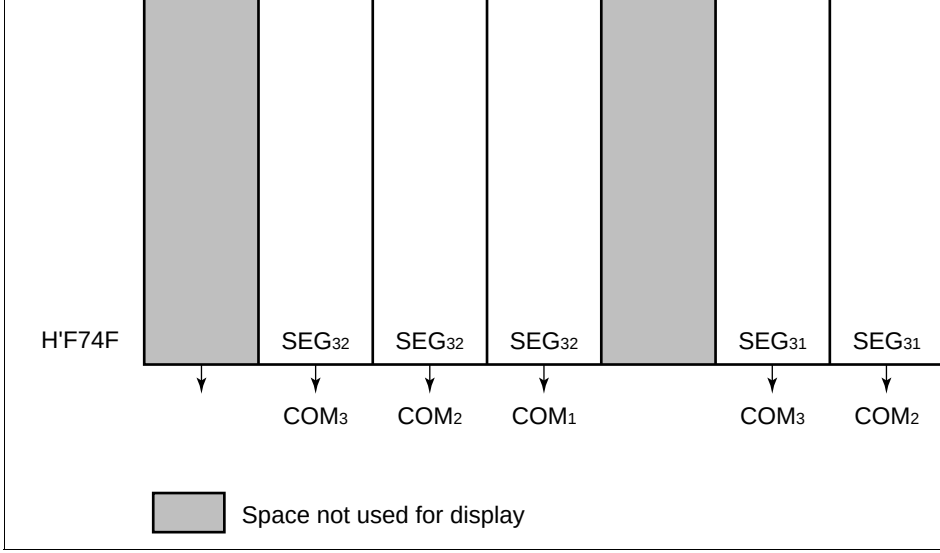


Figure 13.6 LCD RAM Map when Not Using Segment External Expansion (1

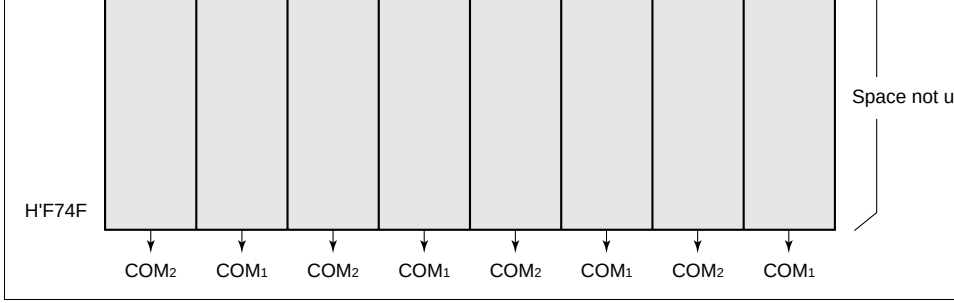


Figure 13.7 LCD RAM Map when Not Using Segment External Expansion (1/

Figure 13.8 LCD RAM Map when Not Using Segment External Expansion (Sta

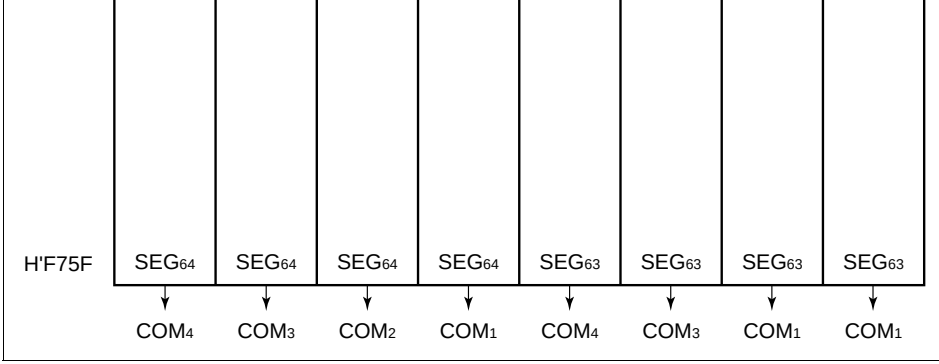


Figure 13.9 LCD RAM Map when Using Segment External Expansio
(SGX = "1", SGS3 to SGS0 = "0000" 1/4 Duty)

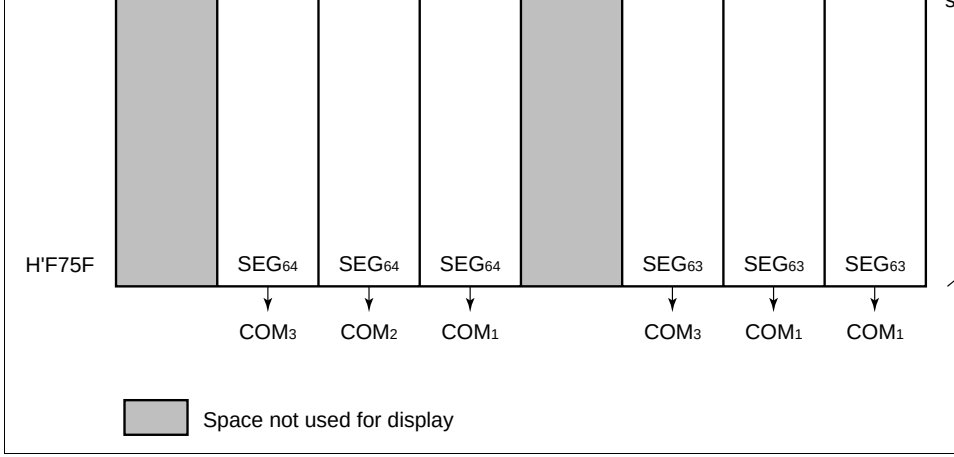


Figure 13.10 LCD RAM Map when Using Segment External Expansion (SGX = "1", SGS3 to SGS0 = "0000" 1/3 Duty)

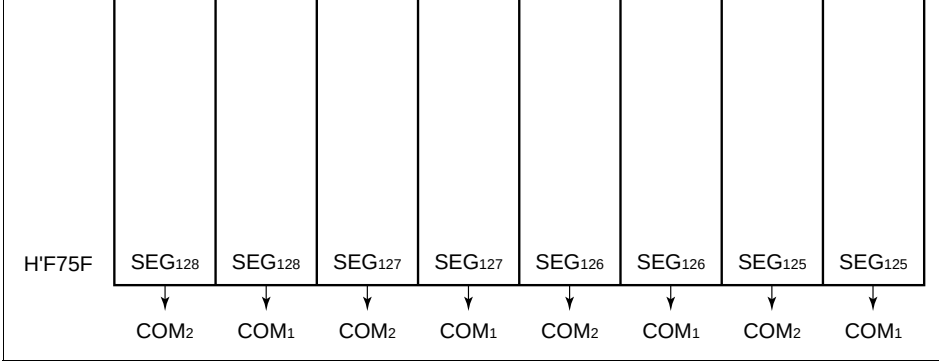


Figure 13.11 LCD RAM Map when Using Segment External Expansio
(SGX = "1", SGS3 to SGS0 = "0000" 1/2 Duty)

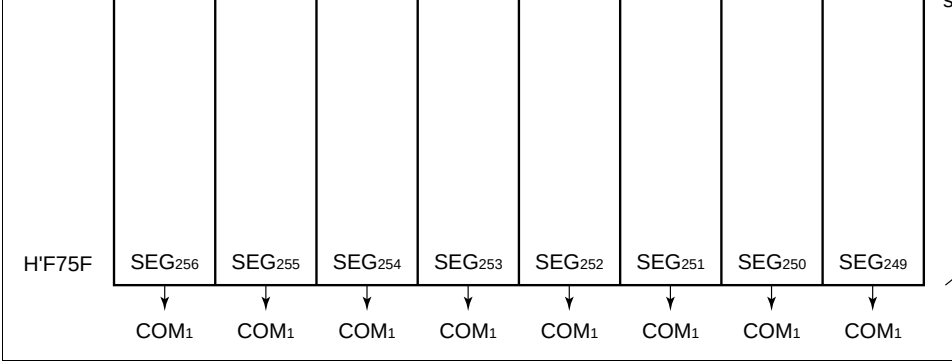


Figure 13.12 LCD RAM Map when Using Segment External Expansion (SGX = “1”, SGS3 to SGS0 = “0000” Static)

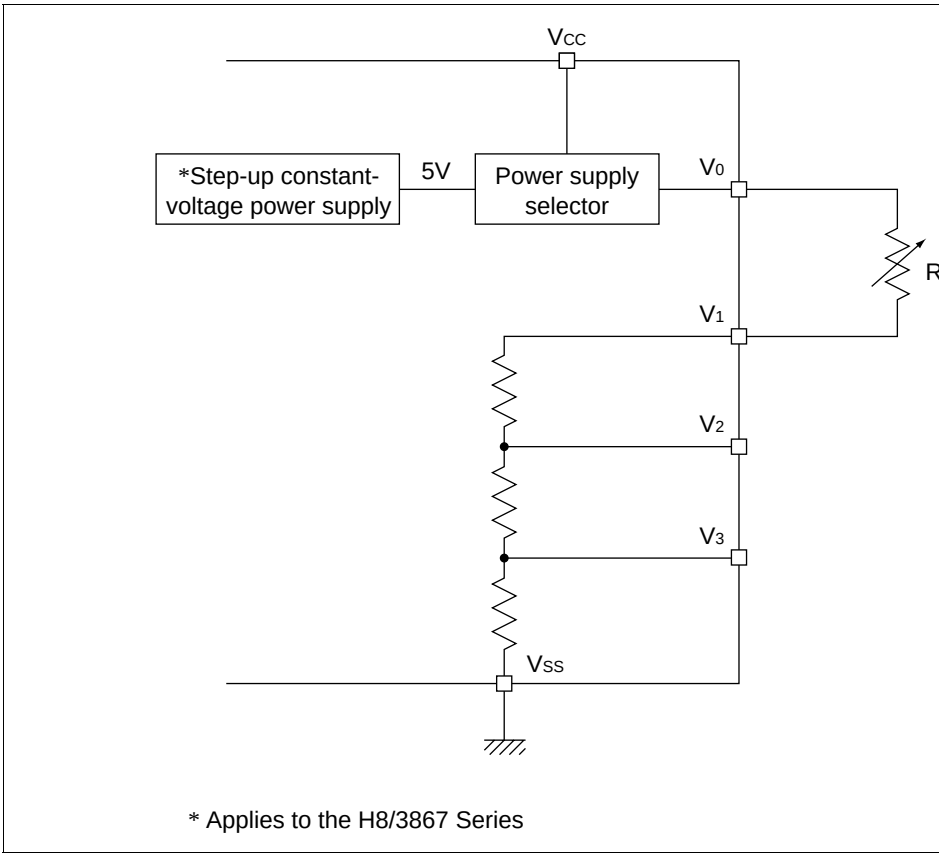


Figure 13.13 LCD Drive Power Supply Unit

Note: The step-up constant-voltage (5 V) power supply must not be used for any purp
than the H8/3864 Series' LCD drive power supply. When a large panel is drive
power supply capacity may be insufficient. In this case, either use V_{CC} as the p
supply or use an external power supply circuit.

13.3.5 Low-Power-Consumption LCD Drive System

The use of the built-in split-resistance is normally the easiest method for implementing
power supply circuit, but since the built-in resistance is fixed, a certain direct current fl
constantly from the built-in resistance's V_{CC} to V_{SS}. As this current does not depend o
current dissipation of the LCD panel, if an LCD panel with a small current dissipation i
wasteful amount of power will be consumed. The H8/3864 Series is equipped with a fu
minimize this waste of power. Use of this function makes it possible to achieve the opt
power supply circuit for the LCD panel's current dissipation.

1. Principles
   a. Capacitors are connected as external circuits to LCD power supply pins V1, V2,
   shown in figure 13.14.
   b. The capacitors connected to V1, V2, and V3 are repeatedly charged and dischar
   cycle shown in figure 13.14, maintaining the potentials.
   c. At this time, the charged potential is a potential corresponding to the V1, V2, an
   respectively. (For example, with 1/3 bias drive, the charge for V2 is 2/3 that of V
   for V3 is 1/3 that of V1.)
   d. Power is supplied to the LCD panel by means of the charges accumulated in the
   capacitors.
   e. The capacitances and charging/discharging periods of these capacitors are theref
   determined by the current dissipation of the LCD panel.
   f. The charging and discharging periods can be selected by software.

affect the driving of the LCD panel.

d. In this way, the capacitors connected to V1, V2, and V3 are repeatedly charge discharged in the cycle shown in figure 13.14, maintaining the potentials and c driving the LCD panel.

e. As can be seen from the above description, the capacitances and charging/discl periods of the capacitors are determined by the current dissipation of the LCD The charging/discharging periods can be selected with bits CDS3 to CDS0.

f. The actual capacitor capacitances and charging/discharging periods must be de experimentally in accordance with the current dissipation requirements of the I An optimum current value can be selected, in contrast to the case in which a di flows constantly in the built-in split-resistance.

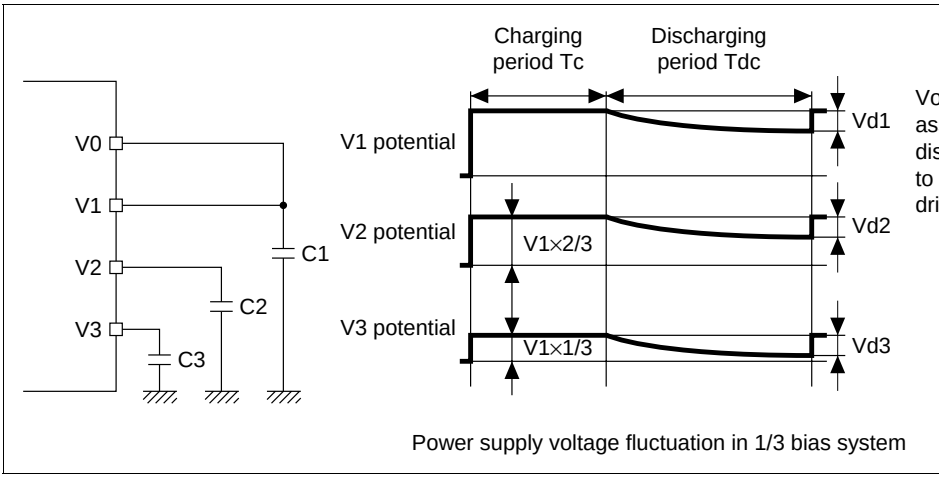


Figure 13.14 Example of Low-Power-Consumption LCD Drive Operat

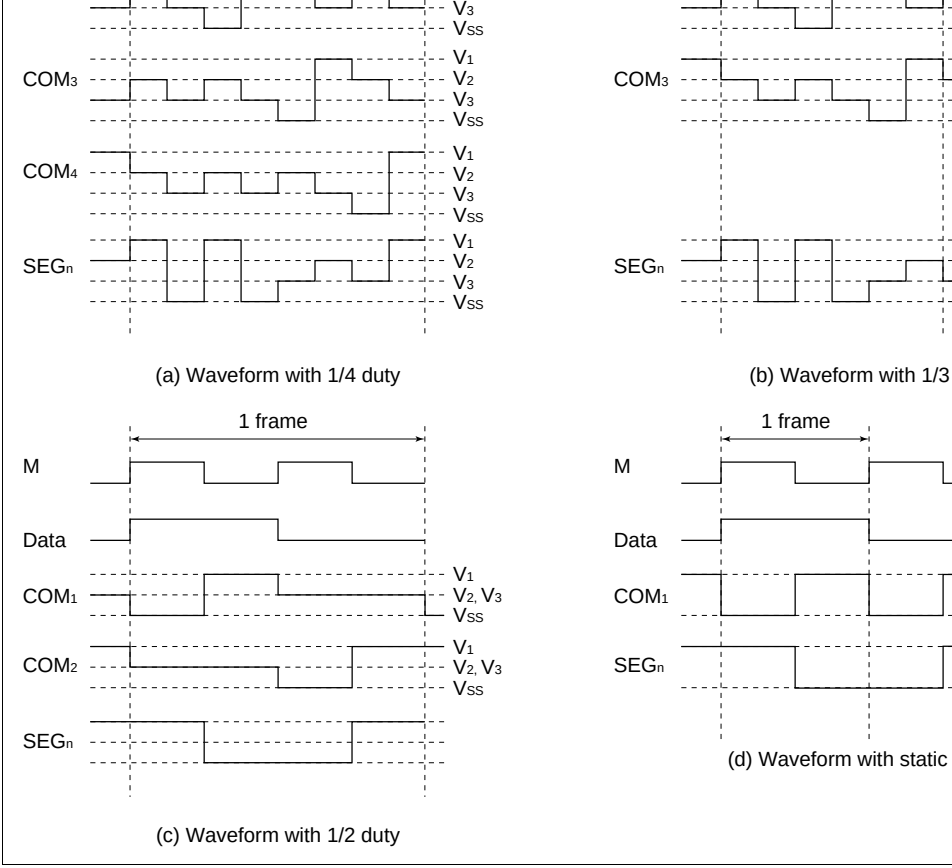


Figure 13.15 Output Waveforms for Each Duty Cycle (A Waveform)

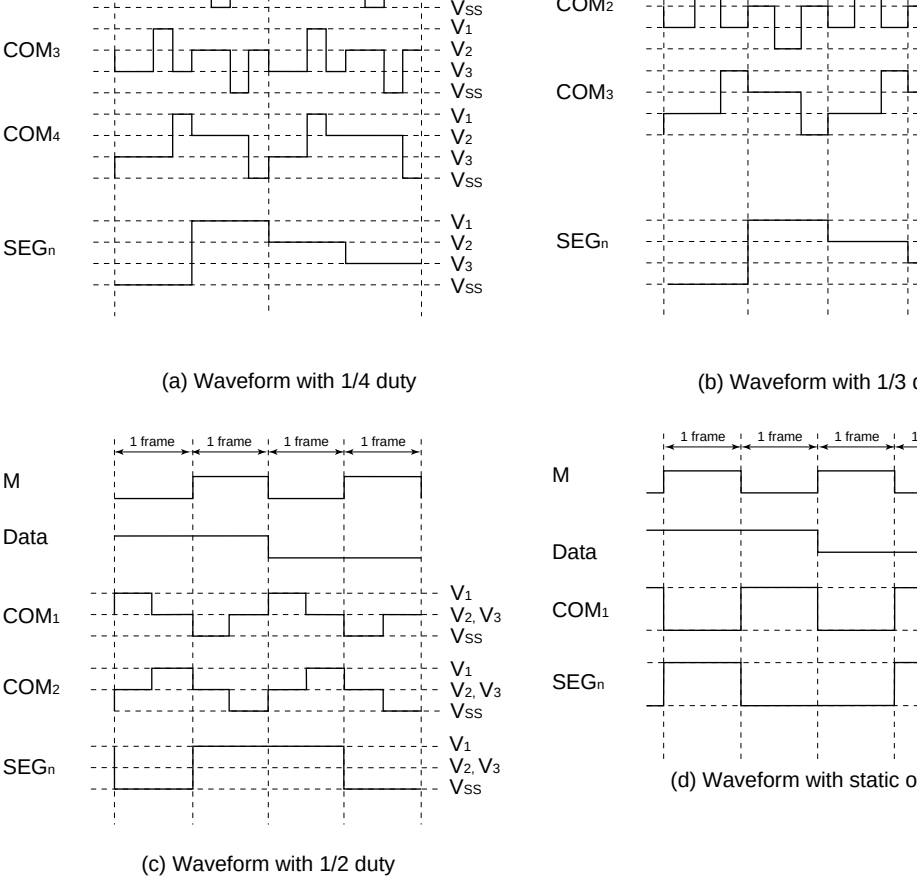


Figure 13.16 Output Waveforms for Each Duty Cycle (B Waveform)

1/3 duty	Common output	V_3	V_2	V_1	V_S
	Segment output	V_2	V_3	V_{SS}	V_1
1/4 duty	Common output	V_3	V_2	V_1	V_S
	Segment output	V_2	V_3	V_{SS}	V_1

13.3.6 Operation in Power-Down Modes

In the H8/3867 Series, the LCD controller/driver can be operated even in the power-do
The operating state of the LCD controller/driver in the power-down modes is summariz
13.4.

In subactive mode, watch mode, and subsleep mode, the system clock oscillator stops, a
therefore, unless øw, øw/2, or øw/4 has been selected by bits CKS3 to CKS0, the clock
supplied and display will halt. Since there is a possibility that a direct current will be ap
the LCD panel in this case, it is essential to ensure that øw, øw/2, or øw/4 is selected. I
(medium-speed) mode, the system clock is switched, and therefore CKS3 to CKS0 mus
modified to ensure that the frame frequency does not change.

Table 13.4 Power-Down Modes and Display Operation

Mode		Reset	Active	Sleep	Watch	Sub-active	Sub-sleep	Standb
Clock	ø	Runs	Runs	Runs	Stops	Stops	Stops	Stops
	øw	Runs	Runs	Runs	Runs	Runs	Runs	Stops[*1]
Display	ACT = “0”	Stops	Stops	Stops	Stops	Stops	Stops	Stops[*2]
operation	ACT = “1”	Stops	Functions	Functions	Functions[*3]	Functions[*3]	Functions[*3]	Stops[*2]

Notes: 1. The subclock oscillator does not stop, but clock supply is halted.
2. The LCD drive power supply is turned off regardless of the setting of the PSW
3. Display operation is performed only if øw, øw/2, or øw/4 is selected as the op
clock.
4. The clock supplied to the LCD stops.

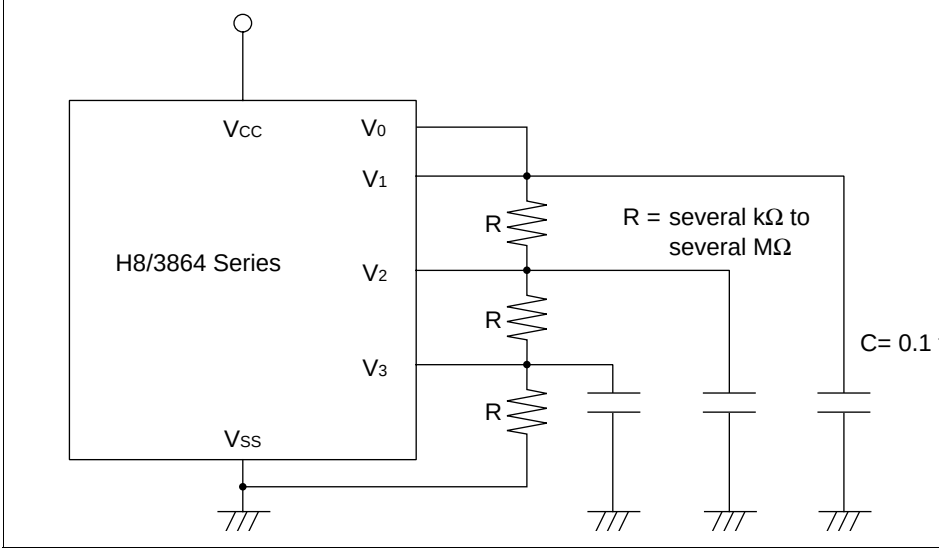
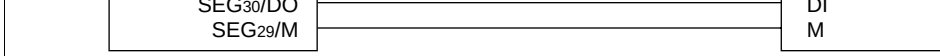


Figure 13.17 Connection of External Split-Resistance

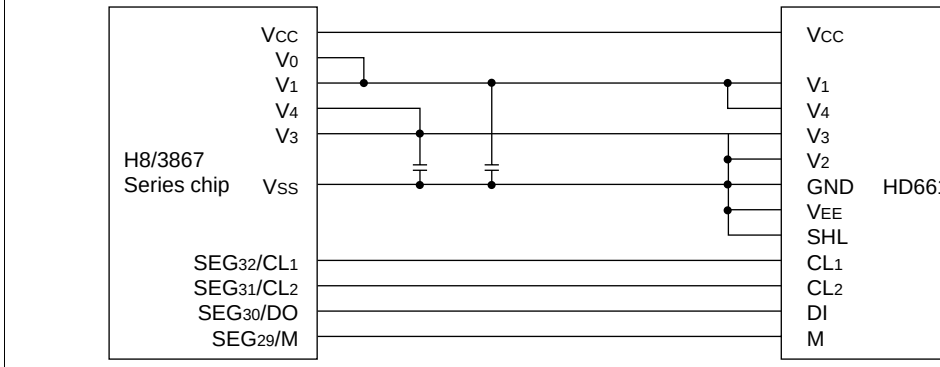
combination of the data and the M pin output, but these combinations differ from those
HD66100. Table 13.3 shows the output levels of the LCD drive power supply, and figu
and 13.16 show the common and segment waveforms for each duty cycle.

When ACT is cleared to 0, operation stops with $CL_2 = 0$, $CL_1 = 0$, M = 0, and DO at th
value (1 or 0) being output at that instant. In standby mode, the expansion pins go to th
impedance (floating) state.

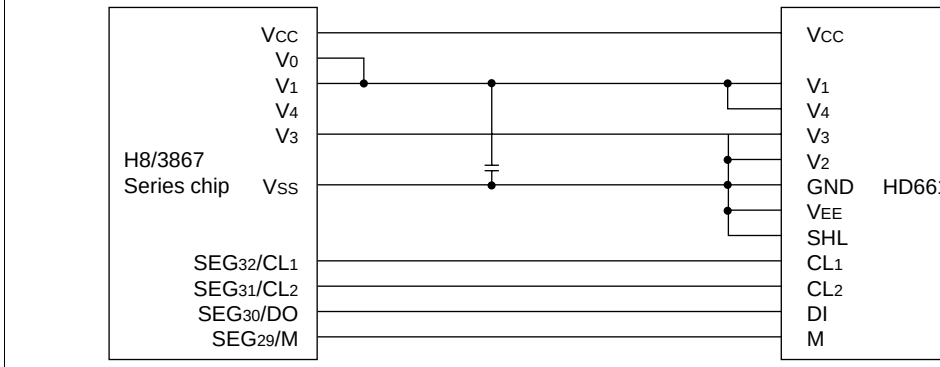
When external expansion is implemented, the load in the LCD panel increases and the
power supply may not provide sufficient current capacity. In this case, measures shoul
as described in section 13.3.7, Boosting the LCD Drive Power Supply.



(a) 1/3 bias, 1/4 or 1/3 duty



(b) 1/2 duty



(c) Static mode

Figure 13.18 Connection to HD66100

to virtually the same low level as when used at approximately 1.5 V. It is, of course, a
to use the same level of external power supply voltage and internal power supply volta
using the internal power supply step-down circuit.

14.2 When Using the Internal Power Supply Step-Down Circuit

Connect the external power supply to the V_{CC} pin, and connect a capacitance of appro
µF between CV_{CC} and V_{SS}, as shown in figure 14.1. The internal step-down circuit is
effective simply by adding this external circuit.

Notes: 1. In the external circuit interface, the external power supply voltage connecte
and the GND potential connected to V_{SS} are the reference levels. For exam
input/output levels, the V_{CC} level is the reference for the high level, and th
is that for the low level.

2. When the internal power supply step-down circuit is used, operating freque
0.4 MHz to 2 MHz when V_{CC} = 2.2 V to 5.5 V, and 0.4 MHz to 1 MHz oth

3. The LCD power supply and A/D converter analog power supply are not aff
internal step-down processing.

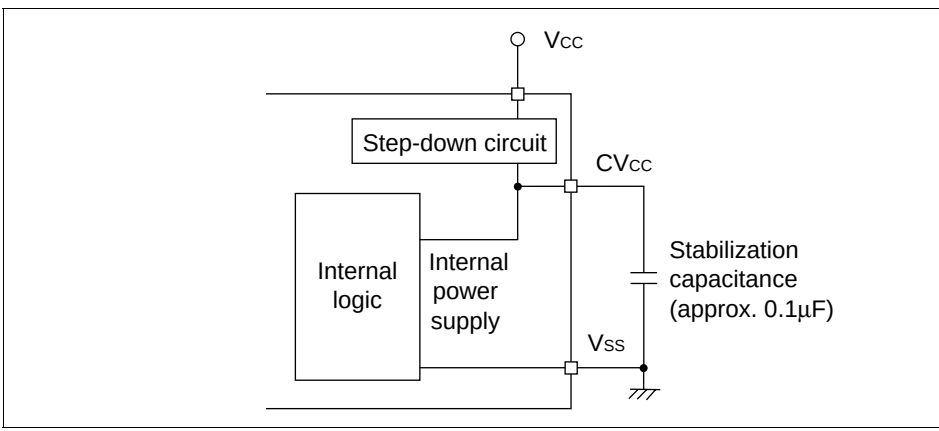


Figure 14.1 Power Supply Connection when Internal Step-Down Circuit I

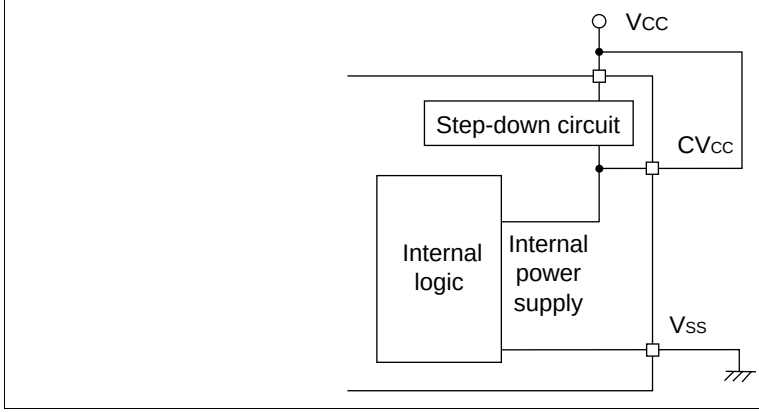
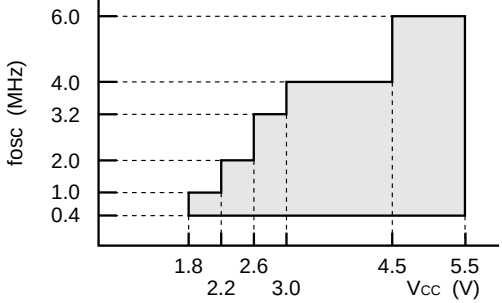


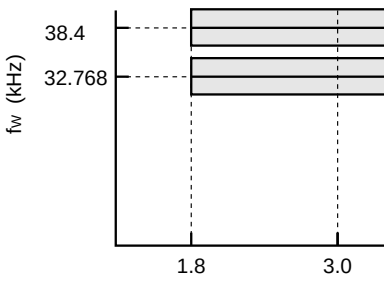
Figure 14.2 Power Supply Connection when Internal Step-Down Circuit Is N

Item		Symbol	Value
Power supply voltage		V_{CC}, CV_{CC}	–0.3 to +7.0
Analog power supply voltage		AV_{CC}	–0.3 to +7.0
Programming voltage		V_{PP}	–0.3 to +13.0
Input voltage	Ports other than Port B	V_{in}	–0.3 to V_{CC} +0.3
	Port B	AV_{in}	–0.3 to AV_{CC} +0.3
Operating temperature		T_{opr}	–20 to +75
Storage temperature		T_{stg}	–55 to +125

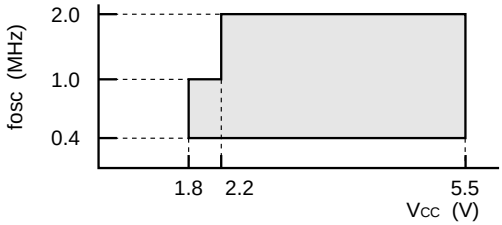
Note: Permanent damage may occur to the chip if maximum ratings are exceeded. N
operation should be under the conditions specified in Electrical Characteristics.
these values can result in incorrect operation and reduced reliability.



- Active (high-speed) mode
- Sleep (high-speed) mode
- Internal power supply step-down circuit not used

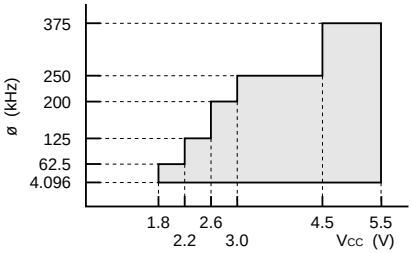


- All operating modes

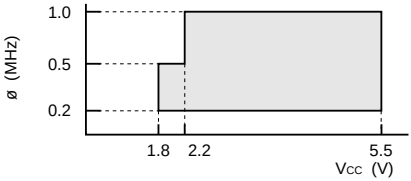


- Active (high-speed) mode
- Sleep (high-speed) mode
- Internal power supply step-down circuit used

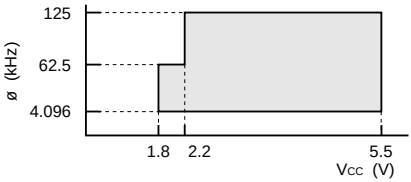
- Active (high-speed) mode
- Sleep (high-speed) mode (except CPU)
- Internal power supply step-down circuit not used



- Active (medium-speed) mode (except A/D converter and PWM)
- Sleep (medium-speed) mode (except A/D converter and PWM)
- Internal power supply step-down circuit not used

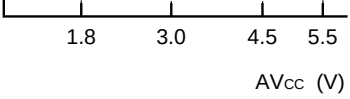


- Active (high-speed) mode
- Sleep (high-speed) mode (except CPU)
- Internal power supply step-down circuit used

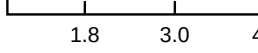


- Active (medium-speed) mode (except A/D converter and PWM)
- Sleep (medium-speed) mode (except A/D converter and PWM)
- Internal power supply step-down circuit used

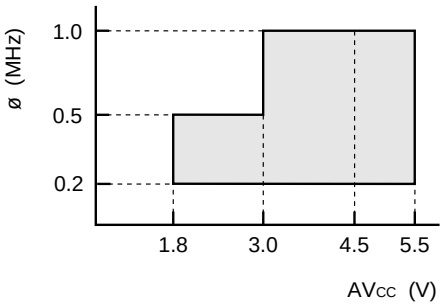
- Subactive mode
- Subsleep mode (except CPU)
- Watch mode (except CPU)



- Active (high-speed) mode
- Sleep (high-speed) mode
- Internal power supply step-down circuit not used



- Active (medium-speed) mode
- Sleep (medium-speed) mode
- Internal power supply step-d not used



- Active (high-speed) mode
- Sleep (high-speed) mode
- Internal power supply step-down circuit used

Item	Symbol	Applicable Pins	Values Min	Typ	Max	Unit	Test Condition
Input high voltage	V_{IH}	$\overline{RES}$,	0.8 V_{CC}	—	Vcc + 0.3	V	V_{CC} = 4.0 to 5.5 V
		$\overline{WKP_0}$ to $\overline{WKP_7}$, $\overline{IRQ_0}$ to $\overline{IRQ_4}$, AEVL, AEVH, TMIC, TMIF, TMIG SCK_{31}, SCK_{32}, $\overline{ADTRG}$	0.9 V_{CC}	—	V_{CC} + 0.3		Except the above
		RXD_{31}, RXD_{32},	0.7 V_{CC}	—	V_{CC} + 0.3	V	V_{CC} = 4.0 to 5.5 V
		UD	0.8 V_{CC}	—	V_{CC} + 0.3		Except the above
		OSC_1	0.8 V_{CC}	—	V_{CC} + 0.3	V	V_{CC} = 4.0 to 5.5 V
			0.9 V_{CC}	—	V_{CC} + 0.3		Except the above
		X_1	0.9 V_{CC}	—	V_{CC} + 0.3	V	V_{CC} = 1.8 V to 5.5
		$P1_0$ to $P1_7$,	0.7 V_{CC}	—	V_{CC} + 0.3	V	V_{CC} = 4.0 V to 5.5
		$P3_0$ to $P3_7$, $P4_0$ to $P4_3$, $P5_0$ to $P5_7$, $P6_0$ to $P6_7$, $P7_0$ to $P7_7$, $P8_0$ to $P8_7$, PA_0 to PA_3	0.8 V_{CC}	—	V_{CC} + 0.3		Except the above
		PB_0 to PB_7	0.7 V_{CC}	—	AV_{CC} + 0.3		V_{CC} = 4.0 V to 5.5
			0.8 V_{CC}	—	AV_{CC} + 0.3		Except the above

Note: Connect the TEST pin to V_{SS}.

		$\overline{IRQ_0}$ to $\overline{IRQ_4}$, AEVL, AEVH, TMIC, TMIF, TMIG SCK_{31}, SCK_{32}, $\overline{ADTRG}$					
		RXD_{31}, RXD_{32},	–0.3	—	0.3 V_{CC}	V	V_{CC} = 4.0 to 5.5 V
		UD	–0.3	—	0.2 V_{CC}		Except the above
		OSC_1	–0.3	—	0.2		When internal step down circuit is used.
			–0.3	—	0.2 V_{CC}	V	V_{CC} = 4.0 to 5.5 V
			–0.3	—	0.1 V_{CC}		Except the above
		X_1	–0.3	—	0.1 V_{CC}	V	V_{CC} = 1.8 V to 5.5 V
		$P1_0$ to $P1_7$,	–0.3	—	0.3 V_{CC}	V	V_{CC} = 4.0 V to 5.5 V
		$P3_0$ to $P3_7$, $P4_0$ to $P4_3$, $P5_0$ to $P5_7$, $P6_0$ to $P6_7$, $P7_0$ to $P7_7$, $P8_0$ to $P8_7$, PA_0 to PA_3, PB_0 to PB_7	–0.3	—	0.2 V_{CC}		Except the above
Output high voltage	V_{OH}	$P1_0$ to $P1_7$, $P3_0$ to $P3_7$,	V_{CC} – 1.0	—	—	V	V_{CC} = 4.0 V to 5.5 V, $-I_{OH}$ = 1.0 mA
		$P4_0$ to $P4_2$, $P5_0$ to $P5_7$,	V_{CC} – 0.5	—	—		V_{CC} = 4.0 V to 5.5 V, $-I_{OH}$ = 0.5 mA
		$P6_0$ to $P6_7$, $P7_0$ to $P7_7$, $P8_0$ to $P8_7$, PA_0 to PA_3	V_{CC} – 0.3	—	—		$-I_{OH}$ = 0.1 mA

			—	—	0.5		I_{OL} = 0.4 mA
		$P5_0$ to $P5_7$, $P6_0$ to $P6_7$, $P7_0$ to $P7_7$, $P8_0$ to $P8_7$, PA_0 to PA_3	—	—	0.5		I_{OL} = 0.4 mA
		$P3_0$ to $P3_7$	—	—	1.5		V_{CC} = 4.0 V to 5.5 I_{OL} = 10 mA
			—	—	0.6		V_{CC} = 4.0 V to 5.5 I_{OL} = 1.6 mA
			—	—	0.5		I_{OL} = 0.4 mA
Input/output leakage current	$\lvert I_{IL} \rvert$	$\overline{RES}$, $P4_3$	—	—	20.0	µA	V_{IN} = 0.5 V to V_{CC} – 0.5 V
			—	—	1.0		
		OSC_1, X_1, $P1_0$ to $P1_7$, $P3_0$ to $P3_7$, $P4_0$ to $P4_2$, $P5_0$ to $P5_7$, $P6_0$ to $P6_7$, $P7_0$ to $P7_7$, $P8_0$ to $P8_7$, PA_0 to PA_3	—	—	1.0	µA	V_{IN} = 0.5 V to V_{CC} – 0.5 V
		PB_0 to PB_7	—	—	1.0		V_{IN} = 0.5 V to AV_{CC} – 0.5 V
Pull-up MOS current	$-I_p$	$P1_0$ to $P1_7$, $P3_0$ to $P3_7$,	50.0	—	300.0	µA	V_{CC} = 5 V, V_{IN} = 0 V
		$P5_0$ to $P5_7$, $P6_0$ to $P6_7$	—	35.0	—		V_{CC} = 2.7 V, V_{IN} = 0 V

		supply, RES, $P4_3$, PB_0 to PB_7					T_a = 25°C
		$\overline{RES}$	—	—	80.0		
			—	—	15.0		
		$P4_3$	—	—	50.0		
			—	—	15.0		
		PB_0 to PB_7	—	—	15.0		
Active mode current dissipation	I_{OPE1}	V_{CC}	—	0.7	1.0	mA	Active (high-speed) mode V_{CC} = 5 V, f_{OSC} = 2MHz
	I_{OPE2}	V_{CC}	—	0.3	0.5	mA	Active (medium-speed) mode V_{CC} = 5 V, f_{OSC} = 2MHz Divided by 128
Sleep mode current dissipation	I_{SLEEP}	V_{CC}	—	0.4	0.6	mA	V_{CC}=5 V, f_{OSC}=2MHz
Subactive mode current dissipation	I_{SUB}	V_{CC}	—	15	30	µA	V_{CC} = 2.7 V, LCD on 32-kHz crystal oscillator ($ø_{SUB}=ø_W/2$)
			—	8	—	µA	V_{CC} = 2.7 V, LCD on 32-kHz crystal oscillator ($ø_{SUB}=ø_W/8$)
Subsleep mode current dissipation	I_{SUBSP}	V_{CC}	—	7.5	16	µA	V_{CC} = 2.7 V, LCD on 32-kHz crystal oscillator ($ø_{SUB}=ø_W/2$)

current dissipation							oscillator LCD not used
Standby mode current dissipation	I_{STBY}	V_{CC}	—	1.0	5.0	µA	32-kHz crystal oscillator not used
RAM data retaining voltage	V_{RAM}	V_{CC}	1.5	—	—	V	

Notes: 1. Applies to the Mask ROM products.
2. Applies to the HD6473867 and HD6473827.
3. Pin states during current measurement.

Mode	$\overline{RES}$ Pin	Internal State	Other Pins	LCD Power Supply	Oscillator
Active (high-speed) mode (I_{OPE1})	V_{CC}	Operates	V_{CC}	Halted	System cl crystal
Active (medium-speed) mode (I_{OPE2})					Subclock Pin X_1 = C
Sleep mode	V_{CC}	Only timers operate	V_{CC}	Halted	
Subactive mode	V_{CC}	Operates	V_{CC}	Halted	System cl
Subsleep mode	V_{CC}	Only timers operate, CPU stops	V_{CC}	Halted	crystal Subclock
Watch mode	V_{CC}	Only time base operates, CPU stops	V_{CC}	Halted	crystal
Standby mode	V_{CC}	CPU and timers both stop	V_{CC}	Halted	System cl crystal Subclock Pin X_1 = C

4. Excludes current in pull-up MOS transistors and output buffers.
5. When internal step-down circuit is used.

current		Port 3	—	—	10.0		V_{CC} = 4.0 V to 5.5 V
(per pin)		All output pins	—	—	0.5		
Allowable output low	I_{OL}	Output pins except port 3	—	—	40.0	mA	V_{CC} = 4.0 V to 5.5 V
current		Port 3	—	—	80.0		V_{CC} = 4.0 V to 5.5 V
(total)		All output pins	—	—	20.0		
Allowable	$-I_{OH}$	All output pins	—	—	2.0	mA	V_{CC} = 4.0 V to 5.5 V
output high current (per pin)			—	—	0.2		Except the above
Allowable	$-I_{OH}$	All output pins	—	—	15.0	mA	V_{CC} = 4.0 V to 5.5 V
output high			—	—	10.0		Except the above

Item	Symbol	Applicable Pins	Values Min	Typ	Max	Unit	Test Condition
System clock	f_{OSC}	OSC_1, OSC_2	0.4	—	6	MHz	V_{CC} = 4.5 V to 5.5
oscillation			0.4	—	4		V_{CC} = 3.0 V to 5.5
frequency			0.4	—	3.2		V_{CC} = 2.6 V to 5.5
			0.4	—	2		V_{CC} = 2.2 V to 5.5
			0.4	—	1		Except the above
OSC clock ($ø_{OSC}$)	t_{OSC}	OSC_1, OSC_2	167	—	2500	ns	V_{CC} = 4.5 V to 5.5
cycle time			250	—	2500		V_{CC} = 3.0 V to 5.5
			313	—	2500		V_{CC} = 2.6 V to 5.5
			500	—	2500		V_{CC} = 2.2 V to 5.5
			1000	—	2500		Except the above
System clock (ø)	t_{cyc}		2	—	128	t_{OSC}	
cycle time			—	—	244.1	µs	
Subclock oscillation frequency	f_W	X_1, X_2	—	32.768 or 38.4	—	kHz	
Watch clock ($ø_W$) cycle time	t_W	X_1, X_2	—	30.5 or 26.0	—	µs	
Subclock ($ø_{SUB}$) cycle time	t_{subcyc}		2	—	8	t_W	
Instruction cycle time			2	—	—	t_{cyc} t_{subcyc}	
Oscillation stabilization time	t_{rc}	OSC_1, OSC_2	—	20	45	µs	Figure 15.9 V_{CC} = 2.2 V to 5.5
			—	0.1	8	ms	Figure 15.9 V_{CC} = 2.2 V to 5.5
			—	—	50	ms	Except the above

External clock high width	t_{CPH}	OSC_1	70	—	—	ns	V_{CC} = 4.5 V to 5.5 V
			100	—	—		V_{CC} = 3.0 V to 5.5 V
			140	—	—		V_{CC} = 2.6 V to 5.5 V
			200	—	—		V_{CC} = 2.2 V to 5.5 V
			400	—	—		Except the above
		X_1	—	15.26 or 13.02	—	µs	
External clock low width	t_{CPL}	OSC_1	70	—	—	ns	V_{CC} = 4.5 V to 5.5 V
			100	—	—		V_{CC} = 3.0 V to 5.5 V
			140	—	—		V_{CC} = 2.6 V to 5.5 V
			200	—	—		V_{CC} = 2.2 V to 5.5 V
			400	—	—		Except the above
		X_1	—	15.26 or 13.02	—	µs	
External clock rise time	t_{CPr}	OSC_1	—	—	20	ns	V_{CC} = 4.5 V to 5.5 V
			—	—	30		V_{CC} = 2.6 V to 5.5 V
			—	—	55		Except the above
		X_1	—	—	55.0	ns	
External clock fall time	t_{CPf}	OSC_1	—	—	20	ns	V_{CC} = 4.5 V to 5.5 V
			—	—	30		V_{CC} = 2.6 V to 5.5 V
			—	—	55		Except the above
		X_1	—	—	55.0	ns	
Pin $\overline{RES}$ low width	t_{REL}	$\overline{RES}$	10	—	—	t_{cyc}	

		$\overline{ADTRG}$, TMIC, TMIF, TMIG, AEVL, AEVH				
Input pin low width	t_{IL}	$\overline{IRQ_0}$ to $\overline{IRQ_4}$, $\overline{WKP_0}$ to $\overline{WKP_7}$, $\overline{ADTRG}$, TMIC, TMIF, TMIG, AEVL, AEVH	2	—	—	t_{cyc} t_{subcyc}
UD pin minimum modulation width	t_{UDH} t_{UDL}	UD	4	—	—	t_{cyc} t_{subcyc}

Notes: 1. Selected with SA1 and SA0 of system clock control register 2 (SYSCR2).
2. Internal power supply step-down circuit not used

Table 15.4 Serial Interface (SCI3-1, SCI3-2) Timing

V_{CC} = 1.8 V to 5.5 V, AV_{CC} = 1.8 V to 5.5 V, V_{SS} = AV_{SS} = 0.0 V, T_a = –20°C to +7
(including subactive mode) unless otherwise indicated.

			Values				
Item		**Symbol**	**Min**	**Typ**	**Max**	**Unit**	**Test Conditions**
Input clock cycle	Asynchronous	t_{scyc}	4	—	—	t_{cyc} or t_{subcyc}	
	Synchronous		6	—	—		
Input clock pulse width		t_{SCKW}	0.4	—	0.6	t_{scyc}	
Transmit data delay time (synchronous)		t_{TXD}	—	—	1	t_{cyc} or t_{subcyc}	V_{CC} = 4.0 V to 5.5 V
			—	—	1		Except the above
Receive data setup time (synchronous)		t_{RXS}	200.0	—	—	ns	V_{CC} = 4.0 V to 5.5 V
			400.0	—	—		Except the above
Receive data hold time (synchronous)		t_{RXH}	200.0	—	—	ns	V_{CC} = 4.0 V to 5.5 V
			400.0	—	—		Except the above

Note: 1. When internal step-down circuit is not used.

Item	Symbol	Pins	Min	Typ	Max	Unit	Test Condition
Analog power supply voltage	AV_{CC}	AV_{CC}	1.8	—	5.5	V	
Analog input voltage	AV_{IN}	AN_0 to AN_7	– 0.3	—	AV_{CC} + 0.3	V	
Analog power supply current	AI_{OPE}	AV_{CC}	—	—	1.5	mA	AV_{CC} = 5 V
	AI_{STOP1}	AV_{CC}	—	600	—	µA	
	AI_{STOP2}	AV_{CC}	—	—	5	µA	
Analog input capacitance	C_{AIN}	AN_0 to AN_7	—	—	15.0	pF	
Allowable signal source impedance	R_{AIN}		—	—	10.0	k	
Resolution (data length)			—	—	10	bit	
Nonlinearity error			—	—	±2.5	LSB	AV_{CC} = 3.0 V to 5.5 V V_{CC} = 3.0 V to 5.5 V
			—	—	±5.5		AV_{CC} = 2.0 V to 5.5 V V_{CC} = 2.0 V to 5.5 V
			—	—	±7.5		Except the above
Quantization error			—	—	±0.5	LSB	

	—	—	±6.0		AV_{CC} = 2.0 V to 5.5 V V_{CC} = 2.0 V to 5.5 V
	—	—	±8.0		Except the above
Conversion time	15.5	—	155	µs	AV_{CC} = 3.0 V to 5.5 V V_{CC} = 3.0 V to 5.5 V
	62	—	155		Except the above

Notes: 1. Set $AV_{CC} = V_{CC}$ when the A/D converter is not used.
2. AI_{STOP1} is the current in active and sleep modes while the A/D converter is
3. AI_{STOP2} is the current at reset and in standby, watch, subactive, and subsle while the A/D converter is idle.
4. When internal step-down circuit is not used.
5. Conversion time 124 µs

Item	Symbol	Applicable Pins	Test Conditions	Values			
				Min	Typ	Max	Unit
Segment driver drop voltage	V_{DS}	SEG_1 to SEG_{32}	I_D = 2 µA V_1 = 2.7 to 5.5 V	—	—	0.6	V
Common driver drop voltage	V_{DC}	COM_1 to COM_4	I_D = 2 µA V_1 = 2.7 to 5.5 V	—	—	0.3	V
LCD power supply split-resistance	R_{LCD}		Between V_1 and V_{SS}	0.5	3.0	9.0	M
On-chip liquid crystal display power supply voltage	V_{LP}	V_0		—	5.0	—	V
Liquid crystal display voltage	V_{LCD}	V_1		2.2	—	5.5	V

Notes: 1. The voltage drop from power supply pins V_1, V_2, V_3, and VSS to each segm common pin.

2. The output voltage in step-up constant-voltage power supply operation (unlo

3. When the liquid crystal display voltage is supplied from an external power so ensure that the following relationship is maintained: V_1 V $_2$ V $_3$ V $_{SS}$.

Clock setup time	t_{CSU}	CL_1, CL_2	*1	500	—	—	ns
Data setup time	t_{SU}	DO	*1	300	—	—	ns
Data hold time	t_{DH}	DO	*1	300	—	—	ns
M delay time	t_{DM}	M		–1000	—	1000	ns
Clock rise and fall times	t_{CT}	CL_1, CL_2		—	—	170	ns

Note: 1. Value when the frame frequency is set to between 30.5 Hz and 488 Hz.

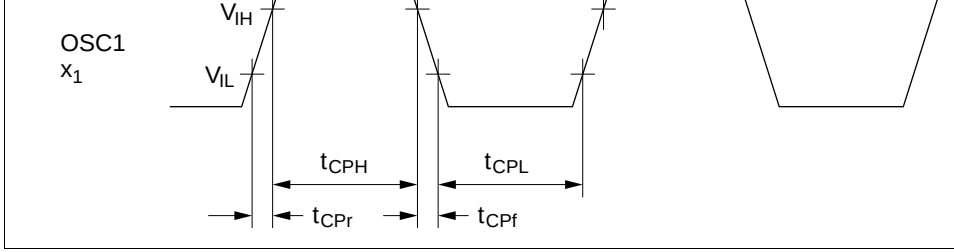


Figure 15.1 Clock Input Timing

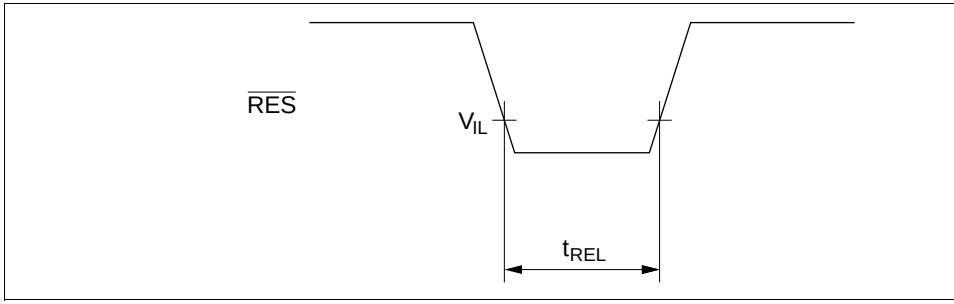


Figure 15.2 RES Low Width

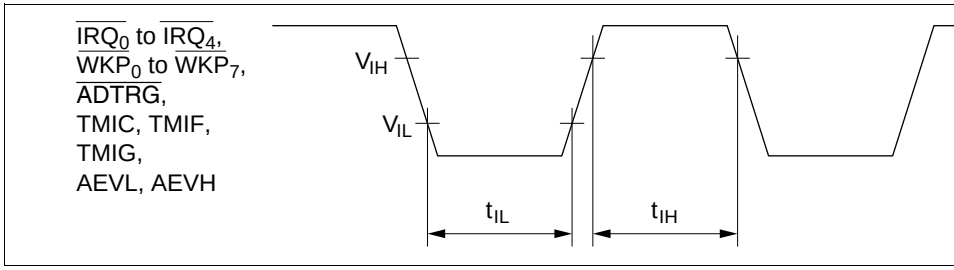


Figure 15.3 Input Timing

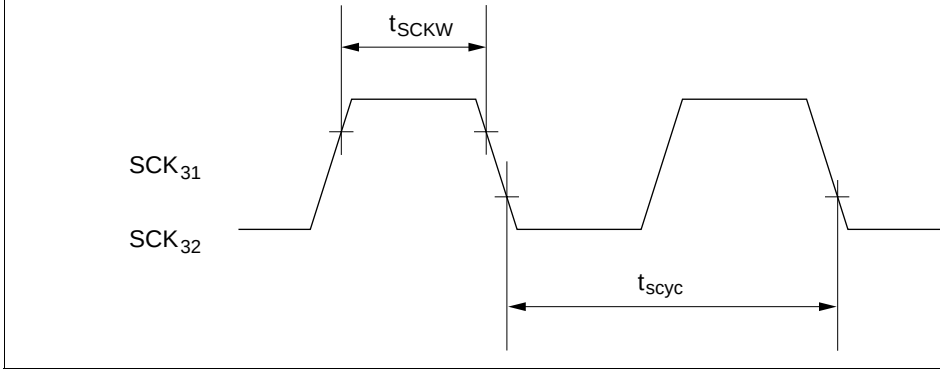
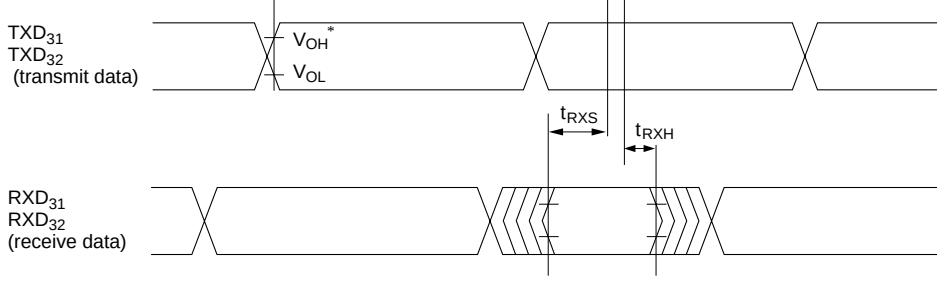


Figure 15.5 SCK3 Input Clock Timing



Note: * Output timing reference levels

Output high	V_{OH} = 1/2Vcc + 0.2 V
Output low	V_{OL} = 0.8 V

Load conditions are shown in figure 15-8.

Figure 15.6 SCI3 Synchronous Mode Input/Output Timing

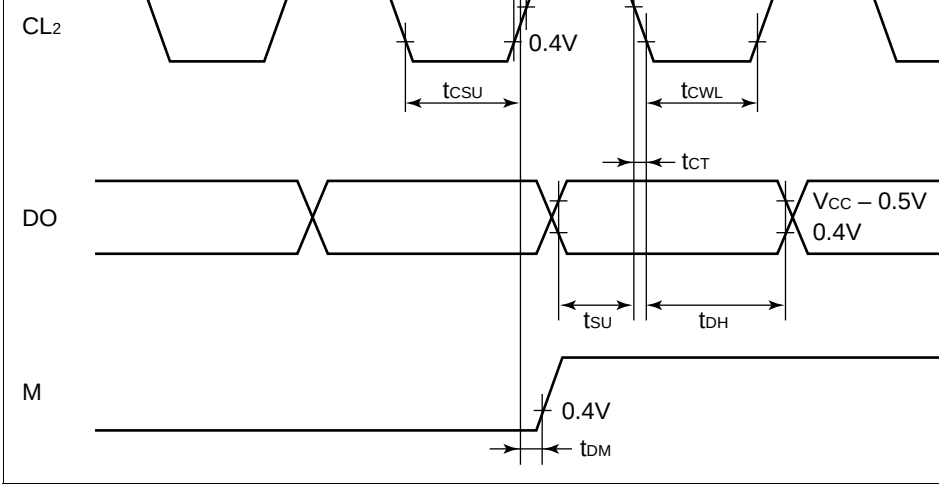


Figure 15.7 Segment Expansion Signal Timing

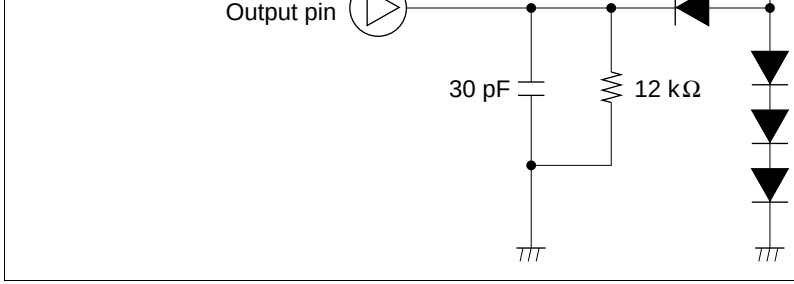
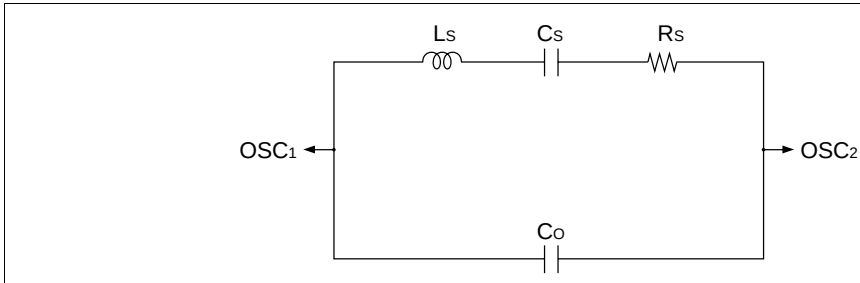


Figure 15.8 Output Load Condition

15.5 Resonator Equivalent Circuit



Crystal Resonator Parameter

Frequency (MHz)	1	4.193
RS (max)	40 Ω	100 Ω
CO (max)	3.5 pF	16 pF

Ceramic Resonator Parameters

Frequency (MHz)	0.4	
RS (max)	8.6 Ω	
CO (max)	326 pF	

Figure 15.9 Resonator Equivalent Circuit

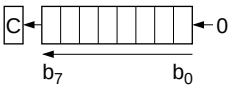
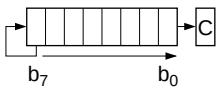
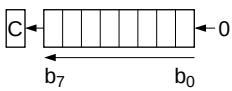
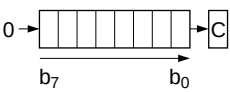
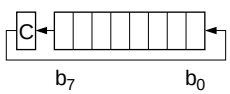
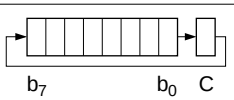
Rn8/16	General register (8 or 16 bits)
CCR	Condition code register
N	N (negative) flag in CCR
Z	Z (zero) flag in CCR
V	V (overflow) flag in CCR
C	C (carry) flag in CCR
PC	Program counter
SP	Stack pointer
#xx: 3/8/16	Immediate data (3, 8, or 16 bits)
d: 8/16	Displacement (8 or 16 bits)
@aa: 8/16	Absolute address (8 or 16 bits)
+	Addition
–	Subtraction
×	Multiplication
÷	Division
∧	Logical AND
∨	Logical OR
⊕	Exclusive logical OR
→	Move
—	Logical complement

Condition Code Notation

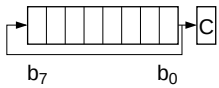
Symbol	
↕	Modified according to the instruction result
*	Not fixed (value not guaranteed)
0	Always cleared to 0
—	Not affected by the instruction execution result

Mnemonic	Op	Operation	#xx	Rn	@Rn	@(d,Rn)	@-Rn	@aa	@(d,PC)	@@aa	Imm	I	H	N
MOV.B #xx:8, Rd	B	#xx:8 → Rd8	2									—	—	↕
MOV.B Rs, Rd	B	Rs8 → Rd8		2								—	—	↕
MOV.B @Rs, Rd	B	@Rs16 → Rd8			2							—	—	↕
MOV.B @(d:16, Rs), Rd	B	@(d:16, Rs16)→ Rd8				4						—	—	↕
MOV.B @Rs+, Rd	B	@Rs16 → Rd8 Rs16+1 → Rs16					2					—	—	↕
MOV.B @aa:8, Rd	B	@aa:8 → Rd8						2				—	—	↕
MOV.B @aa:16, Rd	B	@aa:16 → Rd8						4				—	—	↕
MOV.B Rs, @Rd	B	Rs8 → @Rd16			2							—	—	↕
MOV.B Rs, @(d:16, Rd)	B	Rs8 → @(d:16, Rd16)				4						—	—	↕
MOV.B Rs, @–Rd	B	Rd16–1 → Rd16 Rs8 → @Rd16					2					—	—	↕
MOV.B Rs, @aa:8	B	Rs8 → @aa:8						2				—	—	↕
MOV.B Rs, @aa:16	B	Rs8 → @aa:16						4				—	—	↕
MOV.W #xx:16, Rd	W	#xx:16 → Rd	4									—	—	↕
MOV.W Rs, Rd	W	Rs16 → Rd16		2								—	—	↕
MOV.W @Rs, Rd	W	@Rs16 → Rd16			2							—	—	↕
MOV.W @(d:16, Rs), Rd	W	@(d:16, Rs16) → Rd16				4						—	—	↕
MOV.W @Rs+, Rd	W	@Rs16 → Rd16 Rs16+2 → Rs16					2					—	—	↕
MOV.W @aa:16, Rd	W	@aa:16 → Rd16						4				—	—	↕
MOV.W Rs, @Rd	W	Rs16 → @Rd16			2							—	—	↕
MOV.W Rs, @(d:16, Rd)	W	Rs16 → @(d:16, Rd16)				4						—	—	↕
MOV.W Rs, @–Rd	W	Rd16–2 → Rd16 Rs16 → @Rd16					2					—	—	↕
MOV.W Rs, @aa:16	W	Rs16 → @aa:16						4				—	—	↕
POP Rd	W	@SP → Rd16 SP+2 → SP					2					—	—	↕
PUSH Rs	W	SP–2 → SP Rs16 → @SP					2					—	—	↕

ADD.B Rs, Rd	B	Rd8+Rs8 → Rd8		2								—	↕	↕
ADD.W Rs, Rd	W	Rd16+Rs16 → Rd16		2								—	(1)	↕
ADDX.B #xx:8, Rd	B	Rd8+#xx:8 +C → Rd8	2									—	↕	↕
ADDX.B Rs, Rd	B	Rd8+Rs8 +C → Rd8		2								—	↕	↕
ADDS.W #1, Rd	W	Rd16+1 → Rd16		2								—	—	—
ADDS.W #2, Rd	W	Rd16+2 → Rd16		2								—	—	—
INC.B Rd	B	Rd8+1 → Rd8		2								—	—	↕
DAA.B Rd	B	Rd8 decimal adjust → Rd8		2								—	*	↕
SUB.B Rs, Rd	B	Rd8–Rs8 → Rd8		2								—	↕	↕
SUB.W Rs, Rd	W	Rd16–Rs16 → Rd16		2								—	(1)	↕
SUBX.B #xx:8, Rd	B	Rd8–#xx:8 –C → Rd8	2									—	↕	↕
SUBX.B Rs, Rd	B	Rd8–Rs8 –C → Rd8		2								—	↕	↕
SUBS.W #1, Rd	W	Rd16–1 → Rd16		2								—	—	—
SUBS.W #2, Rd	W	Rd16–2 → Rd16		2								—	—	—
DEC.B Rd	B	Rd8–1 → Rd8		2								—	—	↕
DAS.B Rd	B	Rd8 decimal adjust → Rd8		2								—	*	↕
NEG.B Rd	B	0–Rd → Rd		2								—	↕	↕
CMP.B #xx:8, Rd	B	Rd8–#xx:8	2									—	↕	↕
CMP.B Rs, Rd	B	Rd8–Rs8		2								—	↕	↕
CMP.W Rs, Rd	W	Rd16–Rs16		2								—	(1)	↕

MULXU.B Rs, Rd	B	Rd8×Rs8 → Rd16		2								—	—		
DIVXU.B Rs, Rd	B	Rd16÷Rs8 → Rd16 (RdH: remainder, RdL: quotient)		2								—	—	(5)	
AND.B #xx:8, Rd	B	Rd8∧#xx:8 → Rd8	2									—	—	↕	
AND.B Rs, Rd	B	Rd8∧Rs8 → Rd8		2								—	—	↕	
OR.B #xx:8, Rd	B	Rd8∨#xx:8 → Rd8	2									—	—	↕	
OR.B Rs, Rd	B	Rd8∨Rs8 → Rd8		2								—	—	↕	
XOR.B #xx:8, Rd	B	Rd8⊕#xx:8 → Rd8	2									—	—	↕	
XOR.B Rs, Rd	B	Rd8⊕Rs8 → Rd8		2								—	—	↕	
NOT.B Rd	B	$\overline{\text{Rd}}$ → Rd		2								—	—	↕	
SHAL.B Rd	B			2								—	—	↕	
SHAR.B Rd	B			2								—	—	↕	
SHLL.B Rd	B			2								—	—	↕	
SHLR.B Rd	B			2								—	—	0	
ROTXL.B Rd	B			2								—	—	↕	
ROTXR.B Rd	B			2								—	—	↕	

RENESAS

														
ROTR.B Rd	B			2								—	—	↕
BSET #xx:3, Rd	B	(#xx:3 of Rd8) ← 1		2								—	—	—
BSET #xx:3, @Rd	B	(#xx:3 of @Rd16) ← 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, @Rd	B	(Rn8 of @Rd16) ← 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, @Rd	B	(#xx:3 of @Rd16) ← 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, @Rd	B	(Rn8 of @Rd16) ← 0			4							—	—	—
BCLR Rn, @aa:8	B	(Rn8 of @aa:8) ← 0						4				—	—	—
BNOT #xx:3, Rd	B	(#xx:3 of Rd8) ← $\overline{\text{(\#xx:3 of Rd8)}}$		2								—	—	—
BNOT #xx:3, @Rd	B	(#xx:3 of @Rd16) ← $\overline{\text{(\#xx:3 of @Rd16)}}$			4							—	—	—
BNOT #xx:3, @aa:8	B	(#xx:3 of @aa:8) ← $\overline{\text{(\#xx:3 of @aa:8)}}$						4				—	—	—
BNOT Rn, Rd	B	(Rn8 of Rd8) ← $\overline{\text{(Rn8 of Rd8)}}$		2								—	—	—
BNOT Rn, @Rd	B	(Rn8 of @Rd16) ← $\overline{\text{(Rn8 of @Rd16)}}$			4							—	—	—
BNOT Rn, @aa:8	B	(Rn8 of @aa:8) ← $\overline{\text{(Rn8 of @aa:8)}}$						4				—	—	—

BTST #xx:3, @Rd	B	(#xx:3 of @$\overline{\text{Rd16}}$) → Z			4							—	—	—
BTST #xx:3, @aa:8	B	(#xx:3 of @$\overline{\text{aa:8}}$) → Z						4				—	—	—
BTST Rn, Rd	B	($\overline{\text{Rn8 of Rd8}}$) → Z		2								—	—	—
BTST Rn, @Rd	B	($\overline{\text{Rn8 of @Rd16}}$) → Z			4							—	—	—
BTST Rn, @aa:8	B	($\overline{\text{Rn8 of @aa:8}}$) → Z						4				—	—	—
BLD #xx:3, Rd	B	(#xx:3 of Rd8) → C		2								—	—	—
BLD #xx:3, @Rd	B	(#xx:3 of @Rd16) → C			4							—	—	—
BLD #xx:3, @aa:8	B	(#xx:3 of @aa:8) → C						4				—	—	—
BILD #xx:3, Rd	B	(#xx:3 of $\overline{\text{Rd8}}$) → C		2								—	—	—
BILD #xx:3, @Rd	B	(#xx:3 of @$\overline{\text{Rd16}}$) → C			4							—	—	—
BILD #xx:3, @aa:8	B	(#xx:3 of @$\overline{\text{aa:8}}$) → C						4				—	—	—
BST #xx:3, Rd	B	C → (#xx:3 of Rd8)		2								—	—	—
BST #xx:3, @Rd	B	C → (#xx:3 of @Rd16)			4							—	—	—
BST #xx:3, @aa:8	B	C → (#xx:3 of @aa:8)						4				—	—	—
BIST #xx:3, Rd	B	$\overline{\text{C}}$ → (#xx:3 of Rd8)		2								—	—	—
BIST #xx:3, @Rd	B	$\overline{\text{C}}$ → (#xx:3 of @Rd16)			4							—	—	—
BIST #xx:3, @aa:8	B	$\overline{\text{C}}$ → (#xx:3 of @aa:8)						4				—	—	—
BAND #xx:3, Rd	B	C∧(#xx:3 of Rd8) → C		2								—	—	—
BAND #xx:3, @Rd	B	C∧(#xx:3 of @Rd16) → C			4							—	—	—
BAND #xx:3, @aa:8	B	C∧(#xx:3 of @aa:8) → C						4				—	—	—
BIAND #xx:3, Rd	B	C∧($\overline{\text{#xx:3 of Rd8}}$) → C		2								—	—	—
BIAND #xx:3, @Rd	B	C∧($\overline{\text{#xx:3 of @Rd16}}$) → C			4							—	—	—
BIAND #xx:3, @aa:8	B	C∧($\overline{\text{#xx:3 of @aa:8}}$) → C						4				—	—	—
BOR #xx:3, Rd	B	C∨(#xx:3 of Rd8) → C		2								—	—	—
BOR #xx:3, @Rd	B	C∨(#xx:3 of @Rd16) → C			4							—	—	—
BOR #xx:3, @aa:8	B	C∨(#xx:3 of @aa:8) → C						4				—	—	—
BIOR #xx:3, Rd	B	C∨($\overline{\text{#xx:3 of Rd8}}$) → C		2								—	—	—
BIOR #xx:3, @Rd	B	C∨($\overline{\text{#xx:3 of @Rd16}}$) → C			4							—	—	—

BXOR #xx:3, Rd	B	C⊕(#xx:3 of Rd8) → C			2									—	—	—
BXOR #xx:3, @Rd	B	C⊕(#xx:3 of @Rd16) → C				4								—	—	—
BXOR #xx:3, @aa:8	B	C⊕(#xx:3 of @aa:8) → C							4					—	—	—
BIXOR #xx:3, Rd	B	C⊕($\overline{\text{#xx:3 of Rd8}}$) → C			2									—	—	—
BIXOR #xx:3, @Rd	B	C⊕($\overline{\text{#xx:3 of @Rd16}}$) → C				4								—	—	—
BIXOR #xx:3, @aa:8	B	C⊕($\overline{\text{#xx:3 of @aa:8}}$) → C							4					—	—	—
BRA d:8 (BT d:8)	—	PC ← PC+d:8								2				—	—	—
BRN d:8 (BF d:8)	—	PC ← PC+2								2				—	—	—
BHI d:8	—	If condition is true then PC ← PC+d:8 else next;	C ∨ Z = 0							2				—	—	—
BLS d:8	—		C ∨ Z = 1							2				—	—	—
BCC d:8 (BHS d:8)	—		C = 0							2				—	—	—
BCS d:8 (BLO d:8)	—		C = 1							2				—	—	—
BNE d:8	—		Z = 0							2				—	—	—
BEQ d:8	—		Z = 1							2				—	—	—
BVC d:8	—		V = 0							2				—	—	—
BVS d:8	—		V = 1							2				—	—	—
BPL d:8	—		N = 0							2				—	—	—
BMI d:8	—		N = 1							2				—	—	—
BGE d:8	—		N⊕V = 0							2				—	—	—
BLT d:8	—		N⊕V = 1							2				—	—	—
BGT d:8	—		Z ∨ (N⊕V) = 0							2				—	—	—
BLE d:8	—		Z ∨ (N⊕V) = 1							2				—	—	—
JMP @Rn	—	PC ← Rn16				2								—	—	—
JMP @aa:16	—	PC ← aa:16							4					—	—	—
JMP @@aa:8	—	PC ← @aa:8									2			—	—	—
BSR d:8	—	SP–2 → SP PC → @SP PC ← PC+d:8								2				—	—	—

		PC → @SP PC ← Rn16													
JSR @aa:16	—	SP–2 → SP PC → @SP PC ← aa:16						4				—	—	—	-
JSR @@aa:8		SP–2 → SP PC → @SP PC ← @aa:8								2		—	—	—	-
RTS	—	PC ← @SP SP+2 → SP									2	—	—	—	-
RTE	—	CCR ← @SP SP+2 → SP PC ← @SP SP+2 → SP									2	↕	↕	↕	
SLEEP	—	Transit to sleep mode.									2	—	—	—	-
LDC #xx:8, CCR	B	#xx:8 → CCR	2									↕	↕	↕	
LDC Rs, CCR	B	Rs8 → CCR		2								↕	↕	↕	
STC CCR, Rd	B	CCR → Rd8		2								—	—	—	-
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	—	if R4L≠0 Repeat @R5 → @R6 R5+1 → R5 R6+1 → R6 R4L–1 → R4L Until R4L=0 else next;									4	—	—	—	-

Notes: (1) Set to 1 when there is a carry or borrow from bit 11; otherwise cleared to 0.

(2) If the result is zero, the previous value of the flag is retained; otherwise the flag is cleare

(3) Set to 1 if decimal adjustment produces a carry; otherwise retains value prior to arithme

(4) The number of states required for execution is 4n + 9 (n = value of R4L).

(5) Set to 1 if the divisor is negative; otherwise cleared to 0.

(6) Set to 1 if the divisor is zero; otherwise cleared to 0.

Table A.2 Operation Code Map

High \ Low	0	1	2	3	4	5	6	7	8	9	A	B
0	NOP	SLEEP	STC	LDC	ORC	XORC	ANDC	LDC	ADD		INC	ADDS
1	SHLL / SHAL	SHLR / SHAR	ROTXL / ROTL	ROTXR / ROTR	OR	XOR	AND	NOT / NEG	SUB		DEC	SUBS
2	MOV											
3												
4	BRA	BRN	BHI	BLS	BCC	BCS	BNE	BEQ	BVC	BVS	BPL	BMI
5	MULXU	DIVXU			RTS	BSR	RTE		JMP			
6	BSET	BNOT	BCLR	BTST				BST / BIST	MO			
7					BOR / BIOR	BXOR / BIXOR	BAND / BIAND	BLD / BILD		MOV		EEPMOV
8	ADD											
9	ADDX											
A	CMP											
B	SUBX											
C	OR											
D	XOR											
E	AND											
F	MOV											

Examples: When instruction is fetched from on-chip ROM, and an on-chip RAM is a

BSET #0, @FF00
From table A.4:
I = L = 2, J = K = M = N= 0
From table A.3:
$S_I = 2$, $S_L = 2$
Number of states required for execution = $2 \times 2 + 2 \times 2 = 8$
When instruction is fetched from on-chip ROM, branch address is read from on-chip R
on-chip RAM is used for stack area.

JSR @@ 30
From table A.4:
I = 2, J = K = 1, L = M = N = 0
From table A.3:
$S_I = S_J = S_K = 2$
Number of states required for execution = $2 \times 2 + 1 \times 2 + 1 \times 2 = 8$

Word data access	S_M		—
Internal operation	S_N	1	

Note: * Depends on which on-chip module is accessed. See 2.9.1, Notes on Data Acce
details.

ADDX	ADDX.B #xx:8, Rd	1		
	ADDX.B Rs, Rd	1		
AND	AND.B #xx:8, Rd	1		
	AND.B Rs, Rd	1		
ANDC	ANDC #xx:8, CCR	1		
BAND	BAND #xx:3, Rd	1		
	BAND #xx:3, @Rd	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		
BCLR	BCLR #xx:3, Rd	1		
	BCLR #xx:3, @Rd	2		2
	BCLR #xx:3, @aa:8	2		2
	BCLR Rn, Rd	1		
	BCLR Rn, @Rd	2		2
	BCLR Rn, @aa:8	2		2
BIAND	BIAND #xx:3, Rd	1		
	BIAND #xx:3, @Rd	2		1
	BIAND #xx:3, @aa:8	2		1

	BIOR #xx:3, @aa:8	2		1
BIST	BIST #xx:3, Rd	1		
	BIST #xx:3, @Rd	2		2
	BIST #xx:3, @aa:8	2		2
BIXOR	BIXOR #xx:3, Rd	1		
	BIXOR #xx:3, @Rd	2		1
	BIXOR #xx:3, @aa:8	2		1
BLD	BLD #xx:3, Rd	1		
	BLD #xx:3, @Rd	2		1
	BLD #xx:3, @aa:8	2		1
BNOT	BNOT #xx:3, Rd	1		
	BNOT #xx:3, @Rd	2		2
	BNOT #xx:3, @aa:8	2		2
	BNOT Rn, Rd	1		
	BNOT Rn, @Rd	2		2
	BNOT Rn, @aa:8	2		2
BOR	BOR #xx:3, Rd	1		
	BOR #xx:3, @Rd	2		1
	BOR #xx:3, @aa:8	2		1
BSET	BSET #xx:3, Rd	1		
	BSET #xx:3, @Rd	2		2
	BSET #xx:3, @aa:8	2		2
	BSET Rn, Rd	1		
	BSET Rn, @Rd	2		2
	BSET Rn, @aa:8	2		2
BSR	BSR d:8	2	1	
BST	BST #xx:3, Rd	1		
	BST #xx:3, @Rd	2		2
	BST #xx:3, @aa:8	2		2
BTST	BTST #xx:3, Rd	1		
	BTST #xx:3, @Rd	2		1
	BTST #xx:3, @aa:8	2		1
	BTST Rn, Rd	1		
	BTST Rn, @Rd	2		1

	CMP.B Rs, Rd	1						
	CMP.W Rs, Rd	1						
DAA	DAA.B Rd	1						
DAS	DAS.B Rd	1						
DEC	DEC.B Rd	1						
DIVXU	DIVXU.B Rs, Rd	1						
EEPMOV	EEPMOV	2				2n+2*		
INC	INC.B Rd	1						
JMP	JMP @Rn	2						
	JMP @aa:16	2						
	JMP @@aa:8	2	1					
JSR	JSR @Rn	2		1				
	JSR @aa:16	2		1				
	JSR @@aa:8	2	1	1				
LDC	LDC #xx:8, CCR	1						
	LDC Rs, CCR	1						
MOV	MOV.B #xx:8, Rd	1						
	MOV.B Rs, Rd	1						
	MOV.B @Rs, Rd	1				1		
	MOV.B @(d:16, Rs), Rd	2				1		
	MOV.B @Rs+, Rd	1				1		
	MOV.B @aa:8, Rd	1				1		
	MOV.B @aa:16, Rd	2				1		
	MOV.B Rs, @Rd	1				1		
	MOV.B Rs, @(d:16, Rd)	2				1		
	MOV.B Rs, @–Rd	1				1		
	MOV.B Rs, @aa:8	1				1		
	MOV.B Rs, @aa:16	2				1		
	MOV.W #xx:16, Rd	2						
	MOV.W Rs, Rd	1						
	MOV.W @Rs, Rd	1						1
	MOV.W @(d:16, Rs), Rd	2						1
	MOV.W @Rs+, Rd	1						1
	MOV.W @aa:16, Rd	2						1

Note: n: Initial value in R4L. The source and destination operands are accessed n + 1 times each.

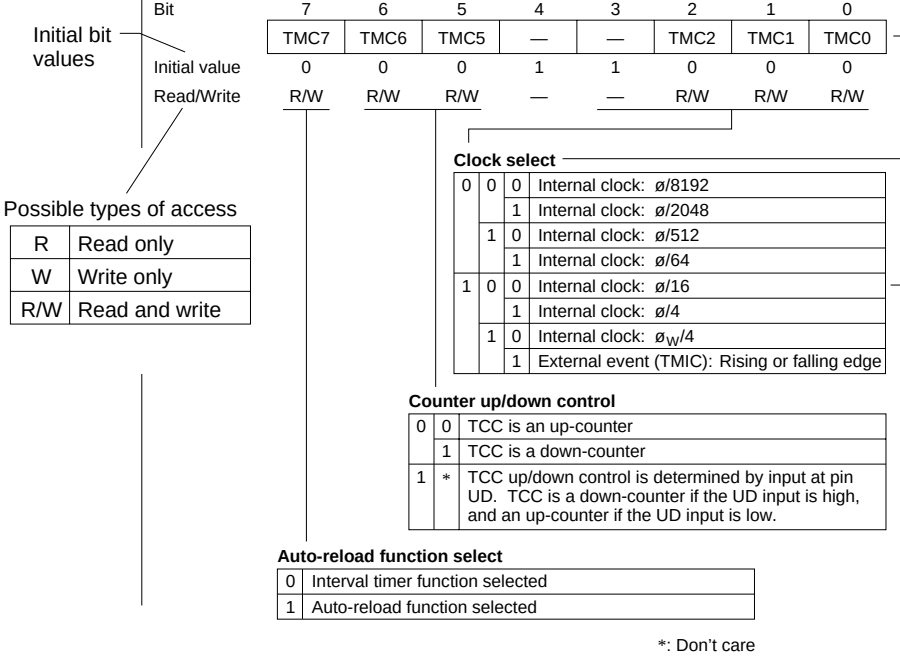
NEG	NEG.B Rd	1		
NOP	NOP	1		
NOT	NOT.B Rd	1		
OR	OR.B #xx:8, Rd	1		
	OR.B Rs, Rd	1		
ORC	ORC #xx:8, CCR	1		
ROTL	ROTL.B Rd	1		
ROTR	ROTR.B Rd	1		
ROTXL	ROTXL.B Rd	1		
ROTXR	ROTXR.B Rd	1		
RTE	RTE	2	2	
RTS	RTS	2	1	
SHAL	SHAL.B Rd	1		
SHAR	SHAR.B Rd	1		
SHLL	SHLL.B Rd	1		
SHLR	SHLR.B Rd	1		
SLEEP	SLEEP	1		
STC	STC CCR, Rd	1		
SUB	SUB.B Rs, Rd	1		
	SUB.W Rs, Rd	1		
SUBS	SUBS.W #1, Rd	1		
	SUBS.W #2, Rd	1		
POP	POP Rd	1	1	
PUSH	PUSH Rs	1	1	
SUBX	SUBX.B #xx:8, Rd	1		
	SUBX.B Rs, Rd	1		
XOR	XOR.B #xx:8, Rd	1		
	XOR.B Rs, Rd	1		
XORC	XORC #xx:8, CCR	1		

H'91	SPSR			SPS32	SPS31	SCINV3	SCINV2	SCINV1	SCINV
H'92	CWOSR	—	—	—	—	—	—	—	CWOS
H'93									
H'94									
H'95	ECCSR	OVH	OVL	—	CH2	CUEH	CUEL	CRCH	CRCL
H'96	ECH	ECH7	ECH6	ECH5	ECH4	ECH3	ECH2	ECH1	ECH0
H'97	ECL	ECL7	ECL6	ECL5	ECL4	ECL3	ECL2	ECL1	ECL0
H'98	SMR31	COM31	CHR31	PE31	PM31	STOP31	MP31	CKS311	CKS31
H'99	BRR31	BRR317	BRR316	BRR315	BRR314	BRR313	BRR312	BRR311	BRR31
H'9A	SCR31	TIE31	RIE31	TE31	RE31	MPIE31	TEIE31	CKE31	CKE31
H'9B	TDR31	TDR317	TDR316	TDR315	TDR314	TDR313	TDR312	TDR311	TDR31
H'9C	SSR31	TDRE31	RDRF31	OER31	FER31	PER31	TEND31	MPBR31	MPBT3
H'9D	RDR31	RDR317	RDR316	RDR315	RDR314	RDR313	RDR312	RDR311	RDR31
H'9E									
H'9F									
H'A0									
H'A1									
H'A2									
H'A3									
H'A4									
H'A5									
H'A6									
H'A7									
H'A8	SMR32	COM32	CHR32	PE32	PM32	STOP32	MP32	CKS321	CKS32
H'A9	BRR32	BRR327	BRR326	BRR325	BRR324	BR323	BRR322	BRR321	BRR32
H'AA	SCR32	TIE32	RIE32	TE32	RE32	MPIE32	TEIE32	CKE321	CKE32
H'AB	TDR32	TDR327	TDR326	TDR325	TDR324	TDR323	TDR322	TDR321	TDR32
H'AC	SSR32	TDRE32	RDRF32	OER32	FER32	PER32	TEND32	MPBR32	MPBT3
H'AD	RDR32	RDR327	RDR326	RDR325	RDR324	RDR323	RDR322	RDR321	RDR32
H'AE									
H'AF									
H'B0	TMA	TMA7	TMA6	TMA5	—	TMA3	TMA2	TMA1	TMA0

H'B8	TCFH	TCFH7	TCFH6	TCFH5	TCFH4	TCFH3	TCFH2	TCFH1	TCFH0
H'B9	TCFL	TCFL7	TCFL6	TCFL5	TCFL4	TCFL3	TCFL2	TCFL1	TCFL0
H'BA	OCRFH	OCRFH7	OCRFH6	OCRFH5	OCRFH4	OCRFH3	OCRFH2	OCRFH1	OCRFH(
H'BB	OCRFL	OCRFL7	OCRFL6	OCRFL5	OCRFL4	OCRFL3	OCRFL2	OCRFL1	OCRFL(
H'BC	TMG	OVFH	OVFL	OVIE	IIEGS	CCLR1	CCLR0	CKS1	CKS0
H'BD	ICRGF	ICRGF7	ICRGF6	ICRGF5	ICRGF4	ICRGF3	ICRGF2	ICRGF1	ICRGFC
H'BE	ICRGR	ICRGR7	ICRGR6	ICRGR5	ICRGR4	ICRGR3	ICRGR2	ICRGR1	ICRGRC
H'BF									
H'C0	LPCR	DTS1	DTS0	CMX	SGX	SGS3	SGS2	SGS1	SGS0
H'C1	LCR	—	PSW	ACT	DISP	CKS3	CKS2	CKS1	CKS0
H'C2	LCR2	LCDAB	—	—	SUPS	CDS3	CDS2	CDS1	CDS0
H'C3									
H'C4	ADRRH	ADR9	ADR8	ADR7	ADR6	ADR5	ADR4	ADR3	ADR2
H'C5	ADRRL	ADR1	ADR0	—	—	—	—	—	—
H'C6	AMR	CKS	TRGE	—	—	CH3	CH2	CH1	CH0
H'C7	ADSR	ADSF	—	—	—	—	—	—	—
H'C8	PMR1	IRQ3	IRQ2	IRQ1	IRQ4	TMIG	TMOFH	TMOFL	TMOW
H'C9									
H'CA	PMR3	AEVL	AEVH	WDCKS	NCS	IRQ0	RESO	UD	PWM
H'CB									
H'CC	PMR5	WKP7	WKP6	WKP5	WKP4	WKP3	WKP2	WKP1	WKP0
H'CD									
H'CE									
H'CF									
H'D0	PWCR	—	—	—	—	—	—	PWCR1	PWCR0
H'D1	PWDRU	—	—	PWDRU5	PWDRU4	PWDRU3	PWDRU2	PWDRU1	PWDRU
H'D2	PWDRL	PWDRL7	PWDRL6	PWDRL5	PWDRL4	PWDRL3	PWDRL2	PWDRL1	PWDRL(
H'D3									
H'D4	PDR1	P17	P16	P15	P14	P13	P12	P11	P10

H'DC									
H'DD	PDRA	—	—	—	—	PA3	PA2	PA1	PA0
H'DE	PDRB	PB7	PB6	PB5	PB4	PB3	PB2	PB1	PB0
H'DF									
H'E0	PUCR1	PUCR17	PUCR16	PUCR15	PUCR14	PUCR13	PUCR12	PUCR11	PUCR1
H'E1	PUCR3	PUCR37	PUCR36	PUCR35	PUCR34	PUCR33	PUCR32	PUCR31	PUCR3
H'E2	PUCR5	PUCR57	PUCR56	PUCR55	PUCR54	PUCR53	PUCR52	PUCR51	PUCR5
H'E3	PUCR6	PUCR67	PUCR66	PUCR65	PUCR64	PUCR63	PUCR62	PUCR61	PUCR6
H'E4	PCR1	PCR17	PCR16	PCR15	PCR14	PCR13	PCR12	PCR11	PCR10
H'E5									
H'E6	PCR3	PCR37	PCR36	PCR35	PCR34	PCR33	PCR32	PCR31	PCR30
H'E7	PCR4	—	—	—	—	—	PCR42	PCR41	PCR40
H'E8	PCR5	PCR57	PCR56	PCR55	PCR54	PCR53	PCR52	PCR51	PCR50
H'E9	PCR6	PCR67	PCR66	PCR65	PCR64	PCR63	PCR62	PCR61	PCR60
H'EA	PCR7	PCR77	PCR76	PCR75	PCR74	PCR73	PCR72	PCR71	PCR70
H'EB	PCR8	PCR87	PCR86	PCR85	PCR84	PCR83	PCR82	PCR81	PCR80
H'EC									
H'ED	PCRA	—	—	—	—	PCRA3	PCRA2	PCRA1	PCRA0
H'EE									
H'EF									
H'F0	SYSCR1	SSBY	STS2	STS1	STS0	LSON	—	MA1	MA0
H'F1	SYSCR2	—	—	—	NESEL	DTON	MSON	SA1	SA0
H'F2	IEGR	—	—	—	IEG4	IEG3	IEG2	IEG1	IEG0
H'F3	IENR1	IENTA	—	IENWP	IEN4	IEN3	IEN2	IEN1	IEN0
H'F4	IENR2	IENDT	IENAD	—	IENTG	IENTFH	IENTFL	IENTC	IENEC
H'F5									
H'F6	IRR1	IRRTA	—	—	IRRI4	IRRI3	IRRI2	IRRI1	IRRI0
H'F7	IRRI2	IRRDT	IRRAD	—	IRRTG	IRRTFH	IRRTFL	IRRTC	IRREC
H'F8									

Legend
SCI: Serial Communication Interface



Possible types of access

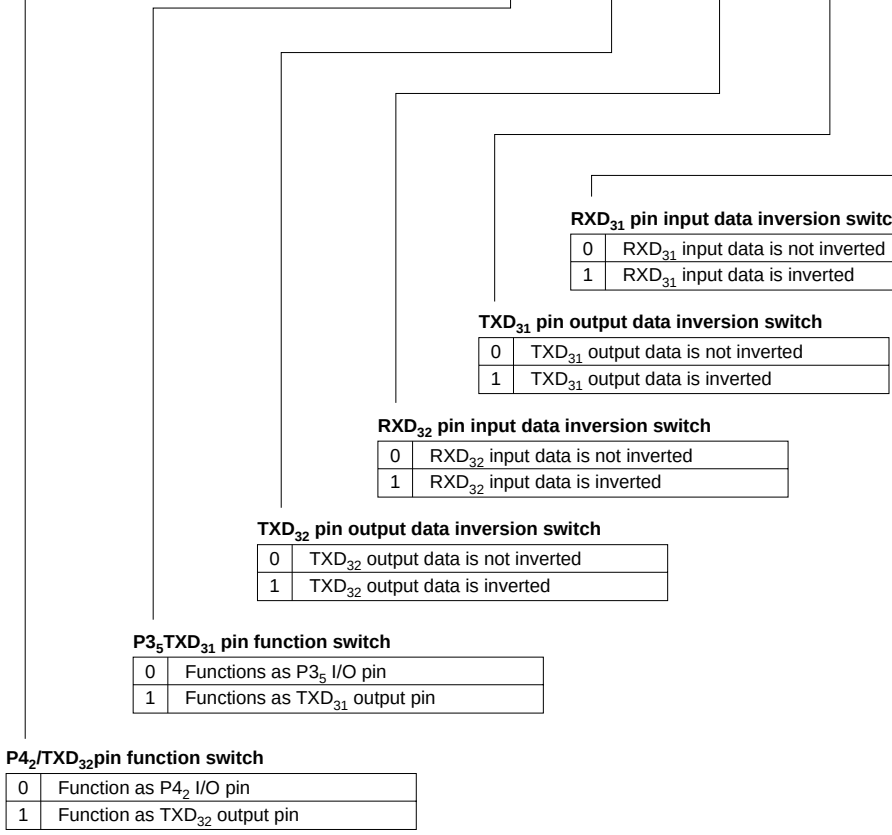
R	Read only
W	Write only
R/W	Read and write

*: Don't care

$\overline{\text{WKPn}}$ edge selected

0	$\overline{\text{WKPn}}$ pin falling edge detected
1	$\overline{\text{WKPn}}$ pin rising edge detected

(n = 0 to 7)



RXD_{31} pin input data inversion switc

0	RXD_{31} input data is not inverted
1	RXD_{31} input data is inverted

TXD_{31} pin output data inversion switch

0	TXD_{31} output data is not inverted
1	TXD_{31} output data is inverted

RXD_{32} pin input data inversion switch

0	RXD_{32} input data is not inverted
1	RXD_{32} input data is inverted

TXD_{32} pin output data inversion switch

0	TXD_{32} output data is not inverted
1	TXD_{32} output data is inverted

$P3_5TXD_{31}$ pin function switch

0	Functions as $P3_5$ I/O pin
1	Functions as TXD_{31} output pin

$P4_2/TXD_{32}$pin function switch

0	Function as $P4_2$ I/O pin
1	Function as TXD_{32} output pin

TMOW pin clock select

0	Clock output from TMA is output
1	$ø_W$ is output

Counter reset co

0	ECL is reset
1	ECL reset is and count-u is enabled

Counter reset control H

0	ECH is reset
1	ECH reset is cleared count-up function is

Count-up enable L

0	ECL event clock input is ECL value is held
1	ECL event clock input is

Count-up enable H

0	ECH event clock input is dis ECH value is held
1	ECH event clock input is en

Channel select

0	ECH and ECL are used together as channel 16-bit event counter
1	ECH and ECL are used as two indep 8-bit event counter channels

Counter overflow L

0	ECL has not overflowed
1	ECL has overflowed

Counter overflow H

0	ECH has not overflowed
1	ECH has overflowed

Bit	7	6	5	4	3	2	1	
	ECL7	ECL6	ECL5	ECL4	ECL3	ECL2	ECL1	
Initial value	0	0	0	0	0	0	0	
Read/Write	R	R	R	R	R	R	R	

Clock se

0	0	ø
0	1	ø
1	0	ø
1	1	ø

Multiprocessor mode

0	Multiprocessor comm function disabled
1	Multiprocessor comm function enabled

Stop bit length

0	1 stop bit
1	2 stop bits

Parity mode

0	Even parity
1	Odd parity

Parity enable

0	Parity bit addition and checking disabled
1	Parity bit addition and checking enabled

Character length

0	8-bit data/5-bit data
1	7-bit data/5-bit data

Communication mode

0	Asynchronous mode
1	Synchronous mode

Clock enable

Bit 1	Bit 0	Description		
CKE311	CKE310	Communication Mode	Clock Source	SCK_3 Pin Function
0	0	Asynchronous	Internal clock	I/O port
		Synchronous	Internal clock	Serial clock output
0	1	Asynchronous	Internal clock	Clock output
		Synchronous	Reserved (Do not specify this combination	
1	0	Asynchronous	External clock	Clock input
		Synchronous	External clock	Serial clock input
1	1	Asynchronous	Reserved (Do not specify this combination	
		Synchronous	Reserved (Do not specify this combination	

Transmit end interrupt enable

0	Transmit end interrupt request (TEI) disabled
1	Transmit end interrupt request (TEI) enabled

Multiprocessor interrupt enable

0	Multiprocessor interrupt request disabled (normal receive operation) [Clearing conditions] When data is received in which the multiprocessor bit is set to 1
1	Multiprocessor interrupt request enabled The receive interrupt request (RXI), receive error interrupt request (ERI), and setting of th RDRF, FER, and OER flags in the serial status register (SSR), are disabled until data with the multiprocessor bit set to 1 is received.

Receive enable

0	Receive operation disabled (RXD pin is I/O port)
1	Receive operation enabled (RXD pin is receive data pin)

Transmit enable

0	Transmit operation disabled (TXD pin is transmit data pin)
1	Transmit operation enabled (TXD pin is transmit data pin)

Receive interrupt enable

0	Receive data full interrupt request (RXI) and receive error interrupt request (ERI) disabled
1	Receive data full interrupt request (RXI) and receive error interrupt request (ERI) enabled

Transmit interrupt enable

0	Transmit data empty interrupt request (TXI) disabled
1	Transmit data empty interrupt request (TXI) enabled

Multiprocessor bit transfer

0	A 0 multiprocessor bit is transmitted
1	A 1 multiprocessor bit is transmitted

Multiprocessor bit receive

0	Data in which the multiprocessor bit is 0 has been received
1	Data in which the multiprocessor bit is 1 has been received

Transmit end

0	Transmission in progress [Clearing conditions] • After reading TDRE31 = 1, cleared by writing 0 to TDRE • When data is written to TDR31 by an instruction
1	Transmission ended [Setting conditions] • When bit TE in serial control register 31 (SCR31) is cleared to 0 • When bit TDRE31 is set to 1 when the last bit of a transmit character is sent

Parity error

0	Reception in progress or completed normally [Clearing conditions] After reading PER31 = 1, cleared by writing 0 to PER31
1	A parity error has occurred during reception [Setting conditions] When the number of 1 bits in the receive data plus parity bit does not match the parit designated by the parity mode bit (PM31) in the serial mode register (SMR31)

Framing error

0	Reception in progress or completed normally [Clearing conditions] After reading FER31 = 1, cleared by writing 0 to FER31
1	A framing error has occurred during reception [Setting conditions] When the stop bit at the end of the receive data is checked for a value of 1 at completion reception, and the stop bit is 0

Overrun error

0	Reception in progress or completed [Clearing conditions] After reading OER31 = 1, cleared by writing 0 to OER31
1	An overrun error has occurred during reception [Setting conditions] When the next serial reception is completed with RDRF31 set to 1

Receive data register full

0	There is no receive data in RDR31 [Clearing conditions] • After reading RDRF31 = 1, cleared by writing 0 to RDRF31 • When RDR31 data is read by an instruction
1	There is receive data in RDR31 [Setting conditions] When reception ends normally and receive data is transferred from RSR31 to RDR31

Transmit data register empty

0	Transmit data written in TDR31 has not been transferred to TSR31 [Clearing conditions] • After reading TDRE31 = 1, cleared by writing 0 to TDRE31 • When data is written to TDR31 by an instruction
1	Transmit data has not been written to TDR31, or transmit data written in TDR31 has been transferred to TSR31 [Setting conditions] • When bit TE in serial control register 31 (SCR31) is cleared to 0 • When data is transferred from TDR31 to TSR31

Note: * Only a write of 0 for flag clearing is possible.

Clock se

0	0	ø
0	1	ø
1	0	ø
1	1	ø

Multiprocessor mode

0	Multiprocessor comm function disabled
1	Multiprocessor comm function enabled

Stop bit length

0	1 stop bit
1	2 stop bits

Parity mode

0	Even parity
1	Odd parity

Parity enable

0	Parity bit addition and checking disabled
1	Parity bit addition and checking enabled

Character length

0	8-bit data/5-bit data
1	7-bit data/5-bit data

Communication mode

0	Asynchronous mode
1	Synchronous mode

Clock enable

Bit 1	Bit 0	Description		
CKE321	CKE320	Communication Mode	Clock Source	SCK_3 Pin Function
0	0	Asynchronous	Internal clock	I/O port
		Synchronous	Internal clock	Serial clock output
0	1	Asynchronous	Internal clock	Clock output
		Synchronous	Reserved (Do not specify this combination	
1	0	Asynchronous	External clock	Clock input
		Synchronous	External clock	Serial clock input
1	1	Asynchronous	Reserved (Do not specify this combination	
		Synchronous	Reserved (Do not specify this combination	

Transmit end interrupt enable

0	Transmit end interrupt request (TEI) disabled
1	Transmit end interrupt request (TEI) enabled

Multiprocessor interrupt enable

0	Multiprocessor interrupt request disabled (normal receive operation) [Clearing conditions] When data is received in which the multiprocessor bit is set to 1
1	Multiprocessor interrupt request enabled The receive interrupt request (RXI), receive error interrupt request (ERI), and setting of th RDRF, FER, and OER flags in the serial status register (SSR), are disabled until data with the multiprocessor bit set to 1 is received.

Receive enable

0	Receive operation disabled (RXD pin is I/O port)
1	Receive operation enabled (RXD pin is receive data pin)

Transmit enable

0	Transmit operation disabled (TXD pin is transmit data pin)
1	Transmit operation enabled (TXD pin is transmit data pin)

Receive interrupt enable

0	Receive data full interrupt request (RXI) and receive error interrupt request (ERI) disabled
1	Receive data full interrupt request (RXI) and receive error interrupt request (ERI) enabled

Transmit interrupt enable

0	Transmit data empty interrupt request (TXI) disabled
1	Transmit data empty interrupt request (TXI) enabled

Multiprocessor bit transfer

0	A 0 multiprocessor bit is transmitted
1	A 1 multiprocessor bit is transmitted

Multiprocessor bit receive

0	Data in which the multiprocessor bit is 0 has been received
1	Data in which the multiprocessor bit is 1 has been received

Transmit end

0	Transmission in progress [Clearing conditions] • After reading TDRE32 = 1, cleared by writing 0 to TDRE32 • When data is written to TDR32 by an instruction
1	Transmission ended [Setting conditions] • When bit TE in serial control register 32 (SCR32) is cleared to 0 • When bit TDRE32 is set to 1 when the last bit of a transmit character is sent

Parity error

0	Reception in progress or completed normally [Clearing conditions] After reading PER32 = 1, cleared by writing 0 to PER32
1	A parity error has occurred during reception [Setting conditions] When the number of 1 bits in the receive data plus parity bit does not match the parit designated by the parity mode bit (PM32) in the serial mode register (SMR32)

Framing error

0	Reception in progress or completed normally [Clearing conditions] After reading FER32 = 1, cleared by writing 0 to FER32
1	A framing error has occurred during reception [Setting conditions] When the stop bit at the end of the receive data is checked for a value of 1 at completion reception, and the stop bit is 0

Overrun error

0	Reception in progress or completed [Clearing conditions] After reading OER32 = 1, cleared by writing 0 to OER32
1	An overrun error has occurred during reception [Setting conditions] When the next serial reception is completed with RDRF32 set to 1

Receive data register full

0	There is no receive data in RDR32 [Clearing conditions] • After reading RDRF32 = 1, cleared by writing 0 to RDRF32 • When RDR32 data is read by an instruction
1	There is receive data in RDR32 [Setting conditions] When reception ends normally and receive data is transferred from RSR32 to RDR32

Transmit data register empty

0	Transmit data written in TDR32 has not been transferred to TSR32 [Clearing conditions] • After reading TDRE32 = 1, cleared by writing 0 to TDRE32 • When data is written to TDR32 by an instruction
1	Transmit data has not been written to TDR32, or transmit data written in TDR32 has been transferred to TSR32 [Setting conditions] • When bit TE32 in serial control register 32 (SCR32) is cleared to 0 • When data is transferred from TDR32 to TSR32

Note: * Only a write of 0 for flag clearing is possible.

Bit	7	6	5	4	3	2	1	
	TMA7	TMA6	TMA5	—	TMA3	TMA2	TMA1	
Initial value	0	0	0	1	0	0	0	
Read/Write	R/W	R/W	R/W	—	R/W	R/W	R/W	

Clock output select

0	0	0	ø/32
		1	ø/16
	1	0	ø/8
		1	ø/4
1	0	0	$ø_W$/32
		1	$ø_W$/16
	1	0	$ø_W$/8
		1	$ø_W$/4

Internal clock select

TMA3	TMA2	TMA1	TMA0	Prescaler and Divider Ratio or Overflow Period		F
0	0	0	0	PSS	ø/8192	I
			1	PSS	ø/4096	t
		1	0	PSS	ø/2048	
			1	PSS	ø/512	
	1	0	0	PSS	ø/256	
			1	PSS	ø/128	
		1	0	PSS	ø/32	
			1	PSS	ø/8	
1	0	0	0	PSW	1 s	T
			1	PSW	0.5 s	b
		1	0	PSW	0.25 s	(
			1	PSW	0.03125 s	u
	1	0	0	PSW and TCA are reset		3
			1			
		1	0			
			1			

Watchdog timer reset

0	[Clearing conditions] • Reset by $\overline{\text{RES}}$ pin • When TCSRWE = 1, and 0 is written in both B0WI and WR
1	[Setting condition] When TCW overflows and a reset signal is generated

Bit 0 write inhibit

0	Bit 0 is write-enabled
1	Bit 0 is write-protected

Watchdog timer on

0	Watchdog timer operation is disabled
1	Watchdog timer operation is enabled

Bit 2 write inhibit

0	Bit 2 is write-enabled
1	Bit 2 is write-protected

Timer control/status register W write enable

0	Data cannot be written to bits 2 and 0
1	Data can be written to bits 2 and 0

Bit 4 write inhibit

0	Bit 4 is write-enabled
1	Bit 4 is write-protected

Timer counter W write enable

0	Data cannot be written to TCW
1	Data can be written to TCW

Bit 6 write inhibit

0	Bit 6 is write-enabled
1	Bit 6 is write-protected

Note: * Write is permitted only under certain conditions.

TMC—Timer mode register C **H'B4**

Bit	7	6	5	4	3	2	1
	TMC7	TMC6	TMC5	—	—	TMC2	TMC1
Initial value	0	0	0	1	1	0	0
Read/Write	R/W	R/W	R/W	—	—	R/W	R/W

Clock select

0	0	0	Internal clock: ø/8192
0	0	1	Internal clock: ø/2048
0	1	0	Internal clock: ø/512
0	1	1	Internal clock: ø/64
1	0	0	Internal clock: ø/16
1	0	1	Internal clock: ø/4
1	1	0	Internal clock: øw/4
1	1	1	External event (TMIC): C on rising or falling edge

Counter up/down control

0	0	TCC is an up-counter
0	1	TCC is a down-counter
1	*	Hardware control of TCC up/down operation by UD pin UD pin input high: Down-counter UD pin input low: Up-counter

Auto-reload function select

0	Interval timer function selected
1	Auto-reload function selected

* :

TLC—Timer load register C **H'B5**

Bit	7	6	5	4	3	2	1	
	TLC7	TLC6	TLC5	TLC4	TLC3	TLC2	TLC1	
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	

Reload value

Clock select L

0	*	*	Counting on external ev rising/falling edge
1	0	0	Internal clock ø/32
1	0	1	Internal clock ø/16
1	1	0	Internal clock ø/4
1	1	1	Internal clock øw/4

Toggle output level L

0	Low level
1	High level

Clock select H

0	*	*	16-bit mode, counting on TCFL overflow signal
1	0	0	Internal clock ø/32
1	0	1	Internal clock ø/16
1	1	0	Internal clock ø/4
1	1	1	Internal clock øw/4

* : Don’t care

Toggle output level H

0	Low level
1	High level

Counter clear L

0	TCFL clearing by compare match is disabled
1	TCFL clearing by compare match is enabled

Timer overflow interrupt enable L

0	TCFL overflow interrupt request is disabled
1	TCFL overflow interrupt request is enabled

Compare match flag L

0	Clearing conditions: After reading CMFL = 1, cleared by writing 0 to CMFL
1	Setting conditions: Set when the TCFL value matches the OCRFL value

Timer overflow flag L

0	Clearing conditions: After reading OVFL = 1, cleared by writing 0 to OVFL
1	Setting conditions: Set when TCFL overflows from H'FF to H'00

Counter clear H

0	16-bit mode: TCF clearing by compare match is disabled 8-bit mode: TCFH clearing by compare match is disabled
1	16-bit mode: TCF clearing by compare match is enabled 8-bit mode: TCFH clearing by compare match is enabled

Timer overflow interrupt enable H

0	TCFH overflow interrupt request is disabled
1	TCFH overflow interrupt request is enabled

Compare match flag H

0	Clearing conditions: After reading CMFH = 1, cleared by writing 0 to CMFH
1	Setting conditions: Set when the TCFH value matches the OCRFH value

Timer overflow flag H

0	Clearing conditions: After reading OVFH = 1, cleared by writing 0 to OVFH
1	Setting conditions: Set when TCFH overflows from H'FF to H'00

Note: * Bits 7, 6, 3, and 2 can only be written with 0, for flag clearing.

TCFL—8-bit timer counter FL **H'B9**

Bit	7	6	5	4	3	2	1
	TCFL7	TCFL6	TCFL5	TCFL4	TCFL3	TCFL2	TCFL1
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

Count value

OCRFH—Output compare register FH **H'BA**

Bit	7	6	5	4	3	2	1
	OCRFH7	OCRFH6	OCRFH5	OCRFH4	OCRFH3	OCRFH2	OCRFH1
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

OCRFL—Output compare register FL **H'BB**

Bit	7	6	5	4	3	2	1
	OCRFL7	OCRFL6	OCRFL5	OCRFL4	OCRFL3	OCRFL2	OCRFL1
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

0	0	
0	1	Internal clock: counti
1	0	Internal clock: counti
1	1	Internal clock: count

Counter clear

0	0	TCG clearing is disabled
0	1	TCG cleared by falling edge of input capture input sig
1	0	TCG cleared by rising edge of input capture input sig
1	1	TCG cleared by both edges of input capture input sig

Input capture interrupt edge select

0	Interrupt generated on rising edge of input capture input signal
1	Interrupt generated on falling edge of input capture input signal

Timer overflow interrupt enable

0	TCG overflow interrupt request is disabled
1	TCG overflow interrupt request is enabled

Timer overflow flag L

0	Clearing conditions: After reading OVFL = 1, cleared by writing 0 to OVFL
1	Setting conditions: Set when TCG overflows from H'FF to H'00

Timer overflow flag H

0	Clearing conditions: After reading OVFH = 1, cleared by writing 0 to OVFH
1	Setting conditions: Set when TCG overflows from H'FF to H'00

Note: * Bits 7 and 6 can only be written with 0, for flag clearing.

Bit	7	6	5	4	3	2	1	
	ICRGR7	ICRGR6	ICRGR5	ICRGR4	ICRGR3	ICRGR2	ICRGR1	
Initial value	0	0	0	0	0	0	0	
Read/Write	R	R	R	R	R	R	R	

Bit 4	Bit 3	Bit 2	Bit 1	Bit 0	Function of Pins SEG_{32} to SEG_1							
SGX	SGS3	SGS2	SGS1	SGS0	SEG_{32} to SEG_{29}	SEG_{28} to SEG_{25}	SEG_{24} to SEG_{21}	SEG_{20} to SEG_{17}	SEG_{16} to SEG_{13}	SEG_{12} to SEG_9	SEG_8 to SEG_5	SEG to SE
0	0	0	0	0	Port	Port	Port	Port	Port	Port	Port	Por
	0	0	0	1	Port	Port	Port	Port	Port	Port	Port	Por
	0	0	1	*	SEG	SEG	Port	Port	Port	Port	Port	Por
	0	1	0	*	SEG	SEG	SEG	SEG	Port	Port	Port	Por
	0	1	1	*	SEG	SEG	SEG	SEG	SEG	SEG	Port	Por
	1	*	*	*	SEG	SEG	SEG	SEG	SEG	SEG	SEG	SEG
1	0	0	0	0	Port*	Port	Port	Port	Port	Port	Port	Por
	*	*	*	*	Use prohibited							

Note: * SEG32 to SEG29 are external expansion pins.

Bit 4	Description	
SGX		
0	Pins $SEG_{32}$to SEG_{29}*	(Initial value)
1	Pins CL_1, CL_2, DO, M	

Note: * These pins function as ports when the setting of SGS3 to SGS0 is 0000 or 0001.

Duty select, common function select

Bit 7	Bit 6	Bit 5	Duty Cycle	Common Drivers	Notes
DTS1	DTS0	CMX			
0	0	0	Static	COM_1	
		1		COM_4 to COM_1	COM_4 to COM_2 output the same waveform as COM_1
0	1	0	1/2 duty	COM_2 to COM_1	
		1		COM_4 to COM_1	COM_4 outputs the same waveform as COM_3 and COM_2 outputs the same waveform as COM_1
1	0	0	1/3 duty	COM_3 to COM_1	
		1		COM_4 to COM_1	COM_4 outputs a non-selected waveform
1	1	0	1/4 duty	COM_4 to COM_1	—
		1			

Frame frequency select

Bit 3	Bit 2	Bit 1	Bit 1	Op
CKS3	CKS2	CKS1	CKS0	
0	*	0	0	
0	*	0	1	
0	*	1	*	
1	0	0	0	
1	0	0	1	
1	0	1	0	
1	0	1	1	
1	1	0	0	
1	1	0	1	
1	1	1	0	
1	1	1	1	

Display data control

0	Blank data is displayed
1	LCD RAM data is displayed

Display function activate

0	LCD controller/driver operation halted
1	LCD controller/driver operates

LCD drive power supply on/off control

0	LCD drive power supply off
1	LCD drive power supply on

Charge/discharge pulse duty cyc

Bit 3	Bit 2	Bit 1	Bit 1	Duty
CDS3	CDS2	CDS1	CDS0	
0	0	0	0	
0	0	0	1	
0	0	1	0	
0	0	1	1	
0	1	0	0	
0	1	0	1	
0	1	1	0	
0	1	1	1	
1	0	*	*	
1	1	*	*	

*

5 V regulator control

0	5 V regulator halted
1	5 V regulator operates

Applies to the H8/3867 Series

A waveform/B waveform switching control

0	Drive using A waveform
1	Drive using B waveform

Channel select

Bit 3	Bit 2	Bit 1	Bit 0	
CH3	CH2	CH1	CH0	Analog Input C
0	0	*	*	No channel se
	1	0	0	AN_0
			1	AN_1
		1	0	AN_2
			1	AN_3
1	0	0	0	AN_4
			1	AN_5
		1	0	AN_6
			1	AN_7

* : Do

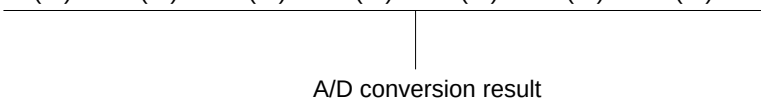
External trigger select

0	Disables start of A/D conversion by external trigger
1	Enables start of A/D conversion by rising or falling edge of external trigger at pin $\overline{\text{ADTRG}}$

Clock select

Bit 7		Conversion Time	
CKS	Conversion Period	ø = 1 MHz	ø = 2 MHz
0	62/ø	62 µs	31 µs
1	31/ø	31 µs	15.5µs*

Note: * Operation is not guaranteed with a conversion time of less than 15.5 µs Select a setting that gives a conversion time of at least 15.5 µs.



ADRRL

Bit	7	6	5	4	3	2	1	
	ADR1	ADR0	—	—	—	—	—	
Initial value	Not fixed	Not fixed	—	—	—	—	—	
Read/Write	R	R	—	—	—	—	—	

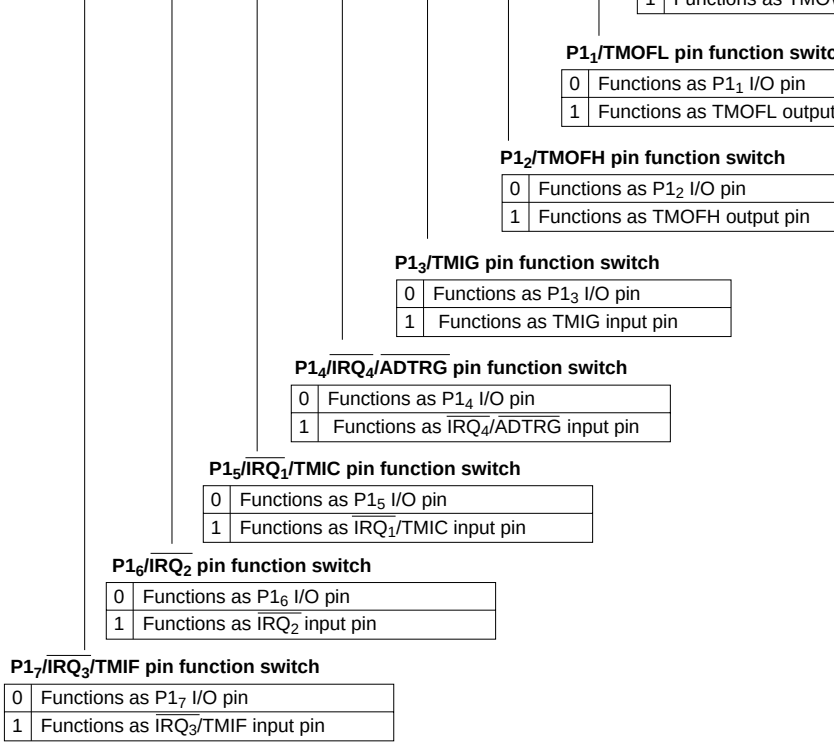
A/D conversion result

ADSR—A/D start register **H'C7** **A/D**

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

A/D status flag

0	Read	Indicates completion of A/D conversion
	Write	Stops A/D conversion
1	Read	Indicates A/D conversion in progress
	Write	Starts A/D conversion


P1$_1$/TMOFL pin function switch
0 Functions as P1$_1$ I/O pin
1 Functions as TMOFL output
P1$_2$/TMOFH pin function switch
0 Functions as P1$_2$ I/O pin
1 Functions as TMOFH output pin
P1$_3$/TMIG pin function switch
0 Functions as P1$_3$ I/O pin
1 Functions as TMIG input pin
P1$_4$/$\overline{IRQ_4}$/$\overline{ADTRG}$ pin function switch
0 Functions as P1$_4$ I/O pin
1 Functions as $\overline{IRQ_4}$/$\overline{ADTRG}$ input pin
P1$_5$/$\overline{IRQ_1}$/TMIC pin function switch
0 Functions as P1$_5$ I/O pin
1 Functions as $\overline{IRQ_1}$/TMIC input pin
P1$_6$/$\overline{IRQ_2}$ pin function switch
0 Functions as P1$_6$ I/O pin
1 Functions as $\overline{IRQ_2}$ input pin
P1$_7$/$\overline{IRQ_3}$/TMIF pin function switch
0 Functions as P1$_7$ I/O pin
1 Functions as $\overline{IRQ_3}$/TMIF input pin

1	Functions as PW

$P3_1$/UD pin function switch

0	Functions as $P3_1$ I/O pin
1	Functions as UD input pin

$P3_2$/RESO pin function switch

0	Functions as $P3_2$ I/O pin
1	Functions as $\overline{RESO}$ I/O pin

$P4_3$/$\overline{IRQ0}$ pin function switch

0	Functions as $P4_3$ I/O pin
1	Functions as $\overline{IRQ_0}$ input pin

TMIG noise canceler select

0	Noise cancellation function not used
1	Noise cancellation function used

Watchdog timer switch

0	ø8192
1	øw/4

$P3_6$/AEVH pin function switch

0	Functions as $P3_6$ I/O pin
1	Functions as AEVH input pin

$P3_7$/AEVL pin function switch

0	Functions as $P3_7$ I/O pin
1	Functions as AEVL input pin

PWCR—PWM control register **H'D0**

Bit	7	6	5	4	3	2	1	
	—	—	—	—	—	—	PWCR1	
Initial value	1	1	1	1	1	1	0	
Read/Write	—	—	—	—	—	—	W	

Clock select

0	The input clock is ø/2 (tø* = 2/ø) The conversion period is 16,384/ø, with a minimum modulation width of
	The input clock is ø/4 (tø* = 4/ø) The conversion period is 32,768/ø, with a minimum modulation width of
1	The input clock is ø/8 (tø* = 8/ø) The conversion period is 65,536/ø, with a minimum modulation width of
	The input clock is ø/16 (tø* = 16/ø) The conversion period is 131,072/ø, with a minimum modulation width

Note: * tø: Period of PWM input clock

PWDRL—PWM data register L **H'D2** **14**

Bit	7	6	5	4	3	2	1	
	PWDRL7	PWDRL6	PWDRL5	PWDRL4	PWDRL3	PWDRL2	PWDRL1	P
Initial value	0	0	0	0	0	0	0	
Read/Write	W	W	W	W	W	W	W	

Lower 8 bits of data for generating PWM waveform

PDR1—Port data register 1 **H'D4**

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	

PDR3—Port data register 3 **H'D6**

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	

PDR4—Port data register 4 **H'D7**

Bit	7	6	5	4	3	2	1	
	—	—	—	—	$P4_3$	$P4_2$	$P4_1$	
Initial value	1	1	1	1	1	0	0	
Read/Write	—	—	—	—	R	R/W	R/W	

Bit	7	6	5	4	3	2	1
	$P6_7$	$P6_6$	$P6_5$	$P6_4$	$P6_3$	$P6_2$	$P6_1$
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

PDR7—Port data register 7 **H'DA**

Bit	7	6	5	4	3	2	1
	$P7_7$	$P7_6$	$P7_5$	$P7_4$	$P7_3$	$P7_2$	$P7_1$
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

PDR8—Port data register 8 **H'DB**

Bit	7	6	5	4	3	2	1
	$P8_7$	$P8_6$	$P8_5$	$P8_4$	$P8_3$	$P8_2$	$P8_1$
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

PDRA—Port data register A **H'DD**

Bit	7	6	5	4	3	2	1
	—	—	—	—	PA_3	PA_2	PA_1
Initial value	1	1	1	1	0	0	0
Read/Write	—	—	—	—	R/W	R/W	R/W

Bit	7	6	5	4	3	2	1	
	$PUCR1_7$	$PUCR1_6$	$PUCR1_5$	$PUCR1_4$	$PUCR1_3$	$PUCR1_2$	$PUCR1_1$	P
Initial value	0	0	0	0	0	0	0	
Read/Write	R/W	R/W	R/W	R/W	R/W	R/W	R/W	

PUCR3—Port pull-up control register 3 **H'E1**

Bit	7	6	5	4	3	2	1	
	$PUCR3_7$	$PUCR3_6$	$PUCR3_5$	$PUCR3_4$	$PUCR3_3$	$PUCR3_2$	$PUCR3_1$	P
Initial value	0	0	0	0	0	0	0	
Read/Write	R/W	R/W	R/W	R/W	R/W	R/W	R/W	

PUCR5—Port pull-up control register 5 **H'E2**

Bit	7	6	5	4	3	2	1	
	$PUCR5_7$	$PUCR5_6$	$PUCR5_5$	$PUCR5_4$	$PUCR5_3$	$PUCR5_2$	$PUCR5_1$	P
Initial value	0	0	0	0	0	0	0	
Read/Write	R/W	R/W	R/W	R/W	R/W	R/W	R/W	

PUCR6—Port pull-up control register 6 **H'E3**

Bit	7	6	5	4	3	2	1	
	$PUCR6_7$	$PUCR6_6$	$PUCR6_5$	$PUCR6_4$	$PUCR6_3$	$PUCR6_2$	$PUCR6_1$	P
Initial value	0	0	0	0	0	0	0	
Read/Write	R/W	R/W	R/W	R/W	R/W	R/W	R/W	

1	Output pin

PCR3—Port control register 3 **H'E6**

Bit	7	6	5	4	3	2	1
	$PCR3_7$	$PCR3_6$	$PCR3_5$	$PCR3_4$	$PCR3_3$	$PCR3_2$	$PCR3_1$
Initial value	0	0	0	0	0	0	0
Read/Write	W	W	W	W	W	W	W

Port 3 input/output select

0	Input pin
1	Output pin

PCR4—Port control register 4 **H'E7**

Bit	7	6	5	4	3	2	1
	—	—	—	—	—	$PCR4_2$	$PCR4_1$
Initial value	1	1	1	1	1	0	0
Read/Write	—	—	—	—	—	W	W

Port 4 input/output se

0	Input pin
1	Output pin

1	Output pin

PCR6—Port control register 6 **H'E9**

Bit	7	6	5	4	3	2	1	
	$PCR6_7$	$PCR6_6$	$PCR6_5$	$PCR6_4$	$PCR6_3$	$PCR6_2$	$PCR6_1$	P
Initial value	0	0	0	0	0	0	0	
Read/Write	W	W	W	W	W	W	W	

Port 6 input/output select

0	Input pin
1	Output pin

PCR7—Port control register 7 **H'EA**

Bit	7	6	5	4	3	2	1	
	$PCR7_7$	$PCR7_6$	$PCR7_5$	$PCR7_4$	$PCR7_3$	$PCR7_2$	$PCR7_1$	P
Initial value	0	0	0	0	0	0	0	
Read/Write	W	W	W	W	W	W	W	

Port 7 input/output select

0	Input pin
1	Output pin

1	Output pin

PCRA—Port control register A H'ED

Bit	7	6	5	4	3	2	1
	—	—	—	—	$PCRA_3$	$PCRA_2$	$PCRA_1$
Initial value	0	0	0	0	0	0	0
Read/Write	—	—	—	—	R/W	R/W	R/W

Port A input/output select

0	Input pin
1	Output pin

mode clock select

0	0	ϕ_{OSC}/16
	1	ϕ_{OSC}/32
1	0	ϕ_{OSC}/64
	1	ϕ_{OSC}/128

Low speed on flag

0	The CPU operates on the system clo
1	The CPU operates on the subclock (

Standby timer select 2 to 0

0	0	0	Wait time = 8,192 states
		1	Wait time = 16,384 states
	1	0	Wait time = 32,768 states
		1	Wait time = 65,536 states
1	0	0	Wait time = 131,072 states
		1	Wait time = 2 states
	1	0	Wait time = 8 states
		1	Wait time = 16 states

Software standby

0	• When a SLEEP instruction is executed in active mode, a transit made to sleep mode • When a SLEEP instruction is executed in subactive mode, a tra is made to subsleep mode
1	• When a SLEEP instruction is executed in active mode, a transit made to standby mode or watch mode • When a SLEEP instruction is executed in subactive mode, a tra is made to watch mode

Subactive mode clock select

0	0	$ø_W/8$
	1	$ø_W/4$
1	*	$ø_W/2$

*: Don't ca

Medium speed on flag

0	Operates in active (high-speed) mode
1	Operates in active (medium-speed) mode

Direct transfer on flag

0	• When a SLEEP instruction is executed in active mode, a transition is made to standby mode, watch mode, or sleep mode • When a SLEEP instruction is executed in subactive mode, a transition is made to watch mode or subsleep mode
1	• When a SLEEP instruction is executed in active (high-speed) mode, a dire transition is made to active (medium-speed) mode if SSBY = 0, MSON = 1 LSON = 0, or to subactive mode if SSBY = 1, TMA3 = 1, and LSON = 1 • When a SLEEP instruction is executed in active (medium-speed) mode, a transition is made to active (high-speed) mode if SSBY = 0, MSON = 0, an LSON = 0, or to subactive mode if SSBY = 1, TMA3 = 1, and LSON = 1 • When a SLEEP instruction is executed in subactive mode, a direct transition is made to active (high-speed) mode if SSBY = 1, TMA3 = 1, LS and MSON = 0, or to active (medium-speed) mode if SSBY = 1, TMA3 = 1 LSON = 0, and MSON = 1

Noise elimination sampling frequency select

0	Sampling rate is $ø_{OSC}/16$
1	Sampling rate is $ø_{OSC}/4$

$\overline{IRQ_0}$ edge select

0	Falling edge of $\overline{IRQ_0}$ pin input is d
1	Rising edge of $\overline{IRQ_0}$ pin input is de

$\overline{IRQ_1}$ edge select

0	Falling edge of $\overline{IRQ_1}$, TMIC pin input is detec
1	Rising edge of $\overline{IRQ_1}$, TMIC pin input is detec

$\overline{IRQ_2}$ edge select

0	Falling edge of $\overline{IRQ_2}$ pin input is detected
1	Rising edge of $\overline{IRQ_2}$ pin input is detected

$\overline{IRQ_3}$ edge select

0	Falling edge of $\overline{IRQ_3}$, TMIF pin input is detected
1	Rising edge of $\overline{IRQ_3}$, TMIF pin input is detected

$\overline{IRQ_4}$ edge select

0	Falling edge of $\overline{IRQ_4}$ pin and $\overline{ADTRG}$ pin is detected
1	3. Rising edge of $\overline{IRQ_4}$ pin and $\overline{ADTRG}$ pin is detected

$\overline{IRQ_4}$ to $\overline{IRQ_0}$ interrupt enable

0	Disables $\overline{IRQ_4}$ to $\overline{IRQ_0}$ interrupt reque
1	Enables $\overline{IRQ_4}$ to $\overline{IRQ_0}$ interrupt reque

Wakeup interrupt enable

0	Disables $\overline{WKP_7}$ to $\overline{WKP_0}$ interrupt requests
1	Enables $\overline{WKP_7}$ to $\overline{WKP_0}$ interrupt requests

Timer A interrupt enable

0	Disables timer A interrupt requests
1	Enables timer A interrupt requests

1	Enables asynchronous event interrupt requests

Timer C interrupt enable

0	Disables timer C interrupt requests
1	Enables timer C interrupt requests

Timer FL interrupt enable

0	Disables timer FL interrupt requests
1	Enables timer FL interrupt requests

Timer FH interrupt enable

0	Disables timer FH interrupt requests
1	Enables timer FH interrupt requests

Timer G interrupt enable

0	Disables timer G interrupt requests
1	Enables timer G interrupt requests

A/D converter interrupt enable

0	Disables A/D converter interrupt requests
1	Enables A/D converter interrupt requests

Direct transition interrupt enable

0	Disables direct transition interrupt requests
1	Enables direct transition interrupt requests

0	Clearing conditions: When IRRIn = 1, it is cleared by writing
1	Setting conditions: When pin IRQn is designated for interr input and the designated signal edge is

(r

Timer A interrupt request flag

0	Clearing conditions: When IRRTA = 1, it is cleared by writing 0
1	Setting conditions: When the timer A counter value overflows (rom H'FF to H'00)

Note: * Bits 7 and 4 to 0 can only be written with 0, for flag clearing.

Asynchronous event counter interrupt request flag

0	Clearing conditions: When IRREC = 1, it is cleared by writing 0
1	Setting conditions: When the asynchronous event counter value overflows

Timer C interrupt request flag

0	Clearing conditions: When IRRTC = 1, it is cleared by writing 0
1	Setting conditions: When the timer C counter value overflows (from H'FF to H'00) or underflows (from H'00 to H'FF)

Timer FL interrupt request flag

0	Clearing conditions: When IRRTFL = 1, it is cleared by writing 0
1	Setting conditions: When counter FL and output compare register FL match in 8-bit timer mode

Timer FH interrupt request flag

0	Clearing conditions: When IRRTFH = 1, it is cleared by writing 0
1	Setting conditions: When counter FH and output compare register FH match in 8-bit timer mode, or when 16-bit counters FL and FH and output compare registers FL and FH match in 16-bit timer mode

Timer G interrupt request flag

0	Clearing conditions: When IRRTG = 1, it is cleared by writing 0
1	Setting conditions: When the TMIG pin is designated for TMIG input and the designated signal edge is input

A/D converter interrupt request flag

0	Clearing conditions: When IRRAD = 1, it is cleared by writing 0
1	Setting conditions: When the A/D converter completes conversion and ADSF is reset

Direct transition interrupt request flag

0	Clearing conditions: When IRRDT = 1, it is cleared by writing 0
1	Setting conditions: When a SLEEP instruction is executed while DTON is set to 1, and a direct transition is made

Note: * Bits 7, 6 and 4 to 0 can only be written with 0, for flag clearing.

0	Clearing conditions: When IWPFn = 1, it is cleared by writing 0
1	Setting conditions: When pin $\overline{\text{WKPn}}$ is designated for wakeup input an falling edge is input at that pin

(n =

Note: * All bits can only be written with 0, for flag clearing.

Timer A module standby mode cont

0	Timer A is set to module standby m
1	Timer A module standby mode is c

Timer C module standby mode control

0	Timer C is set to module standby mode
1	Timer C module standby mode is cleared

Timer F module standby mode control

0	Timer F is set to module standby mode
1	Timer F module standby mode is cleared

Timer G interrupt enable

0	Timer G is set to module standby mode
1	Timer G module standby mode is cleared

A/D converter module standby mode control

0	A/D converter is set to module standby mode
1	A/D converter module standby mode is cleared

SCI3-2 module standby mode control

0	SCI3-2 is set to module standby mode
1	SCI3-2 module standby mode is cleared

SCI3-1 module standby mode control

0	SCI3-1 is set to module standby mode
1	SCI3-1 module standby mode is cleared

LCD module standby mode contro

0	LCD is set to module standby m
1	LCD module standby mode is cl

PWM module standby mode control

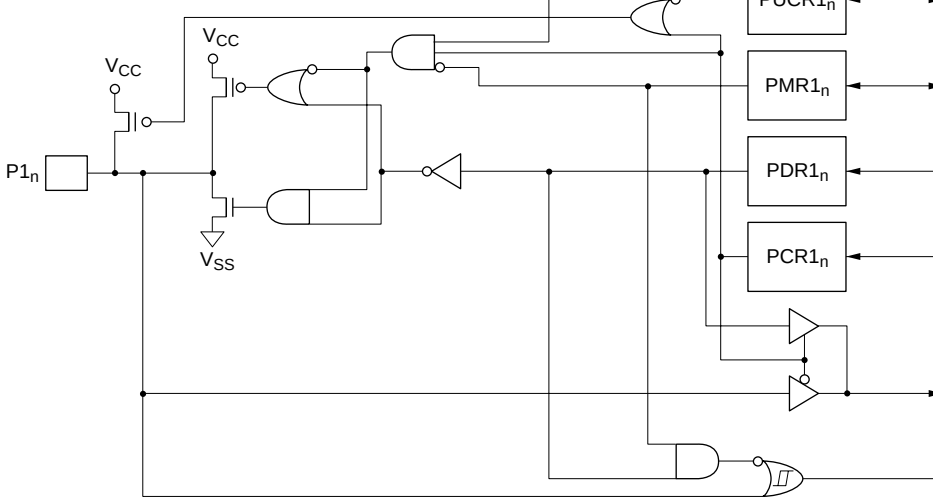
0	PWM is set to module standby mode
1	PWM module standby mode is cleared

WDT module standby mode control

0	WDT is set to module standby mode
1	WDT module standby mode is cleared

Asynchronous event counter module standby mode contro

0	Asynchronous event counter is set to module standby mode
1	Asynchronous event counter module standby mode is cleare



PDR1: Port data register 1
PCR1: Port control register 1
PMR1: Port mode register 1
PUCR1: Port pull-up control register 1

n = 7 to 4

Figure C.1 (a) Port 1 Block Diagram (Pins $P1_7$ to $P1_4$)

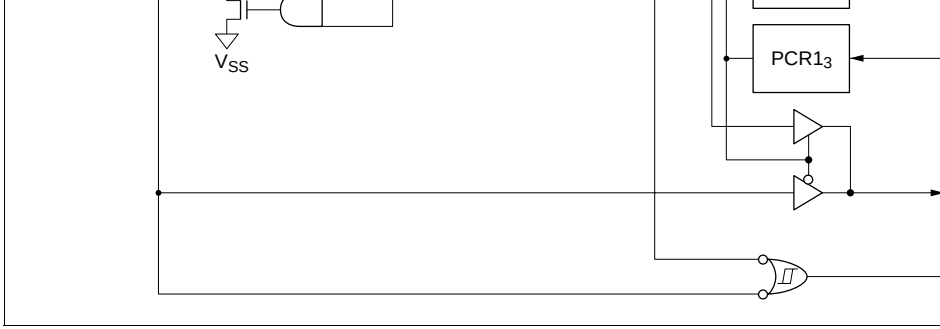


Figure C.1 (b) Port 1 Block Diagram (Pin $P1_3$)

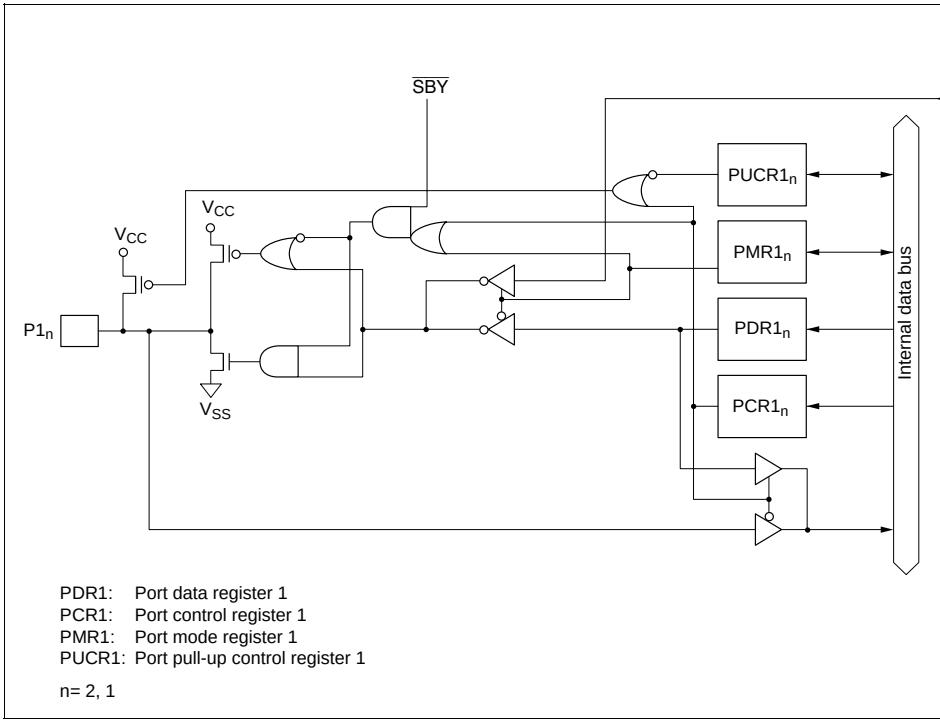
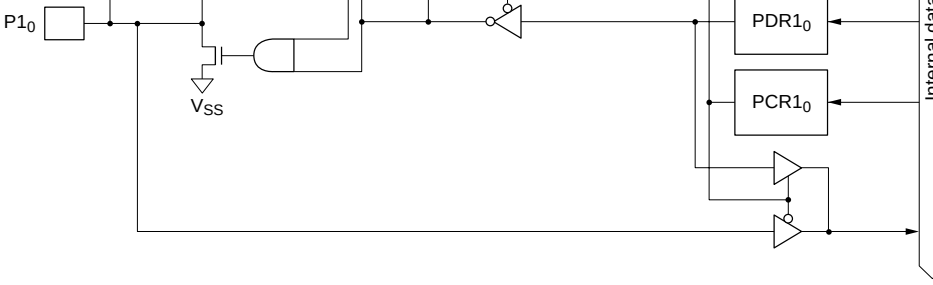
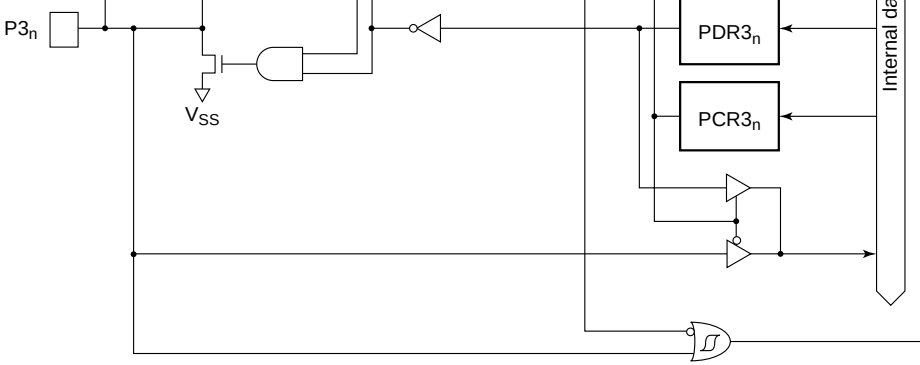


Figure C.1 (c) Port 1 Block Diagram (Pin $P1_2$, $P1_1$)



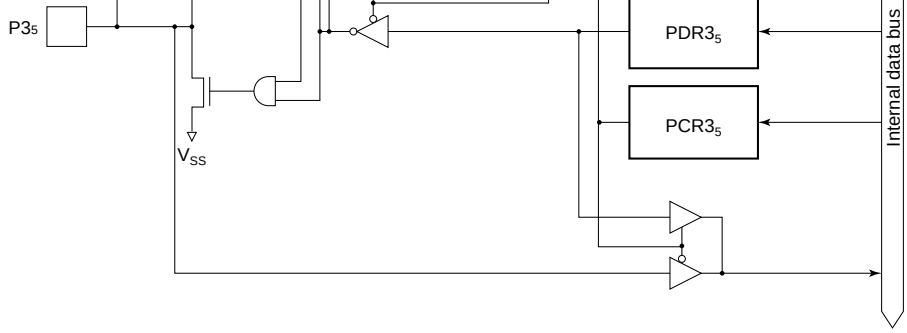
PDR1: Port data register 1
PCR1: Port control register 1
PMR1: Port mode register 1
PUCR1: Port pull-up control register 1

Figure C.1 (d) Port 1 Block Diagram (Pin $P1_0$)



PDR3: Port data register 3
PCR3: Port control register 3
PMR3: Port mode register 3
PUCR3: Port pull-up control register 3
n=7 to 6

Figure C.2 (a) Port 3 Block Diagram (Pin $P3_7$ to $P3_6$)



PDR3: Port data register 3
PCR3: Port control register 3

Figure C.2 (b) Port 3 Block Diagram (Pin $P3_5$)

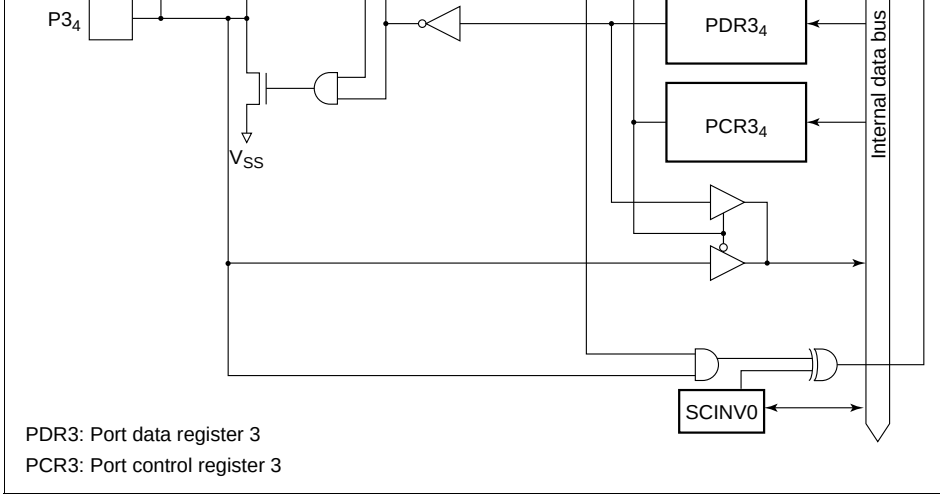
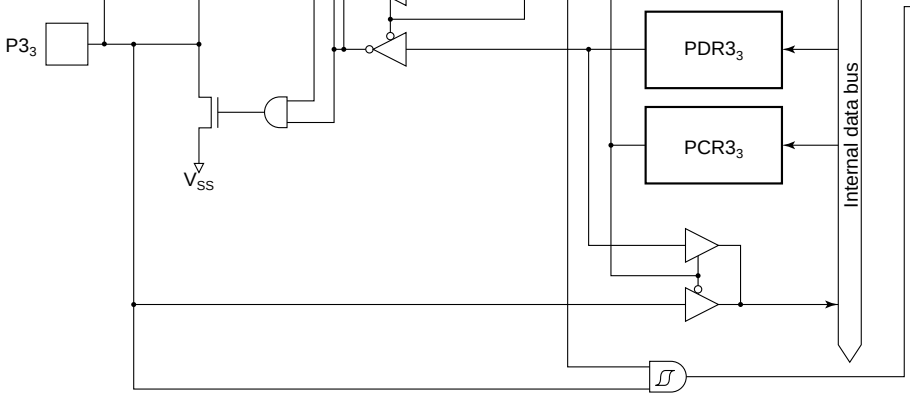


Figure C.2 (c) Port 3 Block Diagram (Pin P3$_4$)

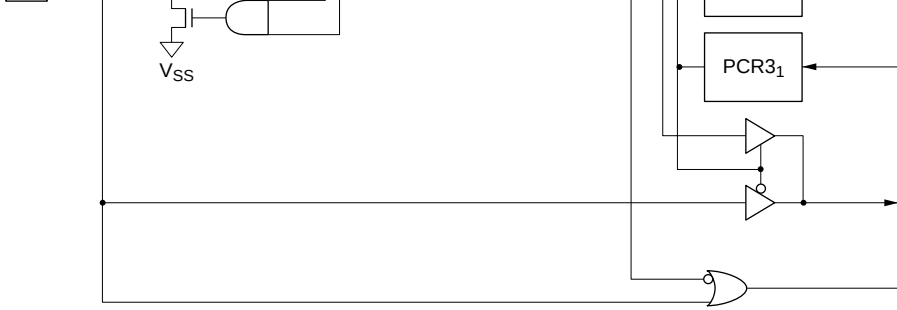


PDR3: Port data register 3
PCR3: Port control register 3

Figure C.2 (d) Port 3 Block Diagram (Pin P3$_3$)

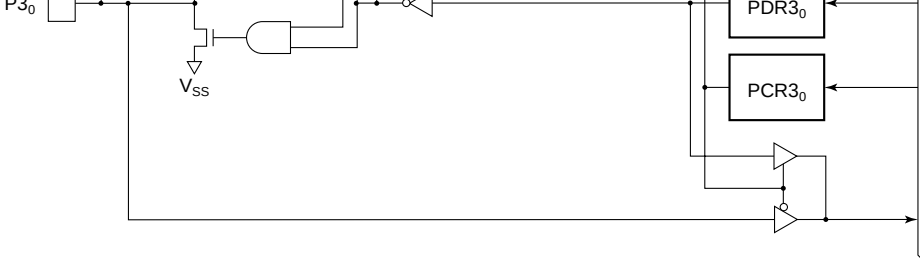
PDR3: Port data register 3
PCR3: Port control register 3
PMR3: Port mode register 3
PUCR3: Port pull-up control register 3

Figure C.2 (e) Port 3 Block Diagram (Pin $P3_2$)



PDR3: Port data register 3
PCR3: Port control register 3
PMR3: Port mode register 3
PUCR3: Port pull-up control register 3

Figure C.2 (f) Port 3 Block Diagram (Pin $P3_1$)



PDR3: Port data register 3
PCR3: Port control register 3
PMR3: Port mode register 3
PUCR3: Port pull-up control register 3

Figure C.2 (g) Port 3 Block Diagram (Pin P3$_0$)

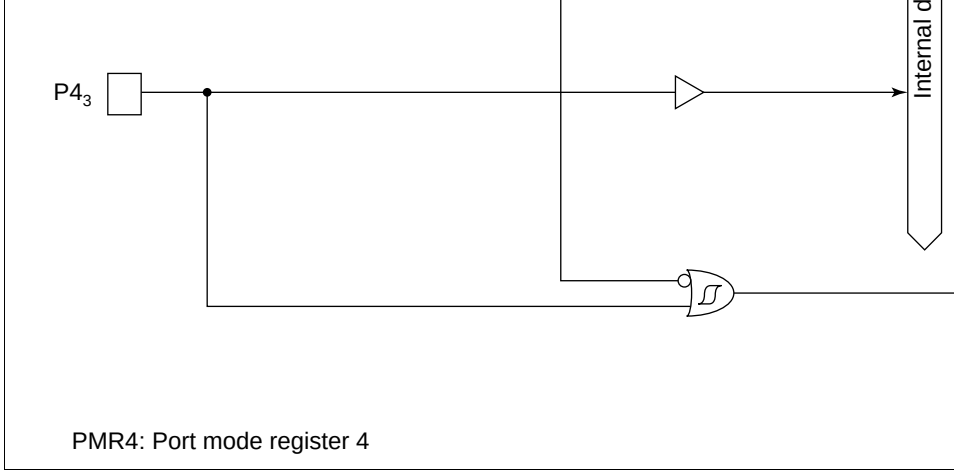
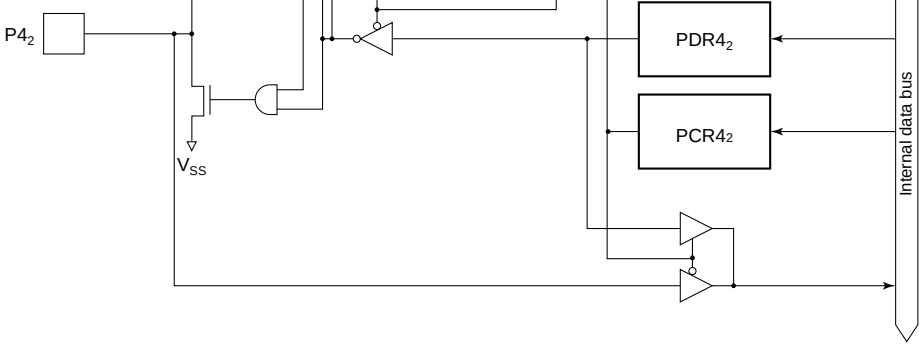
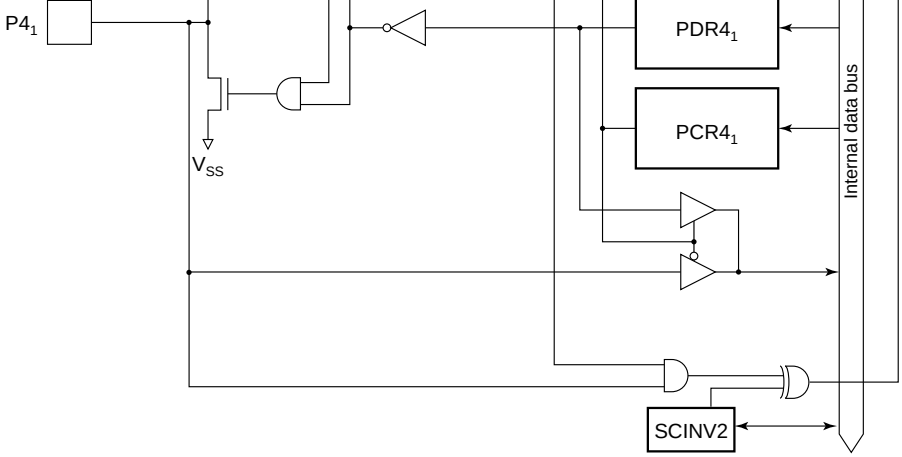


Figure C.3 (a) Port 4 Block Diagram (Pin P4$_3$)



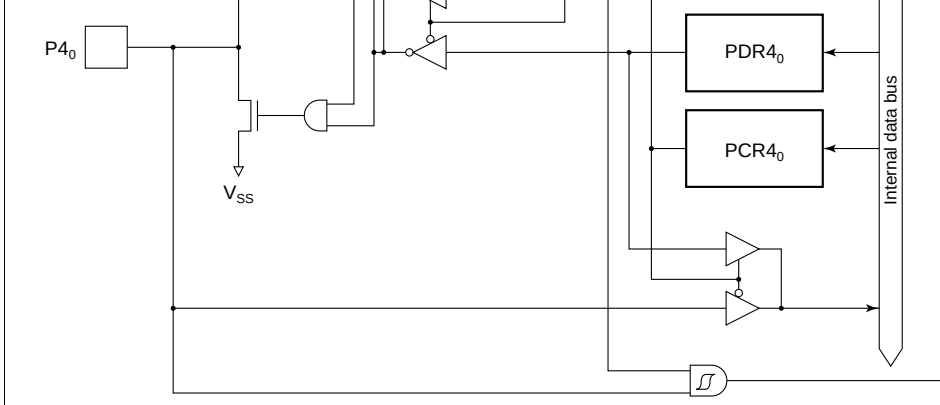
PDR4: Port data register 4
PCR4: Port control register 4

Figure C.3 (b) Port 4 Block Diagram (Pin P4$_2$)



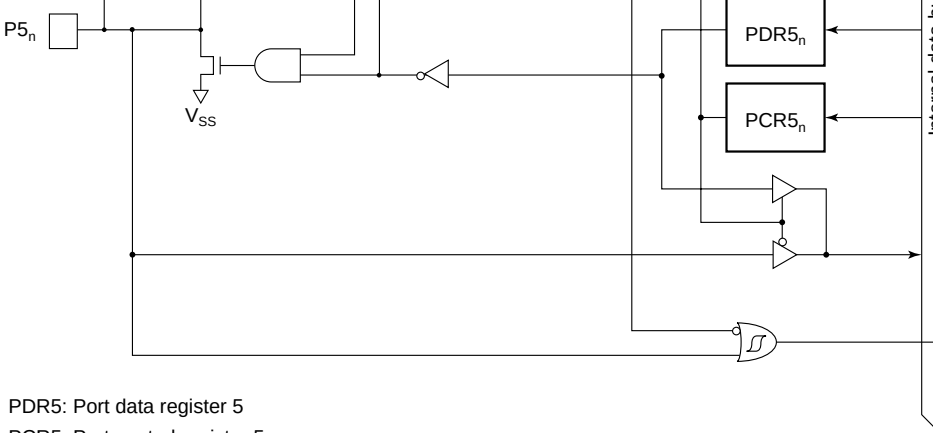
PDR4: Port data register 4
PCR4: Port control register 4

Figure C.3 (c) Port 4 Block Diagram (Pin P4$_1$)



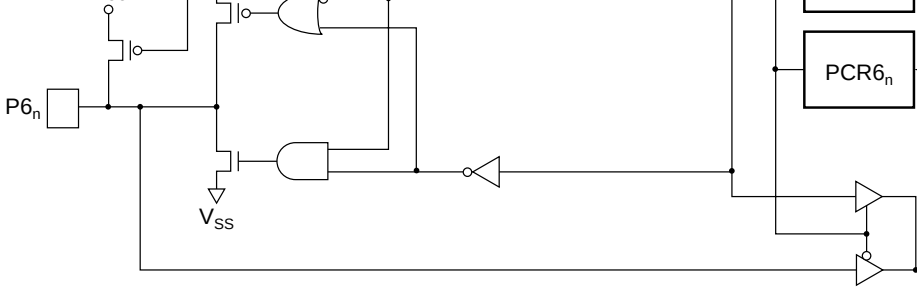
PDR4: Port data register 4
PCR4: Port control register 4

Figure C.3 (d) Port 4 Block Diagram (Pin P4$_0$)



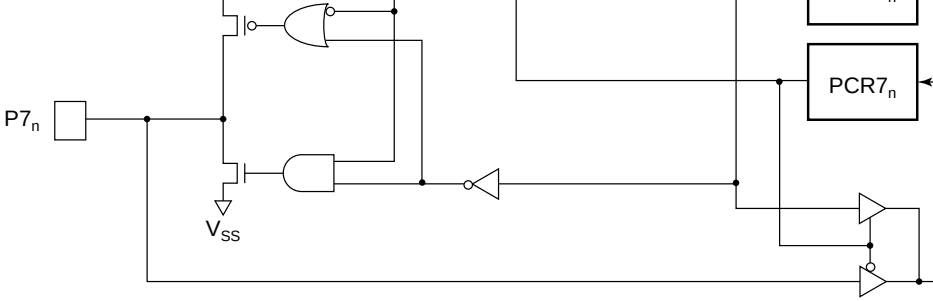
PDR5: Port data register 5
PCR5: Port control register 5
PMR5: Port mode register 5
PUCR5: Port pull-up control register 5
n = 7 to 0

Figure C.4 Port 5 Block Diagram



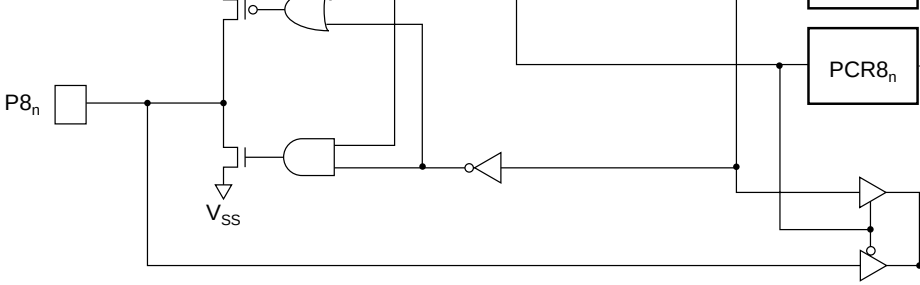
PDR6: Port data register 6
PCR6: Port control register 6
PUCR6: Port pull-up control register 6
n = 7 to 0

Figure C.5 Port 6 Block Diagram



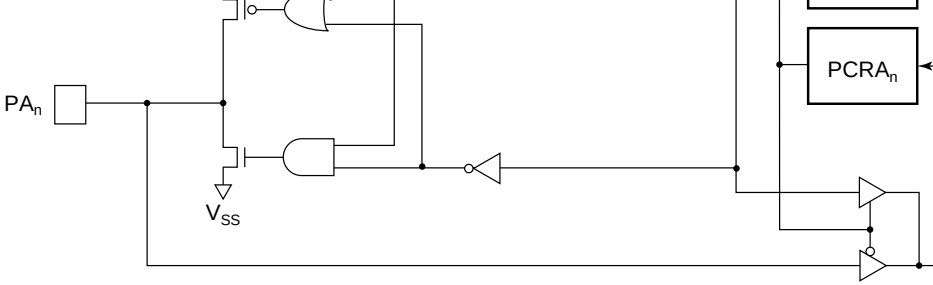
PDR7: Port data register 7
PCR7: Port control register 7
n = 7 to 0

Figure C.6 Port 7 Block Diagram



PDR8: Port data register 8
PCR8: Port control register 8
n= 7 to 0

Figure C.7 Port 8 Block Diagram



PDRA: Port data register A
PCRA: Port control register A
n = 3 to 0

Figure C.8 Port A Block Diagram

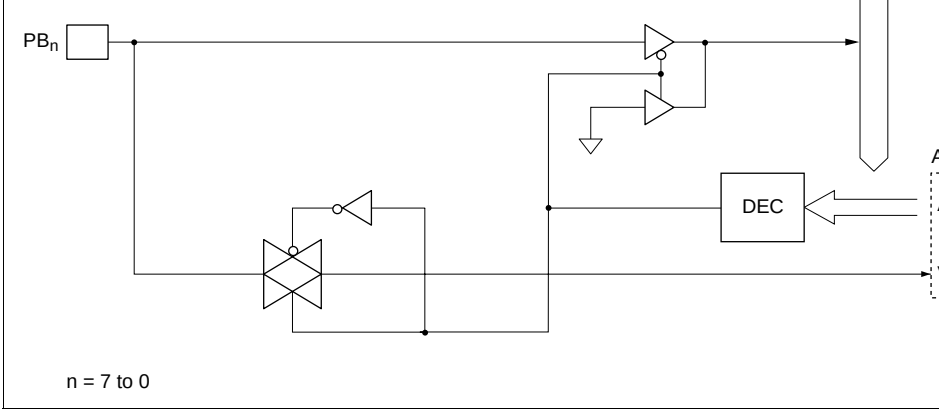


Figure C.9 Port B Block Diagram

$P3_0$	impedance			impedance			
$P4_3$ to $P4_0$	High impedance	Retained	Retained	High impedance	Retained	Functions	I
$P5_7$ to $P5_0$	High impedance	Retained	Retained	High impedance[*1]	Retained	Functions	I
$P6_7$ to $P6_0$	High impedance	Retained	Retained	High impedance	Retained	Functions	I
$P7_7$ to $P7_0$	High impedance	Retained	Retained	High impedance	Retained	Functions	I
$P8_7$ to $P8_0$	High impedance	Retained	Retained	High impedance	Retained	Functions	I
PA_3 to PA_0	High impedance	Retained	Retained	High impedance	Retained	Functions	I
PB_7 to PB_0	High impedance	High impedance	High impedance	High impedance	High impedance	High impedance	i

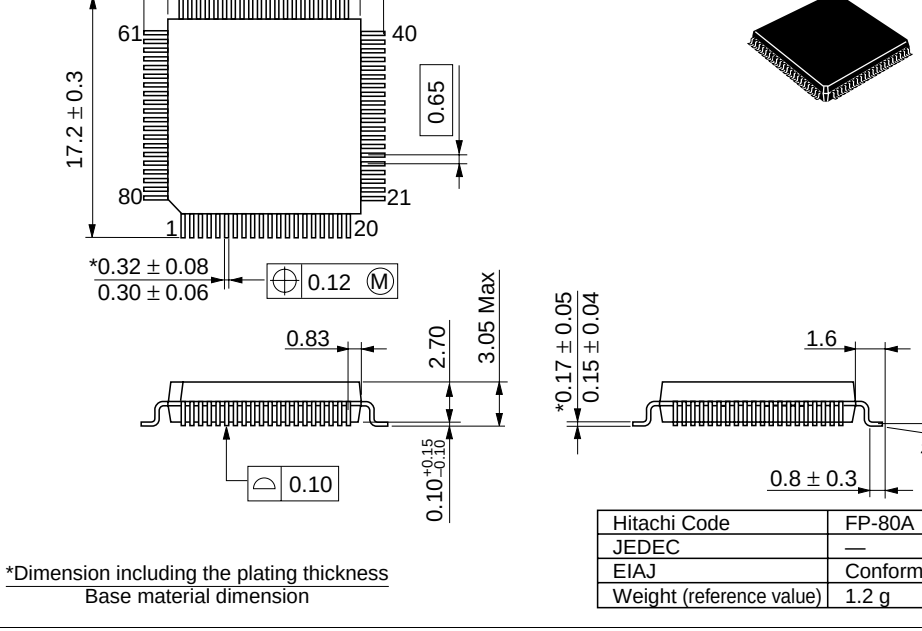
Notes: 1. High level output when MOS pull-up is in on state.
2. $P3_2$ is Functions

		HD6433862W	HD6433862(***)W	80-pin TQFP
H8/3863	Mask ROM versions	HD6433863H	HD6433863(***)H	80-pin QFP
		HD6433863F	HD6433863(***)F	80-pin QFP
		HD6433863W	HD6433863(***)W	80-pin TQFP
H8/3864	Mask ROM versions	HD6433864H	HD6433864(***)H	80-pin QFP
		HD6433864F	HD6433864(***)F	80-pin QFP
		HD6433864W	HD6433864(***)W	80-pin TQFP
H8/3865	Mask ROM versions	HD6433865H	HD6433865(***)H	80-pin QFP
		HD6433865F	HD6433865(***)F	80-pin QFP
		HD6433865W	HD6433865(***)W	80-pin TQFP
H8/3866	Mask ROM versions	HD6433866H	HD6433866(***)H	80-pin QFP
		HD6433866F	HD6433866(***)F	80-pin QFP
		HD6433866W	HD6433866(***)W	80-pin TQFP
H8/3867	Mask ROM versions	HD6433867H	HD6433867(***)H	80-pin QFP
		HD6433867F	HD6433867(***)F	80-pin QFP
		HD6433867W	HD6433867(***)W	80-pin TQFP
	ZTAT versions	HD6473867H	HD6473867H	80-pin QFP
		HD6473867F	HD6473867F	80-pin QFP
		HD6473867W	HD6473867W	80-pin TQFP

Note: For mask ROM versions, (***) is the ROM code.

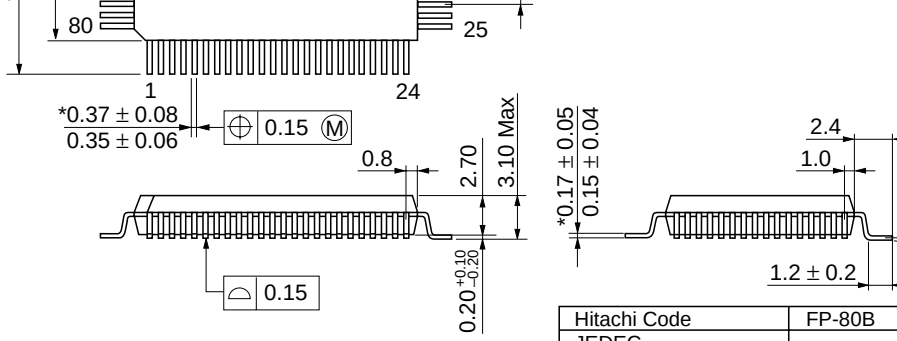
	versions	HD6433823F	HD6433823(***)F	80-pin QFP (F
		HD6433823W	HD6433823(***)W	80-pin TQFP
H8/3824	Mask ROM	HD6433824H	HD6433824(***)H	80-pin QFP (F
	versions	HD6433824F	HD6433824(***)F	80-pin QFP (F
		HD6433824W	HD6433824(***)W	80-pin TQFP
H8/3825	Mask ROM	HD6433825H	HD6433825(***)H	80-pin QFP (F
	versions	HD6433825F	HD6433825(***)F	80-pin QFP (F
		HD6433825W	HD6433825(***)W	80-pin TQFP
H8/3826	Mask ROM	HD6433826H	HD6433826(***)H	80-pin QFP (F
	versions	HD6433826F	HD6433826(***)F	80-pin QFP (F
		HD6433826W	HD6433826(***)W	80-pin TQFP
H8/3827	Mask ROM	HD6433827H	HD6433827(***)H	80-pin QFP (F
	versions	HD6433827F	HD6433827(***)F	80-pin QFP (F
		HD6433827W	HD6433827(***)W	80-pin TQFP
	ZTAT	HD6473827H	HD6473827H	80-pin QFP (F
	versions	HD6473827F	HD6473827F	80-pin QFP (F
		HD6473827W	HD6473827W	80-pin TQFP

Note: For mask ROM versions, (***) is the ROM code.



Hitachi Code	FP-80A
JEDEC	—
EIAJ	Conform
Weight (reference value)	1.2 g

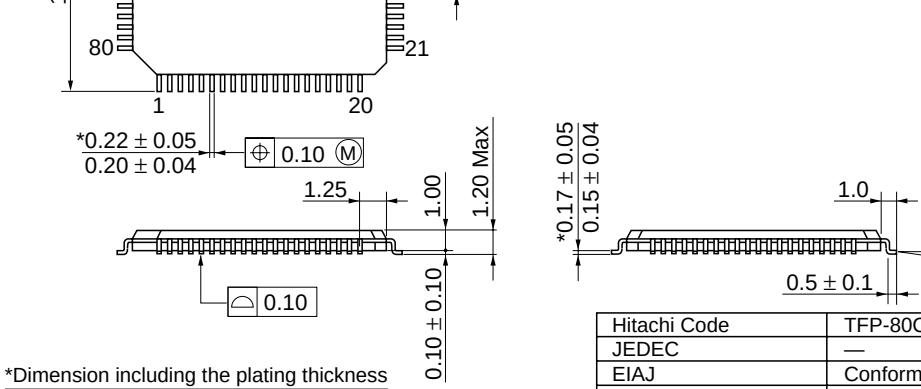
Figure F.1 FP-80A Package Dimensions



*Dimension including the plating thickness
Base material dimension

Hitachi Code	FP-80B
JEDEC	—
EIAJ	—
Weight (reference value)	1.7 g

Figure F.2 FP-80B Package Dimensions



Hitachi Code	TFP-80C
JEDEC	—
EIAJ	Conform
Weight (reference value)	0.4 g

Figure F.3 TFP-80C Package Dimensions

H8/3867 Series, H8/3827 Series Hardware Manual

Publication Date: 1st Edition, November 1997
3rd Edition, February 1999
Published by: Electronic Devices Sales & Marketing Group
Semiconductor & Integrated Circuits
Hitachi, Ltd.
Edited by: Technical Documentation Group
UL Media Co., Ltd.

X-ON Electronics

Largest Supplier of Electrical and Electronic Components

Click to view similar products for 16-bit Microcontrollers - MCU ***category:***

Click to view products by Renesas ***manufacturer:***

Other Similar products are found below :

M30302FCPFP#U3 MB90F036APMC-GSE1 MB90F428GCPFR-GSE1 MB96F683RBPMC-GSAE1 R5F10MMGDFB#30 R5F111PGGFB#30 R5F117BCGNA#20 DF3026XBL25V DF36014GFTV DF36014GFXV DF36034GFPV R5F11B7EANA#U0 R5F21172DSP#U0 MB90092PF-G-BNDE1 MB90F335APMC1-G-SPE1 MB90F345CAPFR-GSE1 MB90F568PMCR-GE1 MB96F395RSAPMC-GSE2 DF36024GFXV UPD78F1018F1-BA4-A MB96F018RBPMC-GSE1 MB90F867ASPFR-GE1 M30290FCHP#U3A DF2239FA20IV R5F117BCGFP#30 LC88F58B0AU-SQFPH MB90F548GPF-GE1 MB90214PF-GT-310-BND-AE1 MB90F342CESPQC-GSE2 MB90F428GAPF-GSE1 ML620Q504H-NNNTBWBX S912ZVH128F2VLL UPD78F1500AGK-GAK-AX HD64F3337SF16V MB90F428GCPF-GSE1 MB90F342ESPMC-G-JNE1 MB90022PF-GS-358E1 MB96F395RWAPMC-GSE2 MB96395RSAPMC-GS-110E2 MB90F883CSPMC-GE1 S912ZVHY64F1CLL S912ZVHY64F1VLQ ST10F280 MB96F338RSAPMCR-GK5E2 CY90096PF-G-002-BND-ERE1 ML62Q1569-NNNGAZ0AX ML62Q1739-NNNGAZ0AX ML62Q1749-NNNGAZ0AX ML62Q1579-NNNGAZ0AX ML62Q1559-NNNGAZ0AX