To our customers,

Old Company Name in Catalogs and Other Documents

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

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

April 1st, 2010
Renesas Electronics Corporation

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

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

Notice

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

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

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

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

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

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

User's Manual

16

H8S/2556 Group, H8S/2552 Group, H8S/2506 Group

Hardware Manual

Renesas 16-Bit Single-Chip Microcomputer H8S Family/H8S/2500 Series

H8S/2556	HD64F2556
H8S/2552	HD64F2552
H8S/2551	HD64F2551
H8S/2506	HD64F2506
H8S/2505	HD64F2505

Renesas Electronics
www.renesas.com

Rev.6.00 2009.09

document, please confirm the latest product information with a Renesas sales office. Also, please pay and careful attention to additional and different information to be disclosed by Renesas such as that di through our website. (http://www.renesas.com)

5. Renesas has used reasonable care in compiling the information included in this document, but Renesa assumes no liability whatsoever for any damages incurred as a result of errors or omissions in the info included in this document.
6. When using or otherwise relying on the information in this document, you should evaluate the informat light of the total system before deciding about the applicability of such information to the intended appl Renesas makes no representations, warranties or guaranties regarding the suitability of its products fo particular application and specifically disclaims any liability arising out of the application and use of the information in this document or Renesas products.
7. With the exception of products specified by Renesas as suitable for automobile applications, Renesas products are not designed, manufactured or tested for applications or otherwise in systems the failure malfunction of which may cause a direct threat to human life or create a risk of human injury or which r especially high quality and reliability such as safety systems, or equipment or systems for transportatio traffic, healthcare, combustion control, aerospace and aeronautics, nuclear power, or undersea commu transmission. If you are considering the use of our products for such purposes, please contact a Renes sales office beforehand. Renesas shall have no liability for damages arising out of the uses set forth ab
8. Notwithstanding the preceding paragraph, you should not use Renesas products for the purposes liste
   (1) artificial life support devices or systems
   (2) surgical implantations
   (3) healthcare intervention (e.g., excision, administration of medication, etc.)
   (4) any other purposes that pose a direct threat to human life

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

induced in the vicinity of LSI, an associated shoot-through current flows internal
malfunctions may occur due to the false recognition of the pin state as an input
Unused pins should be handled as described under Handling of Unused Pins in
manual.

2. Processing at Power-on

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

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

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

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

3. Prohibition of Access to Reserved Addresses

   Access to reserved addresses is prohibited.

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

4. Clock Signals

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

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

5. Differences between Products

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

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

- CPU and System-Control Modules
- On-Chip Peripheral Modules

The configuration of the functional description of each module differs according to th module. However, the generic style includes the following items:

i) Feature
ii) Input/Output Pin
iii) Register Description
iv) Operation
v) Usage Note

When designing an application system that includes this LSI, take notes into account. Eac includes notes in relation to the descriptions given, and usage notes are given, as required final part of each section.

7. List of Registers
8. Electrical Characteristics
9. Appendix
10. Main Revisions for This Edition (only for revised versions)

The list of revisions is a summary of points that have been revised or added to earlier vers This does not include all of the revised contents. For details, see the actual locations in th manual.

11. Index

List of on-chip peripheral functions:

Group Name	H8S/2556 Group	H8S/2552 Group	H8S/250
Product Name	H8S/2556	H8S/2552 H8S/2551	H8S/250 H8S/250
Bus controller	O (16 bits)	O (16 bits)	O (16 bit
Data transfer controller (DTC)	O	O	O
PC break controller (PBC)	O	O	O
16-bit timer pulse unit (TPU)	× 6	× 6	× 6
8-bit timer (TMR)	× 4	× 4	× 4
Watch dog timer (WDT)	× 2	× 2	× 2
Serial communication interface (SCI)	× 5	× 5	× 5
I^2C bus interface 2 (IIC2)	× 2	× 2	× 2
IEBus™*2 controller (IEB)	—	× 1	—
Controller area network (HCAN)	× 1	—	—
D/A converter	× 2	× 2	× 2
A/D converter	× 16	× 16	× 16

Notes: 1. F-ZTAT is a trademark of Renesas Technology Corp.
2. IEBus is a trademark of NEC Electronics Corporation.

- In order to understand the overall functions of the chip

  Read the manual according to the contents. This manual can be roughly categorized in on the CPU, system control functions, peripheral functions and electrical characteristi
- In order to understand the details of the CPU's functions

  Read the H8S/2600 Series, H8S/2000 Series Software Manual.
- In order to understand the details of a register when its name is known

  Read the index that is the final part of the manual to find the page number of the entry register. The addresses, bits, and initial values of the registers are summarized in secti List of Registers.

  Examples:
  - Register name: The following notation is used for cases when the sa similar function, e.g. 16-bit timer pulse unit or serial communication, is implemented on more than one c XXX_N (XXX is the register name and N is the cha number)
  - Bit order: The MSB is on the left and the LSB is on the right.
  - Number notation: Binary is B'xxxx, hexadecimal is H'xxxx, decimal
  - Signal notation: An overbar is added to a low-active signal: $\overline{\text{xxxx}}$

Related Manuals: The latest versions of all related manuals are available from our web sit ensure you have the latest versions of all documents you require.
http://www.renesas.com/

Compiler Package Ver.6.01 User's Manual	
H8S, H8/300 Series Simulator/Debugger User's Manual	REJ10B02
H8S, H8/300 Series High-performance Embedded Workshop User's Manual	REJ10J20

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

Section 9 I/O Ports

Section 10 16-Bit Timer Pulse Unit (TPU)

Section 15 A/D Converter

Section 16 D/A Converter

Section 18 Controller Area Network (HCAN) [H8S/2556 Group]

Section 20 Flash Memory

Section 21 Clock Pulse Generator

Section 22 Power-Down Modes

Section 24 Electrical Characteristics

Section 20 Flash Memory

— PC break controller
— Data transfer controller (DTC)
— 16-bit timer-pulse unit (TPU)
— 8-bit timer (TMR)
— Watchdog timer (WDT)
— Serial communication interface (SCI)
— I^2C bus interface 2 (IIC2)
— 10-bit A/D converter
— 8-bit D/A converter
— IEBus™ controller (IEB) (H8S/2552, H8S/2551)
— Controller area network (HCAN) (H8S/2556)

- On-chip memory

ROM	Part No.	ROM	RAM	Remarks
Flash memory version	HD64F2556	512 kbytes	32 kbytes	
	HD64F2552	512 kbytes	32 kbytes	
	HD64F2551	384 kbytes	24 kbytes	
	HD64F2506	512 kbytes	32 kbytes	
	HD64F2505	384 kbytes	32 kbytes	

- Supports various power-down modes
- Compact package

Package	Code*2	Body Size	Pin Pitch
QFP-144	FP-144J/FP-144JV	20.0 × 20.0 mm	0.5 mm
LFBGA-176*1	BP-176V	13.0 × 13.0 mm	0.8 mm

Notes: 1. Available only in the H8S/2552 Group and H8S/2506 Group.
2. Package code ending in the letter V designate Pb-free Product.

1.2 Internal Block Diagram

Figures 1.1 to 1.3 show the internal block diagrams.

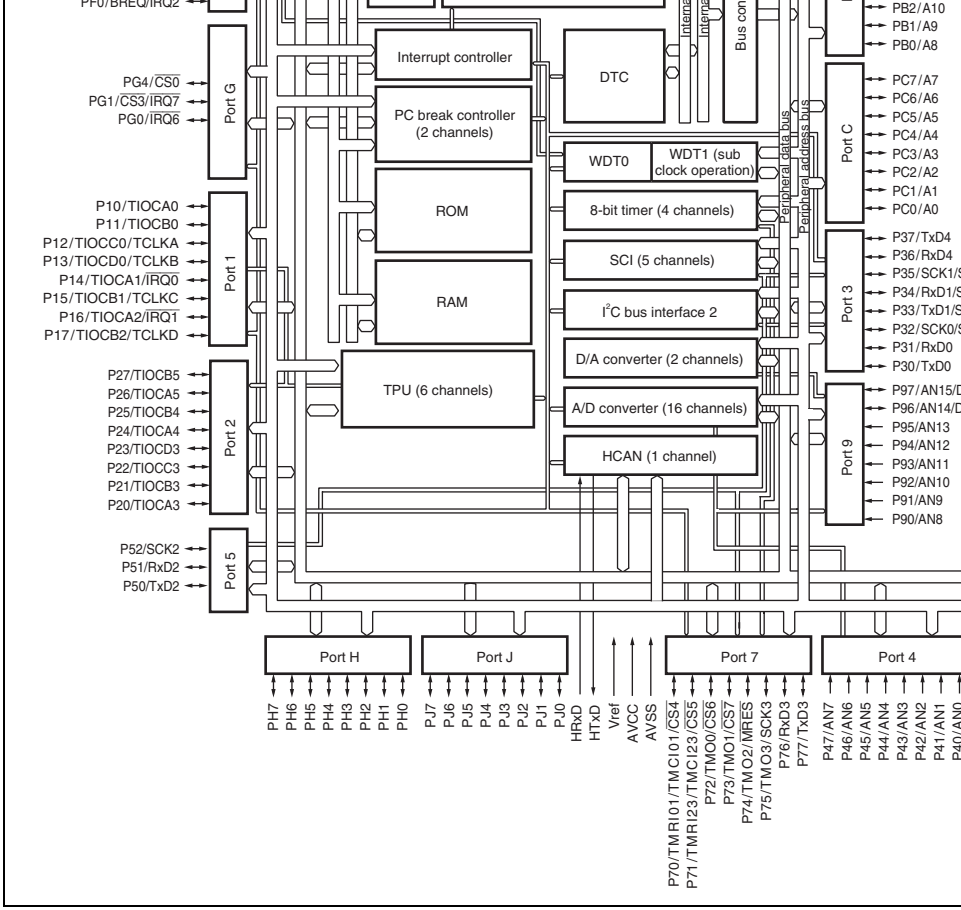


Figure 1.1 Internal Block Diagram of H8S/2556 Group

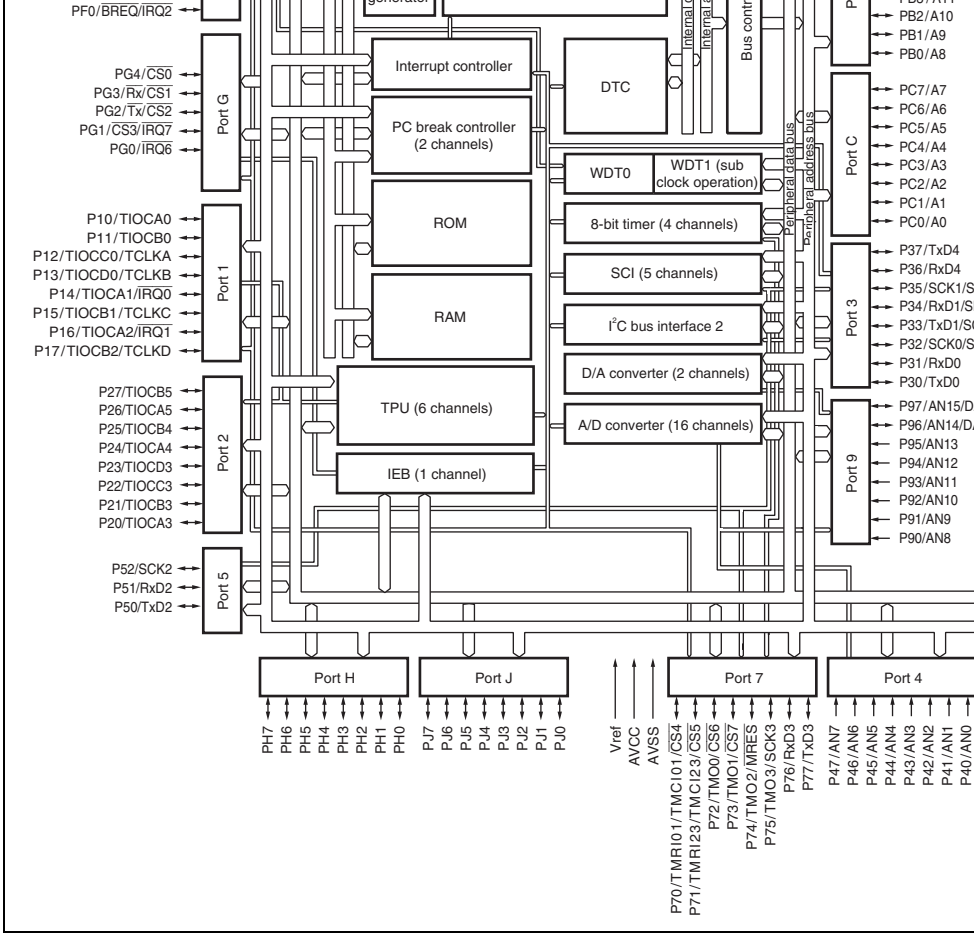


Figure 1.2 Internal Block Diagram of H8S/2552 Group

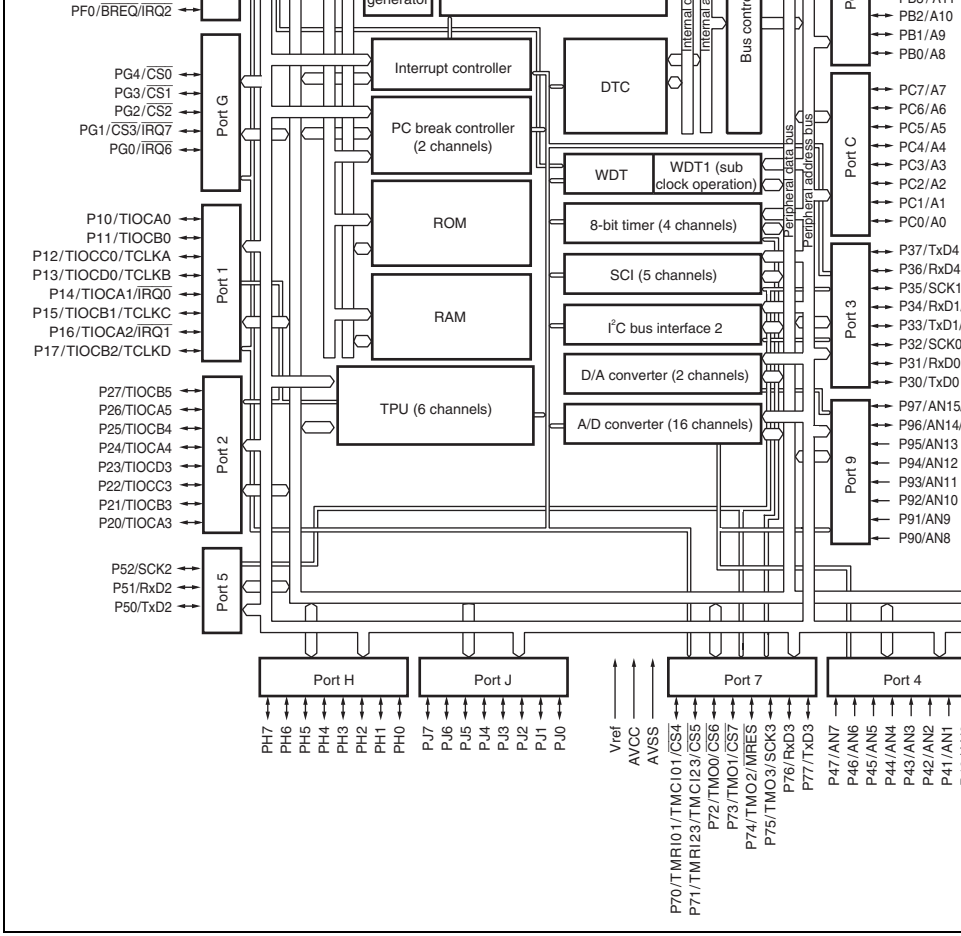


Figure 1.3 Internal Block Diagram of H8S/2506 Group

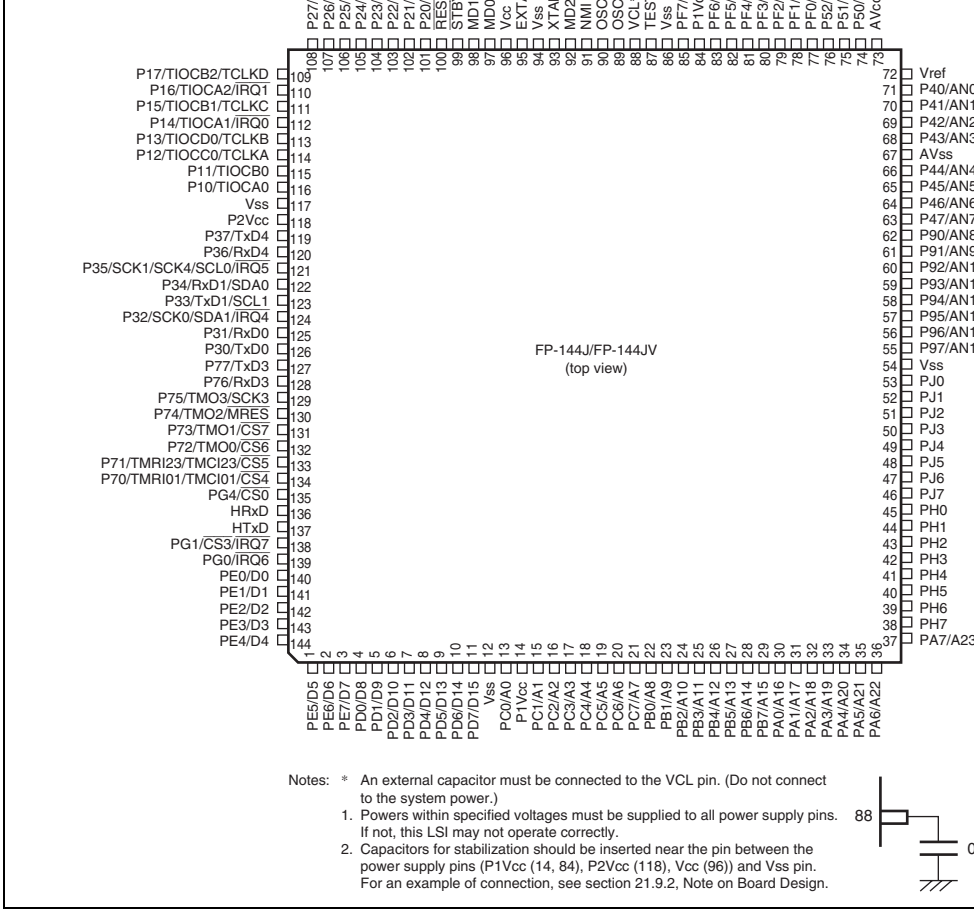
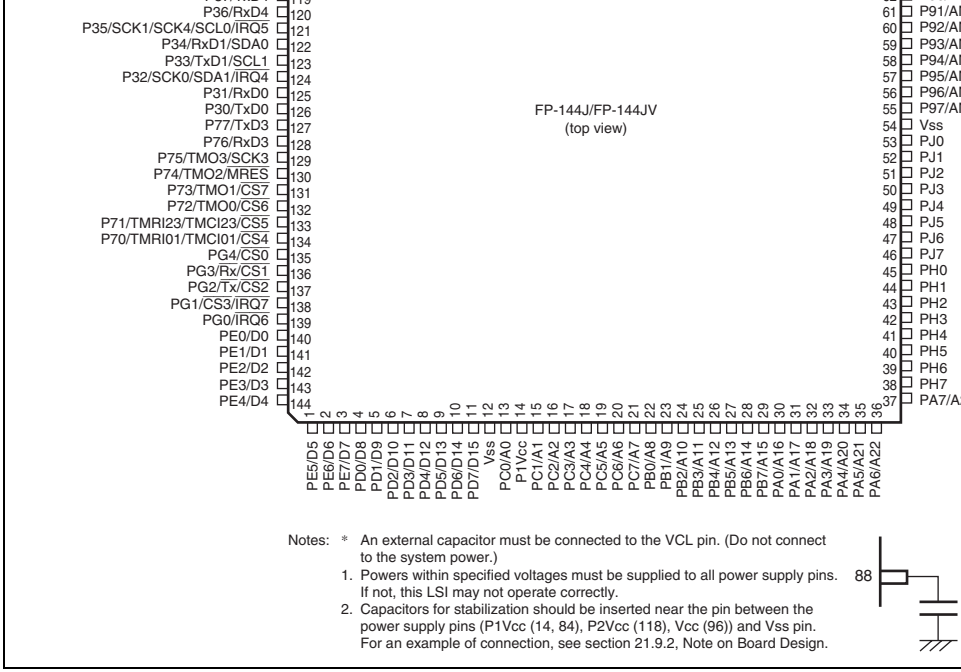


Figure 1.4 Pin Arrangement of H8S/2556 Group (FP-144J and FP-144JV)



Notes: * An external capacitor must be connected to the VCL pin. (Do not connect to the system power.)
1. Powers within specified voltages must be supplied to all power supply pins. If not, this LSI may not operate correctly.
2. Capacitors for stabilization should be inserted near the pin between the power supply pins (P1Vcc (14, 84), P2Vcc (118), Vcc (96)) and Vss pin. For an example of connection, see section 21.9.2, Note on Board Design.

Figure 1.5 Pin Arrangement of H8S/2552 Group (FP-144J and FP-144JV

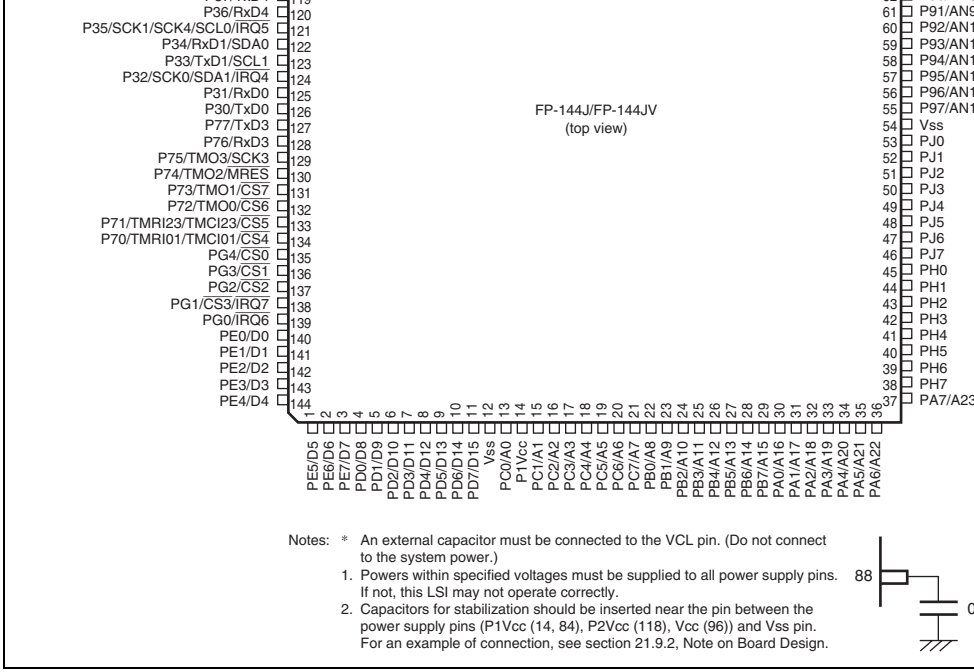
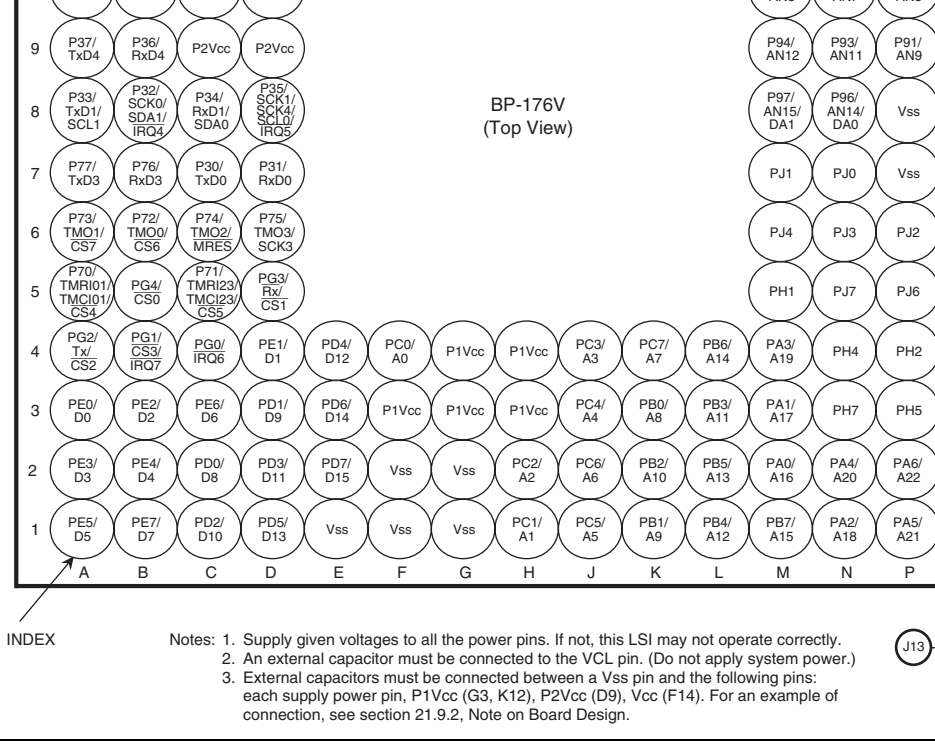


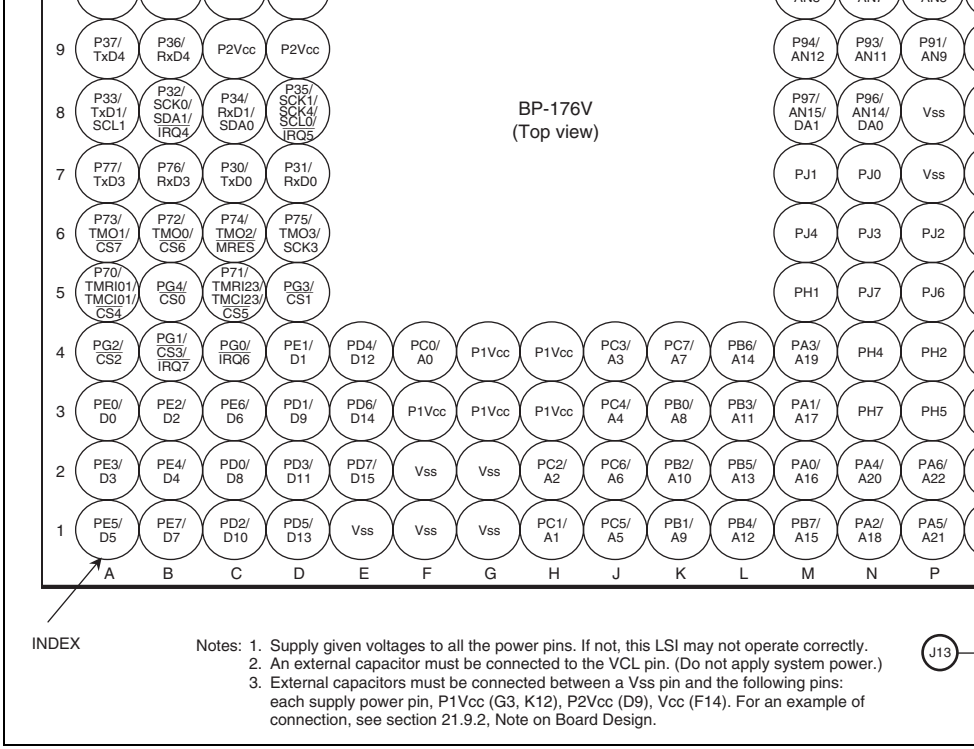
Figure 1.6 Pin Arrangement of H8S/2506 Group (FP-144J and FP-144JV)



Notes: 1. Supply given voltages to all the power pins. If not, this LSI may not operate correctly.
2. An external capacitor must be connected to the VCL pin. (Do not apply system power.)
3. External capacitors must be connected between a Vss pin and the following pins: each supply power pin, P1Vcc (G3, K12), P2Vcc (D9), Vcc (F14). For an example of connection, see section 21.9.2, Note on Board Design.

J13

Figure 1.7 Pin Arrangement of H8S/2552 Group (BP-176V)



Notes: 1. Supply given voltages to all the power pins. If not, this LSI may not operate correctly.
2. An external capacitor must be connected to the VCL pin. (Do not apply system power.)
3. External capacitors must be connected between a Vss pin and the following pins: each supply power pin, P1Vcc (G3, K12), P2Vcc (D9), Vcc (F14). For an example of connection, see section 21.9.2, Note on Board Design.

J13

Figure 1.8 Pin Arrangement of H8S/2506 Group (BP-176V)

3	B1	PE7/D7	PE7	$\overline{CE}$
4	C2	D8	PD0	D0
5	D3	D9	PD1	D1
6	C1	D10	PD2	D2
7	D2	D11	PD3	D3
8	E4	D12	PD4	D4
9	D1	D13	PD5	D5
10	E3	D14	PD6	D6
11	E2	D15	PD7	D7
12	G2, G1, F2, F1, E1	Vss	Vss	Vss
13	F4	PC0/A0	PC0	A0
14	H4, H3, G4, G3, F3	P1Vcc	P1Vcc	Vcc
15	H1	PC1/A1	PC1	A1
16	H2	PC2/A2	PC2	A2
17	J4	PC3/A3	PC3	A3
18	J3	PC4/A4	PC4	A4
19	J1	PC5/A5	PC5	A5

26	L1	PB4/A12	PB4	A12
27	L2	PB5/A13	PB5	A13
28	L4	PB6/A14	PB6	A14
29	M1	PB7/A15	PB7	A15
30	M2	PA0/A16	PA0	A16
31	M3	PA1/A17	PA1	A17
32	N1	PA2/A18	PA2	A18
33	M4	PA3/A19	PA3	NC
34	N2	PA4/A20	PA4	NC
35	P1	PA5/A21	PA5	NC
36	P2	PA6/A22	PA6	NC
37	R1	PA7/A23	PA7	NC
38	N3	PH7	PH7	NC
39	R2	PH6	PH6	NC
40	P3	PH5	PH5	NC
41	N4	PH4	PH4	NC
42	R3	PH3	PH3	NC
43	P4	PH2	PH2	NC
44	M5	PH1	PH1	NC
45	R4	PH0	PH0	NC
46	N5	PJ7	PJ7	NC
47	P5	PJ6	PJ6	NC

54	R7, R6, P8, P7	Vss	Vss	Vss
55	M8	P97/AN15/DA1	P97/AN15/DA1	NC
56	N8	P96/AN14/DA0	P96/AN14/DA0	NC
57	R8	P95/AN13	P95/AN13	NC
58	M9	P94/AN12	P94/AN12	NC
59	N9	P93/AN11	P93/AN11	NC
60	R9	P92/AN10	P92/AN10	NC
61	P9	P91/AN9	P91/AN9	NC
62	M10	P90/AN8	P90/AN8	NC
63	N10	P47/AN7	P47/AN7	NC
64	R10	P46/AN6	P46/AN6	NC
65	P10	P45/AN5	P45/AN5	NC
66	N11	P44/AN4	P44/AN4	NC
67	R12, R11	AVss	AVss	Vss
68	P11	P43/AN3	P43/AN3	NC
69	M11	P42/AN2	P42/AN2	NC
70	P12	P41/AN1	P41/AN1	NC
71	N12	P40/AN0	P40/AN0	NC
72	R15, R14, R13	Vref	Vref	Vcc
73	P14, P13, N13, M12	AVcc	AVcc	Vcc
74	P15	P50/TxD2	P50/TxD2	NC

81	L13	$\overline{\text{HWR}}$	PF4	NC
82	L14	$\overline{\text{RD}}$	PF5	NC
83	L15	$\overline{\text{AS}}$	PF6	NC
84	K12	P1Vcc	P1Vcc	Vcc
85	K13	PF7/ϕ	PF7/ϕ	NC
86	K15, K14	Vss	Vss	Vss
87	J12	TEST	TEST	Vss
88	J13	VCL	VCL	VCL
89	J15	OSC2	OSC2	NC
90	J14	OSC1	OSC1	Vss
91	H12	NMI	NMI	Vcc
92	H13	MD2	MD2	Vss
93	G15	XTAL	XTAL	XTAL
94	H15, H14, G14, G13, G12, F13, F12	Vss	Vss	Vss
95	F15	EXTAL	EXTAL	EXTAL
96	F14	Vcc	Vcc	Vcc
97	E12	MD0	MD0	Vss
98	E15	MD1	MD1	Vss
99	E14	$\overline{\text{STBY}}$	$\overline{\text{STBY}}$	Vcc
100	E13	$\overline{\text{RES}}$	$\overline{\text{RES}}$	$\overline{\text{RES}}$

107	B13	P26/TIOCA5	P26/TIOCA5	NC
108	B14	P27/TIOCB5	P27/TIOCB5	NC
109	A15	P17/TIOCB2/ TCLKD	P17/TIOCB2/ TCLKD	NC
110	C13	P16/TIOCA2/$\overline{\text{IRQ1}}$	P16/TIOCA2/$\overline{\text{IRQ1}}$	Vss
111	B13	P15/TIOCB1/ TCLKC	P15/TIOCB1/ TCLKC	NC
112	C12	P14/TIOCA1/$\overline{\text{IRQ0}}$	P14/TIOCA1/$\overline{\text{IRQ0}}$	Vss
113	B12	P13/TIOCD0/ TCLKB	P13/TIOCD0/ TCLKB	NC
114	D11	P12/TIOCC0/ TCLKA	P12/TIOCC0/ TCLKA	NC
115	C11	P11/TIOCB0	P11/TIOCB0	NC
116	B11	P10/TIOCA0	P10/TIOCA0	NC
117	B10, A14, A13, A12, A11, A10	Vss	Vss	Vss
118	D10, D9, C10, C9	P2Vcc	P2Vcc	Vcc
119	A9	P37/TxD4	P37/TxD4	NC
120	B9	P36/RxD4	P36/RxD4	NC
121	D8	P35/SCK1/SCK4/ SCL0/$\overline{\text{IRQ5}}$	P35/SCK1/SCK4/ SCL0/$\overline{\text{IRQ5}}$	NC
122	C8	P34/RxD1/SDA0	P34/RxD1/SDA0	NC
123	A8	P33/TxD1/SCL1	P33/TxD1/SCL1	NC
124	B8	P32/SCK0/SDA1/ $\overline{\text{IRQ4}}$	P32/SCK0/SDA1/ $\overline{\text{IRQ4}}$	NC

131	A6	P73/TMO1/$\overline{CS7}$	P73/TMO1	NC
132	B6	P72/TMO0/$\overline{CS6}$	P72/TMO0	NC
133	C5	P71/TMRI23/ TMCI23/$\overline{CS5}$	P71/TMRI23/ TMCI23	NC
134	A5	P70/TMRI01/ TMCI01/$\overline{CS4}$	P70/TMRI01/ TMCI01	NC
135	B5	PG4/$\overline{CS0}$	PG4	NC
136	D5	HRxD*[1]	HRxD*[1]	NC
		PG3/$\overline{Rx}$/$\overline{CS1}$*[2]	PG3/$\overline{Rx}$*[2]	
		PG3/$\overline{CS1}$*[3]	PG3*[3]	
137	A4	HTxD*[1]	HTxD*[1]	NC
		PG2/$\overline{Tx}$/$\overline{CS2}$*[2]	PG2/$\overline{Tx}$*[2]	
		PG2/$\overline{CS2}$*[3]	PG2*[3]	
138	B4	PG1/$\overline{CS3}$/$\overline{IRQ7}$	PG1/$\overline{IRQ7}$	NC
139	C4	PG0/$\overline{IRQ6}$	PG0/$\overline{IRQ6}$	NC
140	A3	PE0/D0	PE0	NC
141	D4	PE1/D1	PE1	NC
142	B3	PE2/D2	PE2	NC
143	A2	PE3/D3	PE3	Vcc
144	B2	PE4/D4	PE4	Vss

Notes: 1. Symbol name for the H8S/2556 Group
2. Symbol name for the H8S/2552 Group
3. Symbol name for the H8S/2506 Group
4. Available only in the H8S/2552 Group and H8S/2506 Group.
5. NC pins should be left open.

	P1Vcc	14, 84	F3, G3, G4, H3, H4, K12	I	Power supply pin for ports indicated that its pow supplied by P1Vcc (see table 1.1).
	P2Vcc	118	C9, C10, D9, D10	I	Pins for connecting a capacitor to stabilize the step-down voltage. Power supply pin for ports indicated that its pow supplied by P2Vcc (see table 1.1).
	VCL	88	J13	O	Pin for connecting the on-chip step-down powe a capacitor for voltage stabilization. Must not b connected to a power supply. A capacitor of 0.4 be connected between this pin and Vss. (Place the pin.)
	VSS	12, 54, 86, 94, 117	G2, G1, F2, F1, E1, R7, R6, P8, P7, K15, K14, H15, H14, G14, G13, G12, F13, F12, B10, A14, A13, A12, A11, A10	I	Ground pins. Connect this pin to the system po (0V).
Clock	XTAL	93	G15	I	For connection to a crystal resonator. For exam connecting crystal resonator and external clock section 21, Clock Pulse Generator.
	EXTAL	95	F15	I	For connection to a crystal resonator or a ceram resonator. This pin can be also used for extern input. For examples of connecting crystal reson external clock input, see section 21, Clock Puls Generator.

Operating mode control	MD1	98	E15		Sets the operating mode. Input at these pins should not be changed during operation. Be sure to fix the level of the mode pins (MD2 to MD0) by pull-down or pull-up except for mode changing.
	MD0	97	E12		
System control	$\overline{\text{RES}}$*[1]	100	E13	I	Reset input pin. When this pin is low, this LSI enters the power-on reset state.
	$\overline{\text{MRES}}$*[1]	130	C6	I	Reset input pin. When this pin is low, this LSI enters the manual reset state.
	$\overline{\text{STBY}}$*[1]	99	E14	I	When this pin is low, a transition is made to hardware standby mode.
	$\overline{\text{BREQ}}$	77	N15	I	Indicates that an external bus master is requesting mastership.
	$\overline{\text{BACK}}$	78	M14	O	Indicates that the bus is released to an external bus master.
	TEST*[1]	87	J12	I	Test pin. Connect to a Vss.
Interrupts	NMI*[1]	91	H12	I	Nonmaskable interrupt pin. If this pin is not used, it should be fixed-high.
	$\overline{\text{IRQ7}}$	138	B4	I	These pins request maskable interrupts.
	$\overline{\text{IRQ6}}$	139	C4		
	$\overline{\text{IRQ5}}$	121	D8		
	$\overline{\text{IRQ4}}$	124	B8		
	$\overline{\text{IRQ3}}$	80	M15		
	$\overline{\text{IRQ2}}$	77	N15		
	$\overline{\text{IRQ1}}$	110	C13		
	$\overline{\text{IRQ0}}$	112	C12		

			J1, J3, J4, H2, H1, F4		
Data bus	D15 to D0	11 to 1, 144 to 140	E2, E3, D1, E4, D2, C1, D3, C2, B1, C3, A1, B2, A2, B3, D4, A3	I/O	Bi-directional bus.
Bus control	$\overline{CS7}$	131	A6	O	Chip select signals for areas 7 to 0.
	$\overline{CS6}$	132	B6		Pins $\overline{CS2}$ and $\overline{CS1}$ are not supported by the H Group.
	$\overline{CS5}$	133	C5		
	$\overline{CS4}$	134	A5		
	$\overline{CS3}$	138	B4		
	$\overline{CS2}$	137	A4		
	$\overline{CS1}$	136	D5		
	$\overline{CS0}$	135	B5		
	$\overline{AS}$	83	L15	O	Indicates that data output on the address bus i when this pin is a low level.
	$\overline{RD}$	82	L14	O	Indicates that an access to the external addres in progress when this pin is a low level.
	$\overline{HWR}$	81	L13	O	Strobe signal. Indicates that data on the upper to D8) of the data bus is valid during a write ac
	$\overline{LWR}$	80	M15	O	Strobe signal. Indicates that data on the upper D0) of the data bus is valid during a write acce

	TIOCB0	115	C11		output compare output, or PWM output.
	TIOCC0	114	D11		
	TIOCD0	113	B12		
	TIOCA1	112	C12	I/O	Pins for the TGRA_1 and TGRB_1 input capture output compare output, or PWM output.
	TIOCB1	111	B13		
	TIOCA2	110	C13	I/O	Pins for the TGRA_2 and TGRB_2 input capture output compare output, or PWM output.
	TIOCB2	109	A15		
	TIOCA3	101	D15	I/O	Pins for the TGRA_3 and TGRD_3 input capture output compare output, or PWM output.
	TIOCB3	102	D14		
	TIOCC3	103	D13		
	TIOCD3	104	C15		
	TIOCA4	105	D12	I/O	Pins for the TGRA_4 and TGRB_4 input capture output compare output, or PWM output.
	TIOCB4	106	C14		
	TIOCA5	107	B15	I/O	Pins for the TGRA_5 and TGRB_5 input capture output compare output, or PWM output.
	TIOCB5	108	B14		
8-bit timer	TMO3	129	D6, C6, A6, B6	O	Compare-match output pins
	TMO2	130			
	TMO1	131			
	TMO0	132			
	TMCI23	133	C5	I	Pins for external clock input to the counter
	TMCI01	134	A5		
	TMRI23	133	C5	I	Counter reset input pins.
	TMRI01	134	A5		

smart card interface	TxD0	126	C7		
	RxD4	120	B9	I	Data input pins
	RxD3	128	B7		
	RxD2	75	N14		
	RxD1	122	C8		
	RxD0	125	D7		
	SCK4	121	D8	I/O	Clock input/output pins
	SCK3	129	D6		SCK4 and SCK1 are NMOS push-pull outputs
	SCK2	76	M13		
	SCK1	121	D8		
	SCK0	124	B8		
I^2C bus interface 2 (IIC2)	SCL1	123	A8	I/O	I^2C clock input/output pins. These pins are cap
	SCL0	121	D8		driving bus. Pin SCL0 is an NMOS open-drain
	SDA1	124	B8	I/O	I^2C data input/output pins. These pins are capa
	SDA0	122	C8		driving bus. Pin SDA0 is an NMOS open-drain
A/D converter	AN15 to AN0	55 to 66, 68 to 71	M8, N8, R8, M9, N9, R9, P9, M10, N10, R10, P10, N11, P11, M11, P12, N12	I	Analog input pins for A/D converter.
	$\overline{\text{ADTRG}}$	80	M15	I	Pin for input of an external trigger to start A/D
D/A converter	DA1	55	M8	O	Analog output pins for the D/A converter.
	DA0	56	N8		

IEBus™ controller (IEB)	$\overline{Tx}$	137	A4	O	Transmit data output pin for the IEB. (Supported only by the H8S/2552 Group.)
	$\overline{Rx}$	136	D5	I	Receive data input pin for the IEB. (Supported only by the H8S/2552 Group.)
Controller area network (HCAN)	HTxD	137	A4	O	Pin for CAN bus transmission. (Supported only by the H8S/2556 Group.)
	HRxD	136	D5	I	Pin for CAN bus reception. (Supported only by the H8S/2556 Group.)
I/O ports	P17 to P10	109 to 116	A15, C13, B13, C12, B12, D11, C11, B11	I/O	8-bit I/O pins
	P27 to P20	108 to 101	B14, B15, C14, D12, C15, D13, D14, D15	I/O	8-bit I/O pins
	P37 to P30	119 to 126	A9, B9, D8, C8, A8, B8, D7, C7	I/O	8-bit I/O pins. Pins P34 and P35 are NMOS push-pull outputs.
	P47 to P40	63 to 66, 68 to 71	N10, R10, P10, N11, P11, M11, P12, N12	I	8-bit input pins
	P52 to P50	76 to 74	M13, N14, P15	I/O	3-bit I/O pins

PA7 to PA0	37 to 30	R1, P2, P1, N2, M4, N1, M3, M2	I/O	8-bit I/O pins
PB7 to PB0	29 to 22	M1, L4, L2, L1, L3, K2, K1, K3	I/O	8-bit I/O pins
PC7 to PC0	21 to 15, 13	K4, J2, J1, J3, J4, H2, H1, F4	I/O	8-bit I/O pins
PD7 to PD0	11 to 4	E2, E3, D1, E4, D2, C1, D3, C2	I/O	8-bit I/O pins
PE7 to PE0	3 to 1, 144 to 140	B1, C3, A1, B2, A2, B3, D4, A3	I/O	8-bit I/O pins
PF7 to PF0	85, 83 to 77	K13, L15, L14, L13, M15, L12, M14, N15	I/O	8-bit I/O pins
PG4 to PG0	135 to 139	B5, D5, A4, B4, C4	I/O	5-bit I/O pins Pins PG3 and PG2 are not supported by the H Group.

Notes: 1. Countermeasure against noise should be executed or may result in malfunctio
2. Available only in the H8S/2552 Group and H8S/2506 Group.

- Upward-compatible with H8/300 and H8/300H CPU
  - Can execute H8/300 and H8/300H CPU object programs
- General-register architecture
  - Sixteen 16-bit general registers also usable as sixteen 8-bit registers or eight 32-b
- 65 basic instructions
  - 8/16/32-bit arithmetic and logic instructions
  - Multiply and divide instructions
  - Powerful bit-manipulation instructions
- Eight addressing modes
  - Register direct [Rn]
  - Register indirect [@ERn]
  - Register indirect with displacement [@(d:16,ERn) or @(d:32,ERn)]
  - Register indirect with post-increment or pre-decrement [@ERn+ or @–ERn]
  - Absolute address [@aa:8, @aa:16, @aa:24, or @aa:32]
  - Immediate [#xx:8, #xx:16, or #xx:32]
  - Program-counter relative [@(d:8,PC) or @(d:16,PC)]
  - Memory indirect [@@aa:8]
- 16-Mbyte address space
  - Program: 16 Mbytes
  - Data: 16 Mbytes
- High-speed operation
  - All frequently-used instructions execute in one or two states
  - 8/16/32-bit register-register add/subtract : 1 state
  - 8 × 8-bit register-register multiply : 12 states
  - 16 ÷ 8-bit register-register divide : 12 states

2.1.1 Differences between H8S/2600 CPU and H8S/2000 CPU

The differences between the H8S/2600 CPU and the H8S/2000 CPU are shown below.

- Register configuration
  - The MAC register is supported by the H8S/2600 CPU only.
- Basic instructions
  - The four instructions MAC, CLRMAC, LDMAC, and STMAC are supported by t H8S/2600 CPU only.
- The number of execution states of the MULXU and MULXS instructions;

		Execution States	
Instruction	**Mnemonic**	**H8S/2600**	**H8S/2000**
MULXU	MULXU.B Rs, Rd	3	12
	MULXU.W Rs, ERd	4	20
MULXS	MULXS.B Rs, Rd	4	13
	MULXS.W Rs, ERd	5	21

In addition, there are differences in address space, CCR and EXR register functions, and down modes, etc., depending on the model.

- Enhanced addressing
  - The addressing modes have been enhanced to make effective use of the 16-Mbyt space.
- Enhanced instructions
  - Addressing modes of bit-manipulation instructions have been enhanced.
  - Signed multiply and divide instructions have been added.
  - Two-bit shift instructions have been added.
  - Instructions for saving and restoring multiple registers have been added.
  - A test and set instruction has been added.
- Higher speed
  - Basic instructions execute twice as fast.

2.1.3 Differences from H8/300H CPU

In comparison to the H8/300H CPU, the H8S/2000 CPU has the following enhancement

- Additional control register
  - One 8-bit control registers have been added.
- Enhanced instructions
  - Addressing modes of bit-manipulation instructions have been enhanced.
  - Two-bit shift instructions have been added.
  - Instructions for saving and restoring multiple registers have been added.
  - A test and set instruction has been added.
- Higher speed
  - Basic instructions execute twice as fast.

Linear access is provided to a maximum address space of 64 kbytes.

- Extended Registers (En)

  The extended registers (E0 to E7) can be used as 16-bit registers, or as the upper 16-b segments of 32-bit registers. When En is used as a 16-bit register it can contain any va when the corresponding general register (Rn) is used as an address register. If the gen register is referenced in the register indirect addressing mode with pre-decrement (@- post-increment (@Rn+) and a carry or borrow occurs, however, the value in the corre extended register (En) will be affected.

- Instruction Set

  All instructions and addressing modes can be used. Only the lower 16 bits of effective addresses (EA) are valid.

- Exception Vector Table and Memory Indirect Branch Addresses

  In normal mode the top area starting at H'0000 is allocated to the exception vector tab branch address is stored per 16 bits. Figure 2.1 shows the structure of the exception ve table in normal mode. For details of the exception vector table, see section 4, Exceptio Handling.

  The memory indirect addressing mode (@@aa:8) employed in the JMP and JSR instr uses an 8-bit absolute address included in the instruction code to specify a memory op that contains a branch address. In normal mode the operand is a 16-bit word operand, providing a 16-bit branch address. Branch addresses can be stored in the top area from to H'00FF. Note that this area is also used for the exception vector table.

- Stack Structure

  When the program counter (PC) is pushed onto the stack in a subroutine call, and the condition-code register (CCR) and extended control register (EXR) are pushed onto th in exception handling, they are stored as shown in figure 2.2. EXR is not pushed onto in interrupt control mode 0. For details, see section 4, Exception Handling.

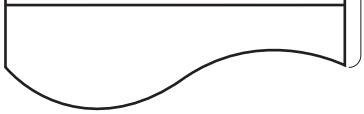
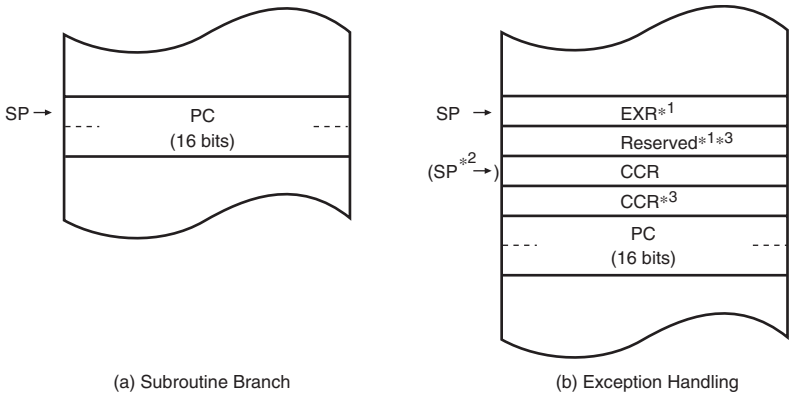


Figure 2.1 Exception Vector Table (Normal Mode)



Notes: 1. When EXR is not used it is not stored on the stack.
2. SP when EXR is not used.
3. Ignored when returning.

Figure 2.2 Stack Structure in Normal Mode

2.2.2 Advanced Mode

- Address Space
  Linear access is provided to a maximum 16-Mbyte address space.
- Extended Registers (En)

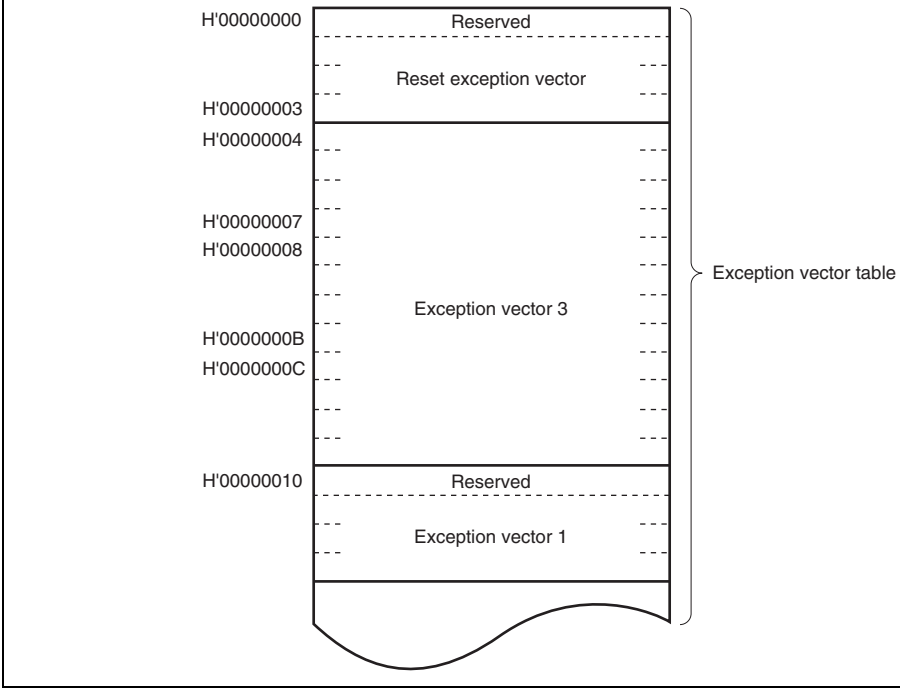
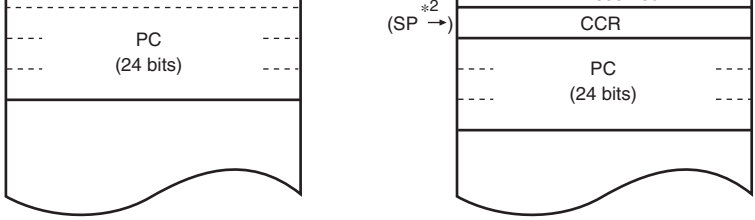


Figure 2.3 Exception Vector Table (Advanced Mode)

The memory indirect addressing mode (@@aa:8) employed in the JMP and JSR instr
uses an 8-bit absolute address included in the instruction code to specify a memory op
that contains a branch address. In advanced mode, the operand is a 32-bit longword op
providing a 32-bit branch address. The upper 8 bits of these 32 bits is a reserved area
regarded as H'00. Branch addresses can be stored in the area from H'00000000 to H'0
Note that the first part of this range is also the exception vector table.



(a) Subroutine Branch

(b) Exception Handling

Notes: 1. When EXR is not used it is not stored on the stack.
2. SP when EXR is not used.
3. Ignored when returning.

Figure 2.4 Stack Structure in Advanced Mode

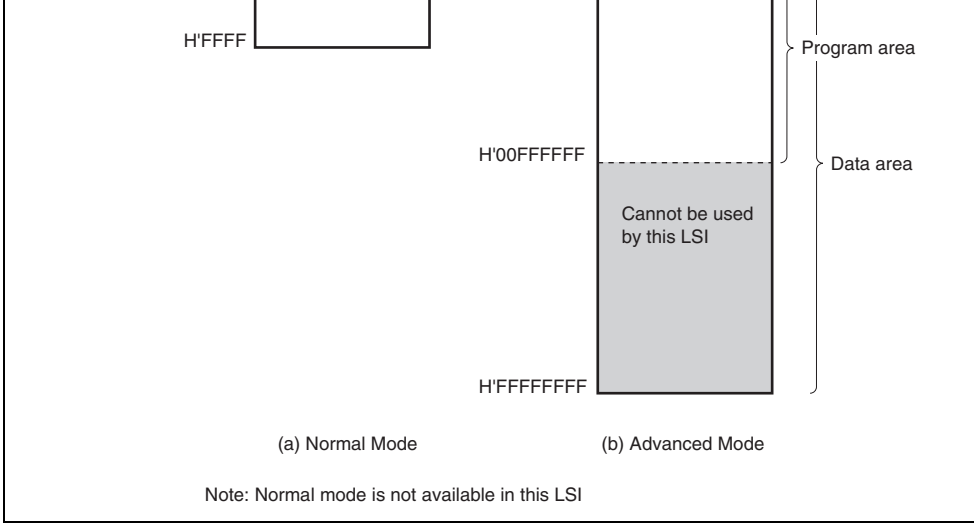
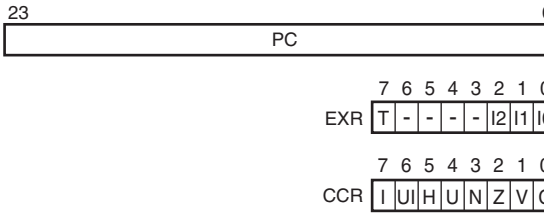


Figure 2.5 Memory Map

ER3	E3	R3H	R3L
ER4	E4	R4H	R4L
ER5	E5	R5H	R5L
ER6	E6	R6H	R6L
ER7 (SP)	E7	R7H	R7L

Control Registers (CR)



Legend:

SP	:Stack pointer	H	:Half-carry flag
PC	:Program counter	U	:User bit
EXR	:Extended control register	N	:Negative flag
T	:Trace bit	Z	:Zero flag
I2 to I0	:Interrupt mask bits	V	:Overflow flag
CCR	:Condition-code register	C	:Carry flag
I	:Interrupt mask bit		
UI	:User bit or interrupt mask bit*		

Note: * The interrupt mask bit is not available in this LSI.

Figure 2.6 CPU Registers

(R0 to R7). These registers are functionally equivalent, providing a maximum sixteen 16-
registers. The E registers (E0 to E7) are also referred to as extended registers.

The R registers divide into 8-bit general registers designated by the letters RH (R0H to R
RL (R0L to R7L). These registers are functionally equivalent, providing a maximum sixt
2egisters.

The usage of each register can be selected independently.

General register ER7 has the function of stack pointer (SP) in addition to its general-regis
function, and is used implicitly in exception handling and subroutine calls. Figure 2.8 sho
stack.

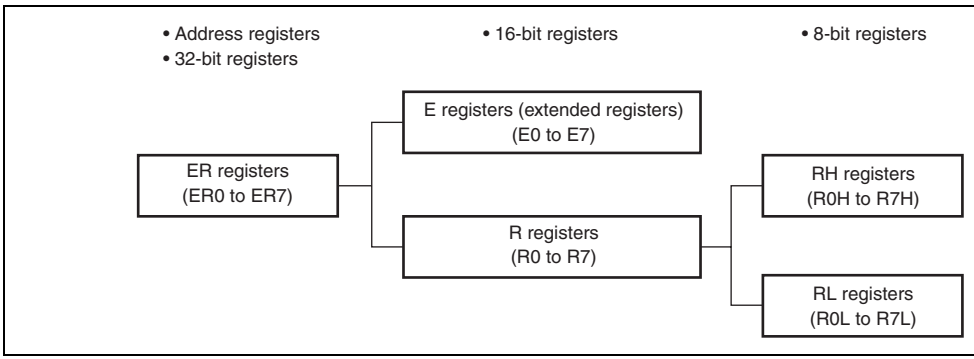


Figure 2.7 Usage of General Registers

Figure 2.8 Stack Status

2.4.2 Program Counter (PC)

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

2.4.3 Extended Control Register (EXR)

EXR is an 8-bit register that manipulates the LDC, STC, ANDC, ORC, and XORC instr
When these instructions except for the STC instruction is executed, all interrupts includi
will be masked for three states after execution is completed.

Bit	Bit Name	Initial Value	R/W	Description
7	T	0	R/W	Trace Bit When this bit is set to 1, a trace exceptio generated each time an instruction is ex When this bit is cleared to 0, instructions executed in sequence.
6 to 3	—	All 1	—	Reserved These bits are always read as 1.
2	I2	1	R/W	These bits designate the interrupt mask
1	I1	1	R/W	7). For details, see section 5, Interrupt C
0	I0	1	R/W	

				Masks interrupts other than NMI when se NMI is accepted regardless of the I bit se I bit is set to 1 by hardware at the start of exception-handling sequence. For details section 5, Interrupt Controller.
6	UI	Undefined	R/W	User Bit or Interrupt Mask Bit Can be written and read by software usin LDC, STC, ANDC, ORC, and XORC instr This bit cannot be used as an interrupt m this LSI.
5	H	Undefined	R/W	Half-Carry Flag When the ADD.B, ADDX.B, SUB.B, SUB CMP.B, or NEG.B instruction is executed is set to 1 if there is a carry or borrow at b cleared to 0 otherwise. When the ADD.W SUB.W, CMP.W, or NEG.W instruction is executed, the H flag is set to 1 if there is borrow at bit 11, and cleared to 0 otherwi When the ADD.L, SUB.L, CMP.L, or NEG instruction is executed, the H flag is set to there is a carry or borrow at bit 27, and cl 0 otherwise.
4	U	Undefined	R/W	User Bit Can be written and read by software usin LDC, STC, ANDC, ORC, and XORC instr
3	N	Undefined	R/W	Negative Flag Stores the value of the most significant bi as a sign bit.

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

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

2.4.5 Initial Values of CPU Registers

Reset exception handling loads the CPU's program counter (PC) from the vector table, c
trace bit in EXR to 0, and sets the interrupt mask bits in CCR and EXR to 1. The other C
and the general registers are not initialized. In particular, the stack pointer (ER7) is not i
The stack pointer should therefore be initialized by an MOV.L instruction executed imm
after a reset.

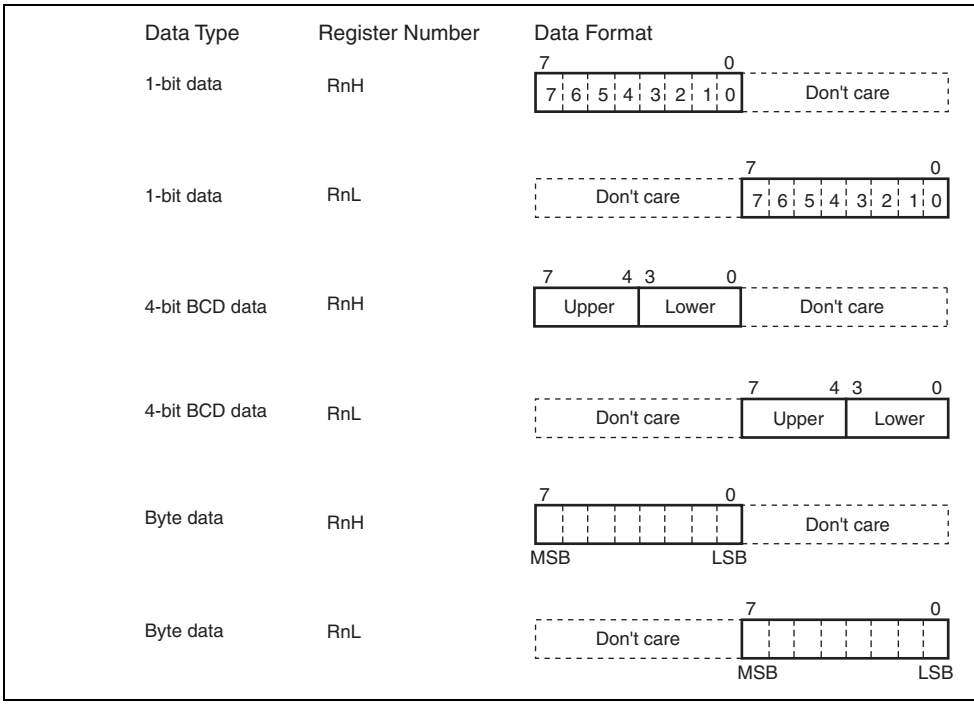
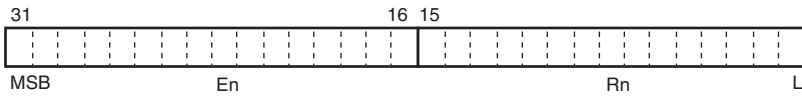


Figure 2.9 General Register Data Formats (1)



Legend:

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

Figure 2.9 General Register Data Formats (2)

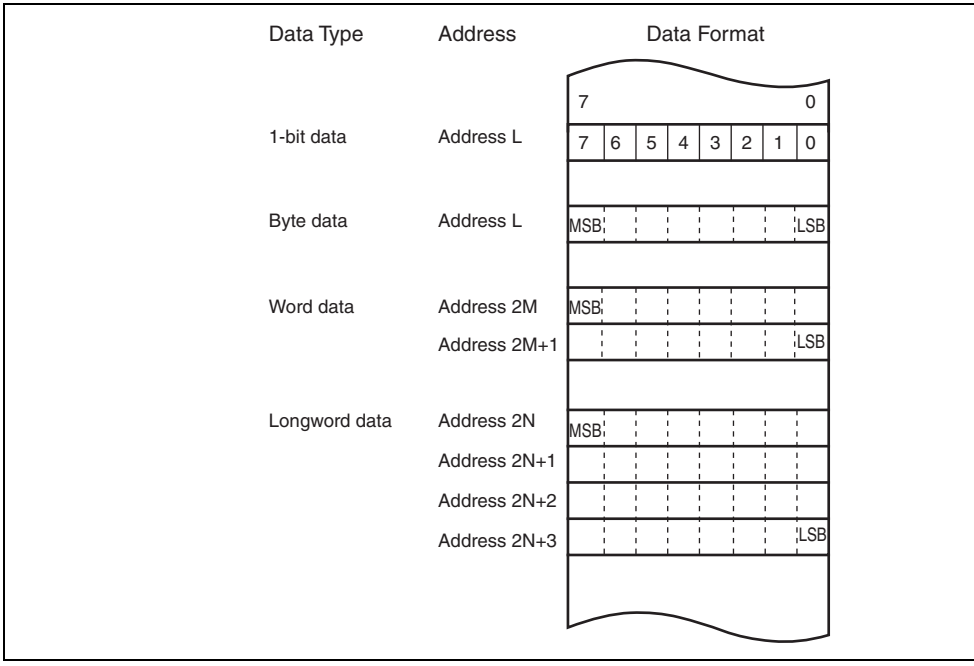


Figure 2.10 Memory Data Formats

2.6 Instruction Set

The H8S/2000 CPU has 65 types of instructions. The instructions are classified by functi
table 2.1.

	ADDS, SUBS	L
	MULXU, DIVXU, MULXS, DIVXS	B/W
	EXTU, EXTS	W/L
	TAS*[4]	B
Logic operations	AND, OR, XOR, NOT	B/W/L
Shift	SHAL, SHAR, SHLL, SHLR, ROTL, ROTR, ROTXL, ROTXR	B/W/L
Bit manipulation	BSET, BCLR, BNOT, BTST, BLD, BILD, BST, BIST, BAND, BIAND, BOR, BIOR, BXOR, BIXOR	B
Branch	Bcc*[2], JMP, BSR, JSR, RTS	—
System control	TRAPA, RTE, SLEEP, LDC, STC, ANDC, ORC, XORC, NOP	—
Block data transfer	EEPMOV	—

Legend:

B: Byte

W: Word

L: Longword

Notes: 1. POP.W Rn and PUSH.W Rn are identical to MOV.W @SP+, Rn and MOV.W @-SP. POP.L ERn and PUSH.L ERn are identical to MOV.L @SP+, ERn and ERn, @-SP.

2. Bcc is the general name for conditional branch instructions.
3. Cannot be used in this LSI.
4. Only register ER0, ER1, ER4, or ER5 should be used when using the TAS ins
5. Only register ER0 to ER6 should be used when using the STM/LDM instructio

ERn	General register (32-bit register)
(EAd)	Destination operand
(EAs)	Source operand
EXR	Extended control register
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
#IMM	Immediate data
disp	Displacement
+	Addition
–	Subtraction
×	Multiplication
÷	Division
∧	Logical AND
∨	Logical OR
⊕	Logical XOR

MOV	B/W/L	(EAs) → Rd, Rs → (EAd) Moves data between two general registers or between a gene register and memory, or moves immediate data to a general re
MOVFPE	B	Cannot be used in this LSI.
MOVTPE	B	Cannot be used in this LSI.
POP	W/L	@SP+ → Rn Pops a general register from the stack. POP.W Rn is identical MOV.W @SP+, Rn. POP.L ERn is identical to MOV.L @SP+,
PUSH	W/L	Rn → @–SP Pushes a general register onto the stack. PUSH.W Rn is ident MOV.W Rn, @–SP. PUSH.L ERn is identical to MOV.L ERn,
LDM*[2]	L	@SP+ → Rn (register list) Pops two or more general registers from the stack.
STM*[2]	L	Rn (register list) → @–SP Pushes two or more general registers onto the stack.

Notes: 1. Refers to the operand size.
B: Byte
W: Word
L: Longword
2. Only register ER0 to ER6 should be used when using the STM/LDM instructio

DEC		Increments or decrements a general register by 1 or 2. (Byte op can be incremented or decremented by 1 only.)
ADDS SUBS	L	Rd ± 1 → Rd, Rd ± 2 → Rd, Rd ± 4 → Rd Adds or subtracts the value 1, 2, or 4 to or from data in a 32-bit
DAA DAS	B	Rd decimal adjust → Rd Decimal-adjusts an addition or subtraction result in a general re referring to the CCR to produce 4-bit BCD data.
MULXU	B/W	Rd × Rs → Rd Performs unsigned multiplication on data in two general registe 8 bits × 8 bits → 16 bits or 16 bits × 16 bits → 32 bits.
MULXS	B/W	Rd × Rs → Rd Performs signed multiplication on data in two general registers: bits × 8 bits → 16 bits or 16 bits × 16 bits → 32 bits.
DIVXU	B/W	Rd ÷ Rs → Rd Performs unsigned division on data in two general registers: eit bits ÷ 8 bits → 8-bit quotient and 8-bit remainder or 32 bits ÷ 16 16-bit quotient and 16-bit remainder.

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

		Takes the two's complement (arithmetic complement) of data i general register.
EXTU	W/L	Rd (zero extension) → Rd Extends the lower 8 bits of a 16-bit register to word size, or the bits of a 32-bit register to longword size, by padding with zeros left.
EXTS	W/L	Rd (sign extension) → Rd Extends the lower 8 bits of a 16-bit register to word size, or the bits of a 32-bit register to longword size, by extending the sign
TAS*[2]	B	@ERd – 0, 1 → (<bit 7> of @ERd) Tests memory contents, and sets the most significant bit (bit 7

Notes: 1. Refers to the operand size.
B: Byte
W: Word
L: Longword
2. Only register ER0, ER1, ER4, or ER5 should be used when using the TAS ins

NOT	B/W/L	~ Rd → Rd Takes the one's complement of general register contents.

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

Table 2.6 Shift Instructions

Instruction	Size*	Function
SHAL SHAR	B/W/L	Rd (shift) → Rd Performs an arithmetic shift on general register contents. 1-bit or 2-bit shifts are possible.
SHLL SHLR	B/W/L	Rd (shift) → Rd Performs a logical shift on general register contents. 1-bit or 2-bit shifts are possible.
ROTL ROTR	B/W/L	Rd (rotate) → Rd Rotates general register contents. 1-bit or 2-bit rotations are possible.
ROTXL ROTXR	B/W/L	Rd (rotate) → Rd Rotates general register contents through the carry flag. 1-bit or 2-bit rotations are possible.

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

BNOT	B	(<bit-No.> of <EAd>) → (<bit-No.> of <EAd>) Inverts a specified bit in a general register or memory operand number is specified by 3-bit immediate data or the lower three general register.
BTST	B	~ (<bit-No.> of <EAd>) → Z Tests a specified bit in a general register or memory operand a or clears the Z flag accordingly. The bit number is specified by immediate data or the lower three bits of a general register.
BAND	B	C ∧ (<bit-No.> of <EAd>) → C ANDs the carry flag with a specified bit in a general register or operand and stores the result in the carry flag.
BIAND	B	C ∧ ← (<bit-No.> of <EAd>) → C ANDs the carry flag with the inverse of a specified bit in a gen register or memory operand and stores the result in the carry f The bit number is specified by 3-bit immediate data.
BOR	B	C ∨ (<bit-No.> of <EAd>) → C ORs the carry flag with a specified bit in a general register or m operand and stores the result in the carry flag.
BIOR	B	C ∨ ← (<bit-No.> of <EAd>) → C ORs the carry flag with the inverse of a specified bit in a gener or memory operand and stores the result in the carry flag. The bit number is specified by 3-bit immediate data.

Note: * Refers to the operand size.
B: Byte

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

Note: * Refers to the operand size.
B: Byte

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

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

		Transfers CCR or EXR contents to a general register or memor Although CCR and EXR are 8-bit registers, word-size transfers performed between them and memory. The upper 8 bits are va
ANDC	B	CCR ∧ #IMM → CCR, EXR ∧ #IMM → EXR Logically ANDs the CCR or EXR contents with immediate data.
ORC	B	CCR ∨ #IMM → CCR, EXR ∨ #IMM → EXR Logically ORs the CCR or EXR contents with immediate data.
XORC	B	CCR ⊕ #IMM → CCR, EXR ⊕ #IMM → EXR Logically XORs the CCR or EXR contents with immediate data.
NOP	—	PC + 2 → PC Only increments the program counter.

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

	else next;
	Transfers a data block. Starting from the address set in ER5, t data for the number of bytes set in R4L or R4 to the address lo in ER6.
	Execution of the next instruction begins as soon as the transfe completed.

2.6.2 Basic Instruction Formats

This LSI instructions consist of 2-byte (1-word) units. An instruction consists of an oper (op field), a register field (r field), an effective address extension (EA field), and a condi (cc).

Figure 2.11 shows examples of instruction formats.

- Operation Field

  Indicates the function of the instruction, the addressing mode, and the operation to b out on the operand. The operation field always includes the first four bits of the instr Some instructions have two operation fields.
- Register Field

  Specifies a general register. Address registers are specified by 3 bits, and data registe bits or 4 bits. Some instructions have two register fields. Some have no register field
- Effective Address Extension

  8, 16, or 32 bits specifying immediate data, an absolute address, or a displacement.
- Condition Field

  Specifies the branching condition of Bcc instructions.

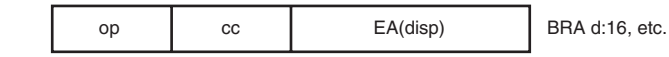


Figure 2.11 Instruction Formats (Examples)

2.7 Addressing Modes and Effective Address Calculation

The H8S/2000 CPU supports the eight addressing modes listed in table 2.11. Each instruc
a subset of these addressing modes. Arithmetic and logic instructions can use the register
and immediate modes. Data transfer instructions can use all addressing modes except pro
counter relative and memory indirect. Bit manipulation instructions use register direct, re
indirect, or the absolute addressing mode to specify an operand, and register direct (BSET
BNOT, and BTST instructions) or immediate (3-bit) addressing mode to specify a bit nur
the operand.

7	Program-counter relative	@(d:8,PC)/@(d:16,PC)
8	Memory indirect	@@aa:8

2.7.1 Register Direct—Rn

The register field of the instruction specifies an 8-, 16-, or 32-bit general register contain operand. R0H to R7H and R0L to R7L can be specified as 8-bit registers. R0 to R7 and can be specified as 16-bit registers. ER0 to ER7 can be specified as 32-bit registers.

2.7.2 Register Indirect—@ERn

The register field of the instruction code specifies an address register (ERn) which conta address of the operand on memory. If the address is a program instruction address, the l bits are valid and the upper 8 bits are all assumed to be 0 (H'00).

2.7.3 Register Indirect with Displacement—@(d:16, ERn) or @(d:32, ERn)

A 16-bit or 32-bit displacement contained in the instruction is added to an address regist specified by the register field of the instruction, and the sum gives the address of a mem operand. A 16-bit displacement is sign-extended when added.

2.7.4 Register Indirect with Post-Increment or Pre-Decrement—@ERn+ or @-

Register indirect with post-increment—@ERn+: The register field of the instruction specifies an address register (ERn) which contains the address of a memory operand. Af operand is accessed, 1, 2, or 4 is added to the address register contents and the sum is st address register. The value added is 1 for byte access, 2 for word transfer instruction, or

The instruction code contains the absolute address of a memory operand. The absolute ad
may be 8 bits long (@aa:8), 16 bits long (@aa:16), 24 bits long (@aa:24), or 32 bits long
(@aa:32). Table 2.12 indicates the accessible absolute address ranges.

To access data, the absolute address should be 8 bits (@aa:8), 16 bits (@aa:16), or 32 bits
(@aa:32) long. For an 8-bit absolute address, the upper 24 bits are all assumed to be 1 (H
For a 16-bit absolute address the upper 16 bits are a sign extension. A 32-bit absolute add
access the entire address space.

A 24-bit absolute address (@aa:24) indicates the address of a program instruction. The up
bits are all assumed to be 0 (H'00).

Table 2.12 Absolute Address Access Ranges

Absolute Address		Normal Mode*	Advanced Mode
Data address	8 bits (@aa:8)	H'FF00 to H'FFFF	H'FFFF00 to H'FFF
	16 bits (@aa:16)	H'0000 to H'FFFF	H'000000 to H'007 H'FF8000 to H'FFF
	32 bits (@aa:32)		H'000000 to H'FFF
Program instruction address	24 bits (@aa:24)		

Note: * Normal mode is not available in this LSI.

2.7.6 Immediate—#xx:8, #xx:16, or #xx:32

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

instruction, so the possible branching range is –126 to +128 bytes (–63 to +64 words) or
+32768 bytes (–16383 to +16384 words) from the branch instruction. The resulting valu
be an even number.

2.7.8 Memory Indirect—@@aa:8

This mode can be used by the JMP and JSR instructions. The instruction code contains a
absolute address specifying a memory operand. This memory operand contains a branch
The upper bits of the absolute address are all assumed to be 0, so the address range is 0 t
(H'0000 to H'00FF in normal mode*, H'000000 to H'0000FF in advanced mode). In nor
the memory operand is a word operand and the branch address is 16 bits long. In advanc
the memory operand is a longword operand, the first byte of which is assumed to be 0 (H

Note that the first part of the address range is also the exception vector area. For further
see section 4, Exception Handling.

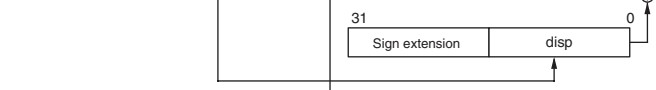
If an odd address is specified in word or longword memory access, or as a branch addres
least significant bit is regarded as 0, causing data to be accessed or instruction code to be
at the address preceding the specified address. (For further information, see section 2.5.2
Data Formats.)

Note: * Normal mode is not available in this LSI.

Figure 2.12 Branch Address Specification in Memory Indirect Mode

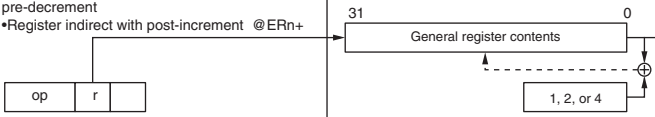
2.7.9 Effective Address Calculation

Table 2.13 indicates how effective addresses are calculated in each addressing mode. In n
mode the upper 8 bits of the effective address are ignored in order to generate a 16-bit ad

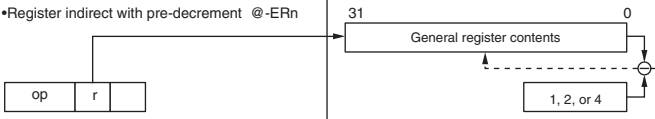


4 Register indirect with post-increment or pre-decrement

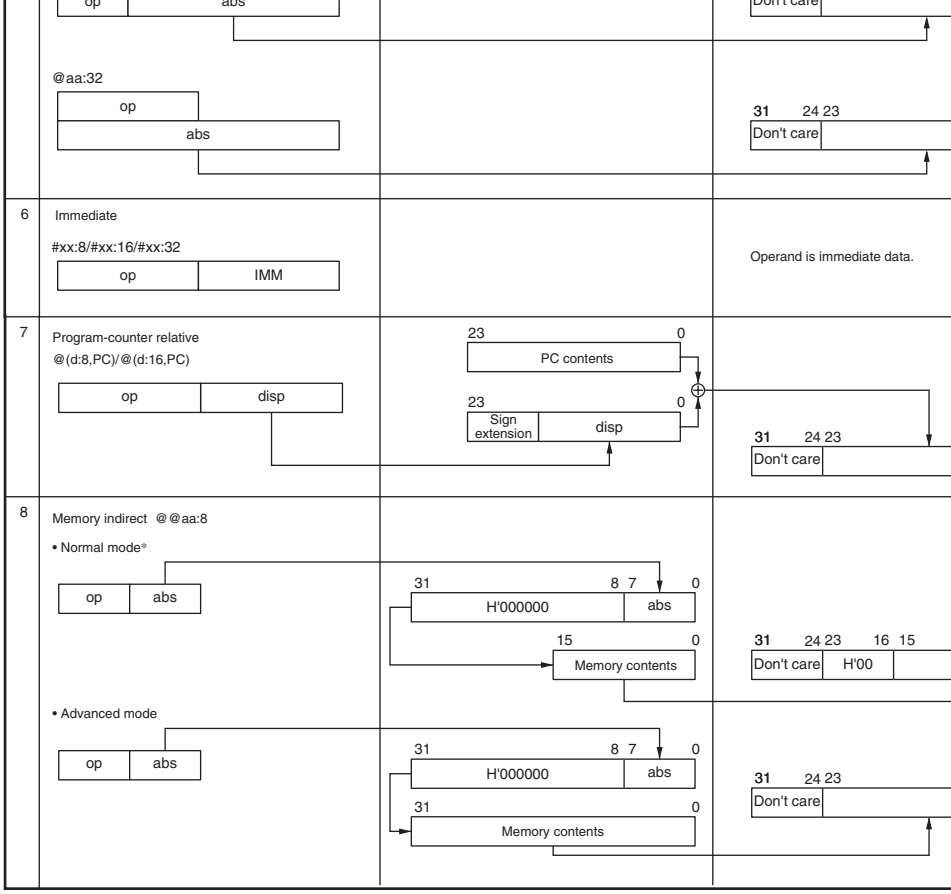
•Register indirect with post-increment @ERn+



•Register indirect with pre-decrement @-ERn



Operand Size	Offset
Byte	1
Word	2
Longword	4



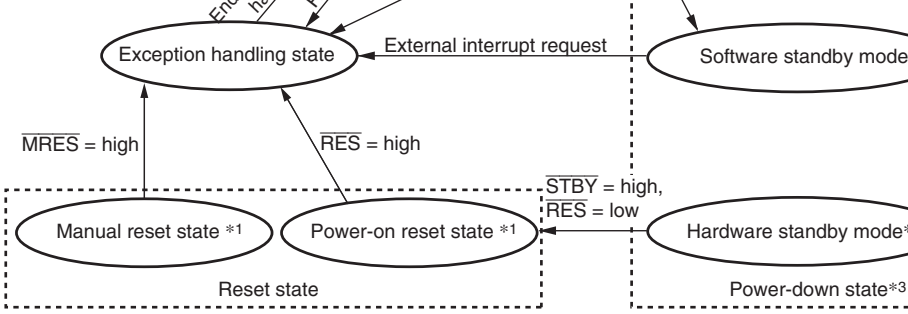
Note: * Normal mode is not available in this LSI.

The reset state can also be entered by a watchdog timer overflow.

- Exception-Handling State
  The exception-handling state is a transient state that occurs when the CPU alters the
  processing flow due to an exception source, such as a reset, trace, interrupt, or trap i
  The CPU fetches a start address (vector) from the exception vector table and branche
  address. For further details, see section 4, Exception Handling.
- Program Execution State
  In this state, the CPU executes program instructions in sequence.
- Bus-Released State
  In a product which has a bus master other than the CPU, such as a data transfer contr
  (DTC), the bus-released state occurs when the bus has been released in response to a
  request from a bus master other than the CPU.

  While the bus is released, the CPU halts operations.
- Power-down State

  This is a power-down state in which the CPU stops operating. The program stop stat
  when a SLEEP instruction is executed or the CPU enters hardware standby mode. Fo
  details, see section 22, Power-Down Modes.



Notes: 1. From any state except hardware standby mode, a transition to the reset state occur whenever RES goes low. A transition can also be made to the reset state when the watchdog timer overflows.
2. From any state, a transition to hardware standby mode occurs when STBY goes low
3. Apart from these states, there is also watch mode. For details, see section 22, Power-Down Modes.

Figure 2.13 State Transitions

With the STM or LDM instruction, the ER7 register is used as the stack pointer, and thu
be used as a register that allows save (STM) or restore (LDM) operation.

With a single STM or LDM instruction, two to four registers can be saved or restored. T
available registers are as follows:

For two registers: ER0 and ER1, ER2 and ER3, or ER4 and ER5

For three registers: ER0 to ER2, or ER4 to ER6

For four registers: ER0 to ER3

For the H8S or H8/300 Series C/C++ Compiler, the STM/LDM instruction including ER
created.

2.9.3 Bit Manipulation Instructions

When bit-manipulation is used with registers that include write-only bits, bits to be man
may not be manipulated properly or bits unrelated to the bit-manipulation may be chang

Some values read from write-only bits are fixed and some are undefined. When such bit
operands of bit-manipulation instructions that use read values in arithmetic operations (I
BTST, BAND, BIAND, BOR, BIOR, BXOR, BIXOR, BLD, BILD), the desired bit-ma
will not be executed.

Also, bit-manipulation instructions that write back data according to the results of arithm
operations (BSET, BCLR, BNOT, BST, BIST) may change bits that are not related to th
manipulation. Therefore, special care is necessary when using these instructions with reg
include write-only bits.

In the following example, the BCLR instruction specifies P14 as an input. Before the operation, P17 to P14 are set as output pins and P13 to P10 are set as input pins. The P1DDR is H'F0.

	P17	P16	P15	P14	P13	P12	P11
I/O	Output	Output	Output	Output	Input	Input	Input
P1DDR	1	1	1	1	0	0	0

To switch P14 from an output to an input, the value of bit 4 in P1DDR has to be chan 1 to 0 (from H'F0 to H'E0). The BCLR instruction used to clear bit 4 in P1DDR is as

```
BCLR      #4, @P1DDR
```

However, the above bit-manipulation of the write-only P1DDR register may cause the following problem.

The data in P1DDR is read in bytes. Data read from P1DDR is undefined. Thus, regar whether the value in the register is 0 or 1, it is impossible to tell which value will be r bits in P1DDR are write-only, thus read as undefined. The actual value in P1DDR is us assume that the value read is H'F8, where the value of bit 3 is read as 1 rather than value of 0.

	P17	P16	P15	P14	P13	P12	P11
I/O	Output	Output	Output	Output	Input	Input	Input
P1DDR	1	1	1	1	0	0	0
Read value	1	1	1	1	1	0	0

BCLR instruction is complete.

	P17	P16	P15	P14	P13	P12	P11
I/O	Output	Output	Output	Input	Output	Input	Input
P1DDR	1	1	1	0	1	0	0
Write value	1	1	1	0	1	0	0

This instruction was meant to change the value of P1DDR to H'E0, but H'E8 was wr instead. P13, which should be an input pin, has been turned into an output pin. Note the error in this case occurred because bit 3 in P1DDR was read as 1, the values read 7 to 0 in P1DDR are undefined. Bit-manipulation instructions that write back values change any bit from 0 to 1 or 1 to 0. Section 2.9.4, Access Method for Registers with Only Bits, describes a way to avoid this possibility when changing the values of regi include write-only bits.

The BCLR instruction can be used to clear flags in the internal I/O registers to 0. In t if it is obvious that a given flag has been set to 1 because an interrupt handler has be there is no need to read the flag .

2.9.4 Access Method for Registers with Write-Only Bits

A read value from a write-only bit using a data-transfer or a bit-manipulation instruction undefined. To avoid using the read value for subsequent operations, follow the procedur below to access registers that include write-only bits.

When writing to registers that include write-only bits, set up a work area in memory suc chip RAM, write the data to the work area, read the data back from the memory, and the the data to the registers that include write-only bits.

- Consider the following example, where only bit 4 in P1DDR of port 1 is cleared.

  P1DDR is an 8-bit register that consists of write-only bits and specifies input or outpu each pin of port 1. Reading of these bits is not valid, since values read are specified as undefined.

  In the following example, the BCLR instruction specifies P14 as an input. Start by wr initial value H'F0, which will be written to P1DDR, to the work area (RAM0) in mem

```
MOV.B     #H'F0, R0L
MOV.B     R0L, @RAM0
MOV.B     R0L, @P1DDR
```

	P17	P16	P15	P14	P13	P12	P11
I/O	Output	Output	Output	Output	Input	Input	Input
P1DDR	1	1	1	1	0	0	0
RAM0	1	1	1	1	0	0	0

  P14 is now an output. To switch P14 from an output to an input, the value of bit 4 in I has to be changed from 1 to 0 (from H'F0 to H'E0). Clear bit 4 of RAM0 using the BC instruction.

```
BCLR              #4,    @RAM0
```

```
MOV.B     @RAM0,          R0L
MOV.B     R0L,            @P1DDR
```

	P17	P16	P15	P14	P13	P12	P11
I/O	Output	Output	Output	Input	Input	Input	Input
P1DDR	1	1	1	0	0	0	0

RAM0	1	1	1	0	0	0	0

Following this procedure in access to registers that include write-only bits makes the
of the program independent of the type of instruction.

8-bit bus mode.

In mode 7, the external address space cannot be used. Mode pins should not be changed operations.

Table 3.1 MCU Operating Mode Selection

MCU Operating Mode	MD2	MD1	MD0	CPU Operating Mode	Description	On-Chip ROM	External Initial Width
6	1	1	0	Advanced mode	On-chip ROM valid expansion mode	Enabled	8 bits
7	1	1	1	Advanced mode	Single-chip mode	Enabled	—

Bit	Bit Name	Initial Value	R/W	Descriptions
7	—	1	—	Reserved This bit is always read as 1 and cannot be modifi
6 to 3	—	All 0	—	Reserved These bits are always read as 0 and cannot be m
2	MDS2	—*	R	Mode Select 2 to 0 These bits indicate the input levels at mode pins MD0 (the current operating mode). Bits MDS2 to correspond to pins MD2 to MD0, respectively. MD MDS0 are read-only bits and cannot be modified. input levels at mode pins MD2 to MD0 are latche these bits when MDCR is read. These latches are canceled by a power-on reset, but retained by a r reset.
1	MDS1	—*	R	
0	MDS0	—*	R	

Note: * Determined by the setting of pins MD2 to MD0.

3.2.2 System Control Register (SYSCR)

SYSCR performs the selection of interrupt control mode, the selection of NMI detection
selection of enable/disable of $\overline{\text{MRES}}$ pin input, and the selection of valid/invalid of on-ch

				01: Setting prohibited 10: Interrupt control mode 2 11: Setting prohibited
3	NMIEG	0	R/W	NMI Edge Select Performs input edge selection of the NMI pin. 0: Interrupt request is generated at the falling ed input. 1: Interrupt request is generated at the rising ed input.
2	MRESE	0	R/W	Manual Reset Selection Bit Selects enable/disable of the $\overline{\text{MRES}}$ pin input. 0: Disables manual reset. 1: Enables manual reset. The $\overline{\text{MRES}}$ pin input is
1	—	0	—	Reserved This bit is always read as 0 and cannot be modi
0	RAME	1	R/W	RAM Enable Selects valid/invalid of the on-chip RAM. The RA initialized when a reset is canceled. 0: The on-chip RAM is disabled. 1: The on-chip RAM is enabled.

The address (A7 to A0) is output when the corresponding DDR is set to 1 at port C.

Ports D and E are data buses, and a part of the port F is the bus control signal.

Immediately after a reset, 8-bits bus mode is set and all the areas become 8-bit access spa
However, when any of the areas is set to 16-bit access space by the bus controller, 16-bit
mode is set and port E becomes the data bus.

3.3.2 Mode 7

The CPU can access a 16-Mbyte address space in advanced mode. The on-chip ROM is v
the external address space cannot be accessed.

All the I/O port can be used as an input/output port.

3.3.3 Pin Functions

Table 3.2 shows the pin functions in modes 6 and 7.

PF3	P*/C
PF2 to PF0	P*/C

Legend:

P: Input/output port

A: Address bus output

D: Data bus Input/output

C: Control signal, clock Input/output

*: Immediately after a reset

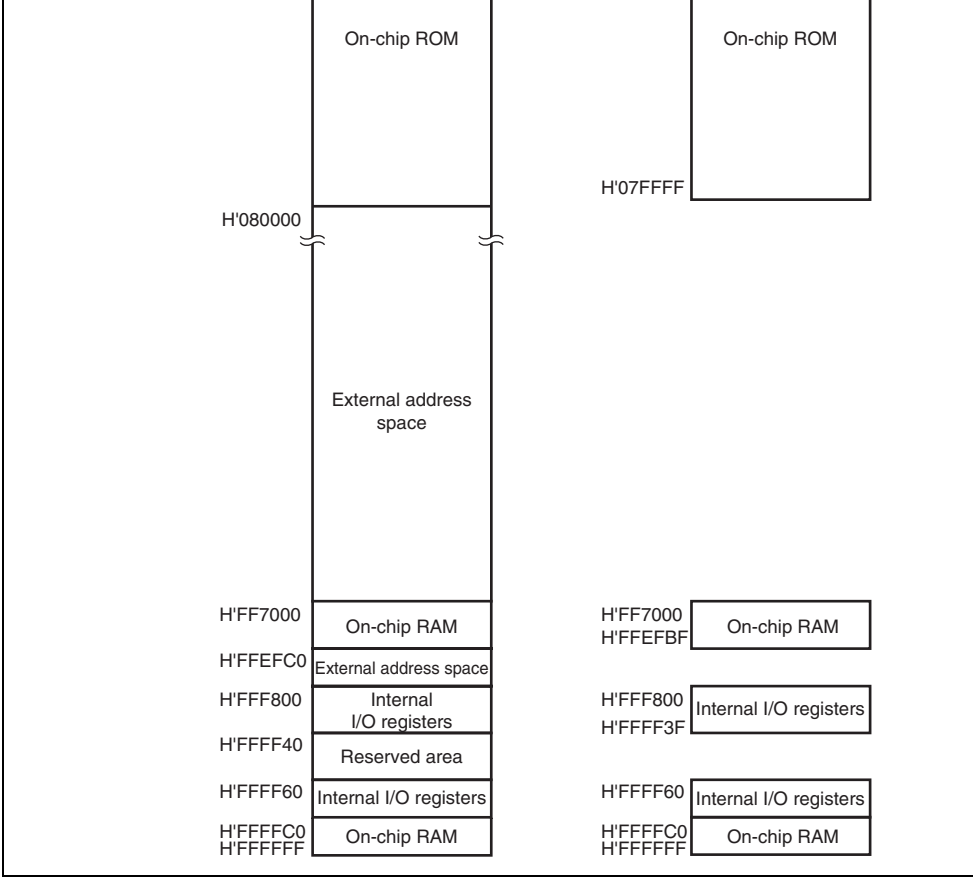


Figure 3.1 Address Map of H8S/2556, H8S/2552, and H8S/2506

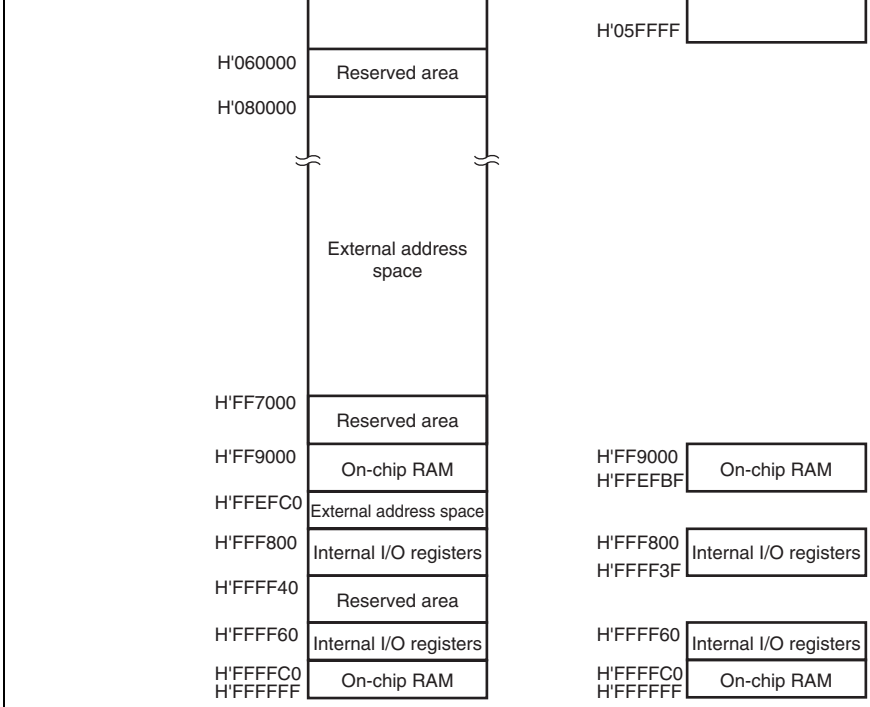


Figure 3.2 Address Map of H8S/2551

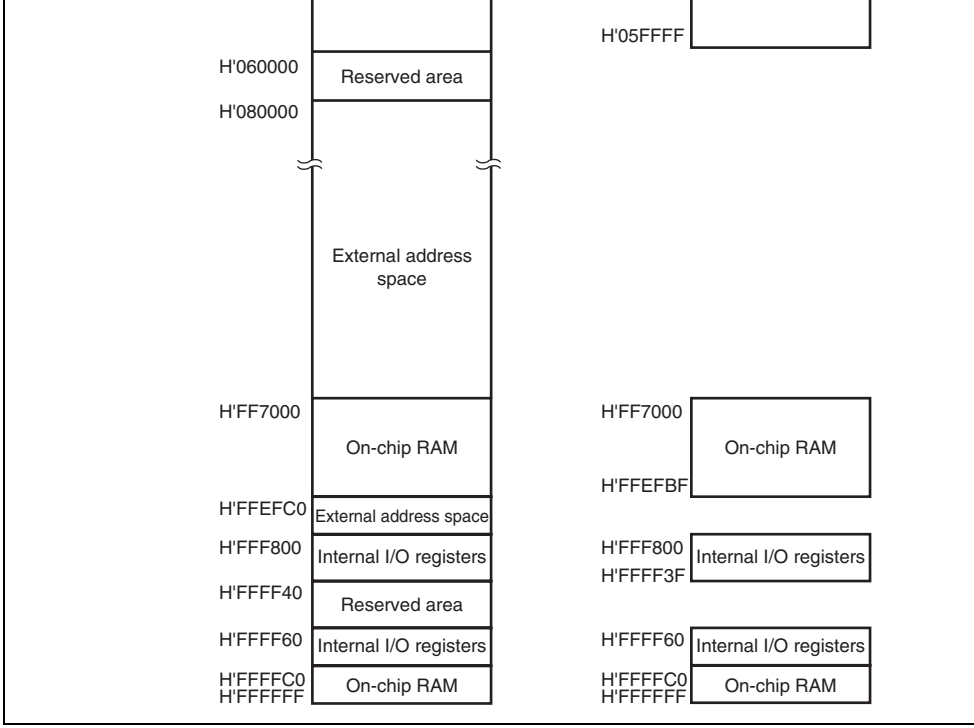


Figure 3.3 Address Map of H8S/2505

interrupt control mode set by the INTM0 and INTM1 bits in SYSCR.

Table 4.1 Exception Types and Priority

Priority	Exception Type	Start of Exception Handling
High	Reset	Starts immediately after a low-to-high transition at th and $\overline{\text{MRES}}$ pins, or when the watchdog timer overflo CPU enters the power-on reset state when the $\overline{\text{RES}}$ The CPU enters the manual reset state when the $\overline{\text{MR}}$ low.
	Trace	Starts when execution of the current instruction or ex handling ends, if the trace (T) bit is set to 1. Trace is only in interrupt control mode 2. Trace exception har not executed after execution of an RTE instruction.
	Interrupt	Starts when execution of the current instruction or ex handling ends, if an interrupt request has been issue Interrupt detection is not performed on completion of ORC, XORC, or LDC instruction execution, or on cor of reset exception handling.
Low	Trap instruction (TRAPA)	Started by execution of a trap instruction (TRAPA). T instruction exception handling requests are accepted times in program execution state.

4.2 Exception Sources and Exception Vector Table

Different vector address is assigned to each exception source. Table 4.2 lists the excepti and their vector addresses.

Direct transitions*3		6	H'0018 to H'001B
External interrupt (NMI)		7	H'001C to H'001F
Trap instruction (four sources)		8	H'0020 to H'0023
		9	H'0024 to H'0027
		10	H'0028 to H'002B
		11	H'002C to H'002F
Reserved for system use		12	H'0030 to H'0033
		13	H'0034 to H'0037
		14	H'0038 to H'003B
		15	H'003C to H'003F
External interrupt	IRQ0	16	H'0040 to H'0043
	IRQ1	17	H'0044 to H'0047
	IRQ2	18	H'0048 to H'004B
	IRQ3	19	H'004C to H'004F
	IRQ4	20	H'0050 to H'0053
	IRQ5	21	H'0054 to H'0057
	IRQ6	22	H'0058 to H'005B
	IRQ7	23	H'005C to H'005F
Internal interrupt*2		24 | 127	H'0060 to H'0063 | H'01FC to H'01FF

Notes: 1. Indicates lower 16 bits of the address.

2. For details on the internal interrupt vector table, see section 5.4.3, Interrupt Ex Handling Vector Table.

3. Direct transitions are not supported in this LSI.

4.3.1 Types of Reset

The LSI supports two types of resets: power-on reset and manual reset.

Table 4.3 shows the types of reset. Set to power-on reset when the power is tuned on.

The CPU internal status is initialized both by the power-on reset and the manual reset. B
power-on reset, all registers of the on-chip peripheral modules are initialized; by a manu
registers of the on-chip peripheral modules, except for the bus controller and I/O ports, a
initialized. The status of the bus controller and I/O ports is maintained.

By a manual reset, on-chip peripheral modules are initialized and thus ports used as inpu
pins of the on-chip peripheral modules are switched to input/output ports controlled by I
DR.

Table 4.3 Types of Reset

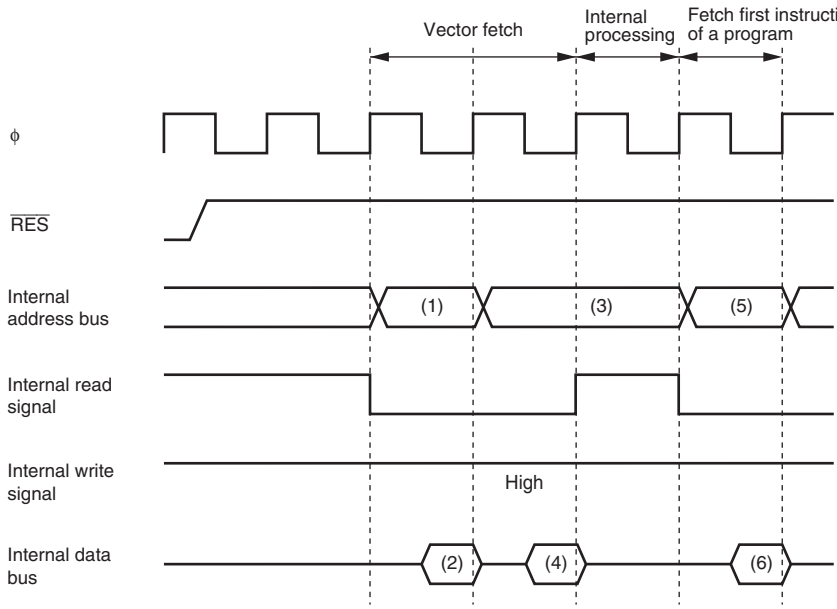
	Reset Shift Conditions		Internal State	
Types	**$\overline{\text{MRES}}$**	**$\overline{\text{RES}}$**	**CPU**	**On-Chip Peripheral Modu**
Power-on reset	*	Low	Initialized	Initialized
Manual reset	Low	High	Initialized	Initialized except for bus c and I/O ports

Legend:
*: Don't care

The power-on reset and the manual reset are also available for the reset by the watchdog
To enable the $\overline{\text{MRES}}$ pin, set the MRESE bit in SYSCR to 1.

2. The reset exception handling vector address is read and transferred to the PC, and pro
execution starts from the address indicated by the PC.

Figures 4.1 shows an example of the reset sequence.



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

Figure 4.1 Reset Sequence (Advanced Mode with On-Chip ROM Enabled

H'FF, and all modules except the DTC enter module stop mode. Consequently, on-chip
module registers cannot be read or written to. Register reading and writing is enabled wh
module stop mode is exited.

4.4 Trace Exception Handling

Trace is enabled in interrupt control mode 2. Trace mode is not entered in interrupt cont
0, irrespective of the state of the T bit. For details on the interrupt control mode, see sect
Interrupt Controller.

If the T bit in EXR is set to 1, trace mode is entered. In trace mode, a trace exception ha
occurs on completion of each instruction. After execution of trace exception handling, th
EXR is cleared to 0 and trace mode is canceled. Trace mode is not affected by interrupt
Table 4.4 shows the state of CCR and EXR after execution of trace exception handling.
are accepted even within the trace exception handling routine.

The T bit saved on the stack retains its value of 1, and when control is returned from the
exception handling routine by the RTE instruction, trace mode resumes. Trace exception
is not carried out after execution of the RTE instruction.

4.5 Interrupt Exception Handling

Interrupts are controlled by the interrupt controller. The interrupt controller has two interr
control modes and can assign interrupts other than NMI to eight priority/mask levels to en
multiplexed interrupt control. For details, see section 5, Interrupt Controller.

Interrupt exception handling is conducted as follows:

1. The values in the program counter (PC), condition code register (CCR), and extended
   (EXR) are saved to the stack.
2. The interrupt mask bit is updated and the T bit is cleared to 0.
3. A vector address corresponding to the interrupt source is generated, the start address
   from the vector table to the PC, and program execution begins from that address.

4.6 Trap Instruction Exception Handling

Trap instruction exception handling starts when a TRAPA instruction is executed. Trap in
exception handling can be executed at all times in the program execution state.

Trap instruction exception handling is conducted as follows:

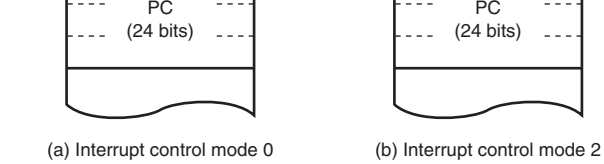
1. The values in the program counter (PC), condition code register (CCR), and extended
   (EXR) are saved to the stack.
2. The interrupt mask bit is updated and the T bit is cleared to 0.
3. A vector address corresponding to the interrupt source is generated, the start address
   from the vector table to the PC, and program execution starts from that address.

Legend:

1: Set to 1

0: Cleared to 0

—: Retains value prior to execution



(a) Interrupt control mode 0 (b) Interrupt control mode 2

Note:* Ignored on return.

Figure 4.2 Stack State after Exception Handling (Advanced Mode)

4.8 Usage Note

When accessing word data or longword data, this LSI assumes that the lowest address bit
stack should always be accessed by word transfer instruction or longword transfer instruc
the value of the stack pointer (SP: ER7) should always be kept even. Use the following
instructions to save registers:

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

Use the following instructions to restore registers:

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

Setting the SP to an odd value may lead to a malfunction. Figure 4.3 shows an example o
happens when the SP value is odd.

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

Note: This diagram illustrates an example in interrupt control mode 0 and advanced mode.

Figure 4.3 Operation when SP Value Is Odd

- Priorities settable with IPR
  - An interrupt priority register (IPR) is provided for setting interrupt priorities. Eig levels can be set for each module for all interrupts except NMI. NMI is assigned highest priority level of 8, and can be accepted at all times.
- Independent vector addresses
  - All interrupt sources are assigned independent vector addresses, making it unnec the source to be identified in the interrupt handling routine.
- Nine external interrupt pins
  - NMI is the highest-priority interrupt, and is accepted at all times. Rising edge or edge can be selected for NMI.
  - Falling edge, rising edge, both edges, or level sensing can be independently selec $\overline{\text{IRQ7}}$ to $\overline{\text{IRQ0}}$.
- DTC control
  - The DTC can be activated by an interrupt request.

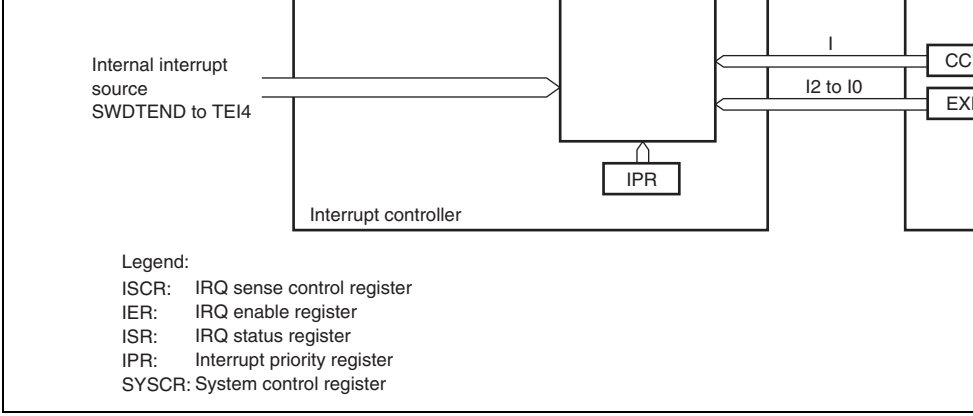


Figure 5.1 Block Diagram of Interrupt Controller

$\overline{\text{IRQ6}}$	Input	Rising edge, falling edge, both edges, or level sensing can be
$\overline{\text{IRQ5}}$	Input	
$\overline{\text{IRQ4}}$	Input	
$\overline{\text{IRQ3}}$	Input	
$\overline{\text{IRQ2}}$	Input	
$\overline{\text{IRQ1}}$	Input	
$\overline{\text{IRQ0}}$	Input	

- Interrupt priority register A (IPRA)
- Interrupt priority register B (IPRB)
- Interrupt priority register C (IPRC)
- Interrupt priority register D (IPRD)
- Interrupt priority register E (IPRE)
- Interrupt priority register F (IPRF)
- Interrupt priority register G (IPRG)
- Interrupt priority register H (IPRH)
- Interrupt priority register I (IPRI)
- Interrupt priority register J (IPRJ)
- Interrupt priority register K (IPRK)
- Interrupt priority register L (IPRL)
- Interrupt priority register M (IPRM)
- Interrupt priority register O (IPRO)

				This bit is always read as 0 and cannot be mod
6	IPR6	1	R/W	These bits set the priority of the corresponding source. 000: Priority level 0 (lowest) 001: Priority level 1 010: Priority level 2 011: Priority level 3 100: Priority level 4 101: Priority level 5 110: Priority level 6 111: Priority level 7 (highest)
5	IPR5	1	R/W	
4	IPR4	1	R/W	
3	—	0	—	Reserved This bit is always read as 0 and cannot be mod
2	IPR2	1	R/W	These bits set the priority of the corresponding source. 000: Priority level 0 (lowest) 001: Priority level 1 010: Priority level 2 011: Priority level 3 100: Priority level 4 101: Priority level 5 110: Priority level 6 111: Priority level 7 (highest)
1	IPR1	1	R/W	
0	IPR0	1	R/W	

5	IRQ5E	0	R/W	IRQ5 Enable The IRQ5 interrupt request is enabled when this
4	IRQ4E	0	R/W	IRQ4 Enable The IRQ4 interrupt request is enabled when this
3	IRQ3E	0	R/W	IRQ3 Enable The IRQ3 interrupt request is enabled when this
2	IRQ2E	0	R/W	IRQ2 Enable The IRQ2 interrupt request is enabled when this
1	IRQ1E	0	R/W	IRQ1 Enable The IRQ1 interrupt request is enabled when this
0	IRQ0E	0	R/W	IRQ0 Enable The IRQ0 interrupt request is enabled when this

				01: Interrupt request is generated at falling edge of IF
				10: Interrupt request is generated at rising edge of IR
				11: Interrupt request is generated at both falling and r edges of $\overline{\text{IRQ7}}$ input
13	IRQ6SCB	0	R/W	IRQ6 Sense Control B
12	IRQ6SCA	0	R/W	IRQ6 Sense Control A
				00: Interrupt request is generated at $\overline{\text{IRQ6}}$ input level
				01: Interrupt request is generated at falling edge of IF
				10: Interrupt request is generated at rising edge of IR
				11: Interrupt request is generated at both falling and r edges of $\overline{\text{IRQ6}}$ input
11	IRQ5SCB	0	R/W	IRQ5 Sense Control B
10	IRQ5SCA	0	R/W	IRQ5 Sense Control A
				00: Interrupt request is generated at $\overline{\text{IRQ5}}$ input level
				01: Interrupt request is generated at falling edge of IF
				10: Interrupt request is generated at rising edge of IR
				11: Interrupt request is generated at both falling and r edges of $\overline{\text{IRQ5}}$ input
9	IRQ4SCB	0	R/W	IRQ4 Sense Control B
8	IRQ4SCA	0	R/W	IRQ4 Sense Control A
				00: Interrupt request is generated at $\overline{\text{IRQ4}}$ input level
				01: Interrupt request is generated at falling edge of IF
				10: Interrupt request is generated at rising edge of IR
				11: Interrupt request is generated at both falling and r edges of $\overline{\text{IRQ4}}$ input

				edges of IRQ3 input
5	IRQ2SCB	0	R/W	IRQ2 Sense Control B
4	IRQ2SCA	0	R/W	IRQ2 Sense Control A 00: Interrupt request is generated at $\overline{\text{IRQ2}}$ input le 01: Interrupt request is generated at falling edge input 10: Interrupt request is generated at rising edge c input 11: Interrupt request is generated at both falling a edges of $\overline{\text{IRQ2}}$ input
3	IRQ1SCB	0	R/W	IRQ1 Sense Control B
2	IRQ1SCA	0	R/W	IRQ1 Sense Control A 00: Interrupt request is generated at $\overline{\text{IRQ1}}$ input le 01: Interrupt request is generated at falling edge input 10: Interrupt request is generated at rising edge c input 11: Interrupt request is generated at both falling a edges of $\overline{\text{IRQ1}}$ input

edges of IRQ0 input

5.3.4 IRQ Status Register (ISR)

ISR indicates the status of IRQ7 to IRQ 0 interrupt requests.

Bit	Bit Name	Initial Value	R/W*	Description
7	IRQ7F	0	R/W	IRQ7 to IRQ0 flags
6	IRQ6F	0	R/W	These bits indicate the status of IRQ7 to IRQ0 in requests.
5	IRQ5F	0	R/W	
4	IRQ4F	0	R/W	[Setting condition]
3	IRQ3F	0	R/W	• When the interrupt source selected by the I registers occurs
2	IRQ2F	0	R/W	[Clearing conditions]
1	IRQ1F	0	R/W	• Cleared by reading IRQnF flag when IRQnF writing 0 to IRQnF flag
0	IRQ0F	0	R/W	• When interrupt exception handling is execu low-level detection is set and $\overline{\text{IRQn}}$ (n = 0 to is high
				• When IRQn interrupt exception handling is while detection of falling edge, rising edge, edges is set
				• When the DTC is activated by an IRQn inte the DISEL bit in MRB of the DTC with the tr counter other than 0 is cleared to 0

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

IRQ7 to IRQ0 Interrupts: IRQ7 to IRQ0 interrupts are requested by an input signal at t
to $\overline{\text{IRQ0}}$ pins. IRQ7 to IRQ0 interrupts have the following features:

- Using ISCR, it is possible to select whether an interrupt is generated by a low level, fa
  edge, rising edge, or both edges, at the $\overline{\text{IRQ7}}$ to $\overline{\text{IRQ0}}$ pins.
- Enabling or disabling of IRQ7 to IRQ0 interrupt requests can be selected with IER.
- The interrupt priority level can be set with IPR.
- The status of IRQ7 to IRQ0 interrupt requests is indicated in ISR. ISR flags can be cle
  by software.

A block diagram of IRQ7 to IRQ0 interrupts is shown in figure 5.2.

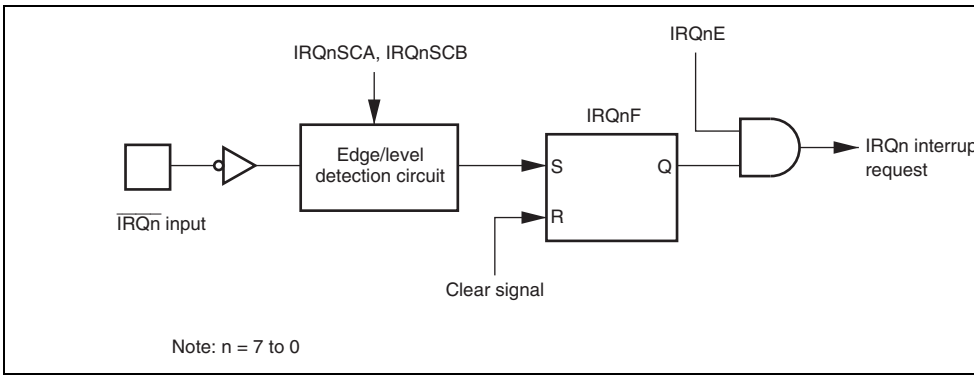


Figure 5.2 Block Diagram of IRQ7 to IRQ0 Interrupts

The set timing for IRQ7F to IRQ0F is shown in figure 5.3.

The detection of IRQ7 to IRQ0 interrupts does not depend on whether the relevant pin h
set for input or output. However, when a pin is used as an external interrupt input pin, d
the corresponding DDR to 0 to use the pin as an I/O pin for another function. The IRQ7
IRQ0F interrupt request flags can be set to 1 when the setting condition is satisfied, rega
IER settings. Accordingly, refer to only necessary flags.

5.4.2 Internal Interrupts

For each on-chip peripheral module, there are flags that indicate the interrupt request sta
enable bits that select enabling or disabling of these interrupts. If both of these are set to
particular interrupt source, an interrupt request is sent to the interrupt controller.

5.4.3 Interrupt Exception Handling Vector Table

Table 5.2 shows interrupt exception handling sources, vector addresses, and interrupt pr
For default priorities, the lower the vector number, the higher the priority.

Priorities among modules can be set by means of the IPR. Modules set at the same prior
conform to their default priorities. Priorities within a module are fixed.

	IRQ4	20	H'0050	IPRB2 to IPRB0
	IRQ5	21	H'0054	
	IRQ6	22	H'0058	IPRC6 to IPRC4
	IRQ7	23	H'005C	
DTC	SWDTEND (completion of software initiation data transfer)	24	H'0060	IPRC2 to IPRC0
Watchdog timer 0	WOVI0 (interval timer 0)	25	H'0064	IPRD6 to IPRD4
PC break	PC break	27	H'006C	IPRE6 to IPRE4
A/D	ADI (completion of A/D conversion)	28	H'0070	IPRE2 to IPRE0
Watchdog timer 1	WOVI1 (interval timer 1)	29	H'0074	
—	Reserved	30 31	H'0078 H'007C	
TPU channel 0	TGI0A (TGR0A input capture/compare-match)	32	H'0080	IPRF6 to IPRF4
	TGI0B (TGR0B input capture/compare-match)	33	H'0084	
	TGI0C (TGR0C input capture/compare-match)	34	H'0088	

TPU channel 1	TGI1A (TGR1A input capture/compare-match)	40	H'00A0	IPRF2 to IPRF0
	TGI1B (TGR1B input capture/compare-match)	41	H'00A4	
	TCI1V (overflow 1)	42	H'00A8	
	TCI1U (underflow 1)	43	H'00AC	
TPU channel 2	TGI2A (TGR2A input capture/compare-match)	44	H'00B0	IPRG6 to IPRG4
	TGI2B (TGR2B input capture/compare-match)	45	H'00B4	
	TCI2V (overflow 2)	46	H'00B8	
	TCI2U (underflow 2)	47	H'00BC	
TPU channel 3	TGI3A (TGR3A input capture/compare-match)	48	H'00C0	IPRG2 to IPRG0
	TGI3B (TGR3B input capture/compare-match)	49	H'00C4	
	TGI3C (TGR3C input capture/compare-match)	50	H'00C8	
	TGI3D (TGR3D input capture/compare-match)	51	H'00CC	
	TCI3V (overflow 3)	52	H'00D0	
—	Reserved	53 54 55	H'00D4 H'00D8 H'00DC	

5	capture/compare-match)			
	TGI5B (TGR5B input capture/compare-match)	61	H'00F4	
	TCI5V (overflow 5)	62	H'00F8	
	TCI5U (underflow 5)	63	H'00FC	
8-bit timer channel 0	CMIA0 (compare-match A0)	64	H'0100	IPRI6 to IPRI4
	CMIB0 (compare-match B0)	65	H'0104	
	OVI0 (overflow 0)	66	H'0108	
—	Reserved	67	H'010C	
8-bit timer channel 1	CMIA1 (compare-match A1)	68	H'0110	IPRI2 to IPRI0
	CMIB1 (compare-match B1)	69	H'0114	
	OVI1 (overflow 1)	70	H'0118	
—	Reserved	71	H'011C	
SCI channel 0	ERI0 (receive error 0)	80	H'0140	IPRJ2 to IPRJ0
	RXI0 (receive completion 0)	81	H'0144	
	TXI0 (transmit data empty 0)	82	H'0148	
	TEI0 (transmit end 0)	83	H'014C	
SCI channel 1	ERI1 (receive error 1)	84	H'0150	IPRK6 to IPRK4
	RXI1 (receive completion 1)	85	H'0154	
	TXI1 (transmit data empty 1)	86	H'0158	
	TEI1 (transmit end 1)	87	H'015C	
SCI channel 2	ERI2 (receive error 2)	88	H'0160	IPRK2 to IPRK0
	RXI2 (receive completion 2)	89	H'0164	

8-bit timer channel 3	CMIA3 (compare-match A3)	96	H'0180	
	CMIB3 (compare-match B3)	97	H'0184	
	OVI3 (overflow 3)	98	H'0188	
—	Reserved	99	H'018C	
IEB (H8S/2552 Group only)	IERSI (reception status)	104	H'01A0	IPRM6 to IPRM4
	IERxI (RxRDY)	105	H'01A4	
	IETxI (TxRDY)	106	H'01A8	
	IETSI (transmission status)	107	H'01AC	
HCAN (H8S/2556 Group only)	ERS0, OVR0, RM1, SLE0	108	H'01B0	IPRM2 to IPRM0
	RM0	109	H'01B4	
IIC2 channel 0	IICI0 (1-byte transmission/ reception completion)	110	H'01B8	
IIC2 channel 1	IICI1 (1-byte transmission/ reception completion)	111	H'01BC	
SCI channel 3	ERI3 (receive error 3)	120	H'01E0	IPRO6 to IPRO4
	RXI3 (receive completion 3)	121	H'01E4	
	TXI3 (transmit data empty 3)	122	H'01E8	
	TEI3 (transmit end 3)	123	H'01EC	
SCI channel 4	ERI4 (receive error 4)	124	H'01F0	IPRO2 to IPRO0
	RXI4 (receive completion 4)	125	H'01F4	
	TXI4 (transmit data empty 4)	126	H'01F8	
	TEI4 (transmit end 4)	127	H'01FC	

Note: * Indicates lower 16 bits of the start address.

Table 5.3 shows the interrupt control modes.

The interrupt controller performs interrupt control according to the interrupt control mode the INTM1 and INTM0 bits in SYSCR, the priorities set in IPR, and the masking state in by the I bit in the CPU's CCR, and bits I2 to I0 in EXR.

Table 5.3 Interrupt Control Modes

Interrupt Control Mode	SYSCR		Priority Setting Register	Interrupt Mask Bits	Description
	INTM1	INTM0			
0	0	0	—	I	Interrupt mask control is performed by the I bit.
—		1	—	—	Setting prohibited
2	1	0	IPR	I2 to I0	8-level interrupt mask co performed by bits I2 to I0 8 priority levels can be se IPR.
—		1	—	—	Setting prohibited

Figures 5.4 shows a block diagram of the priority decision circuit.

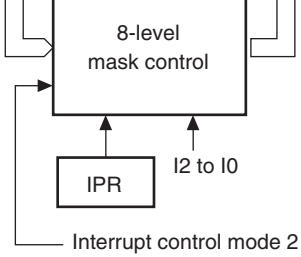


Figure 5.4 Block Diagram of Interrupt Control Operation

2	X	All interrupts

Legend:
X: Don't care

8-Level Control: In interrupt control mode 2, 8-level mask level determination is perfor
the selected interrupts in interrupt acceptance control according to the interrupt priority le
(IPR).

The interrupt source selected is the interrupt with the highest priority level, and whose pri
level set in IPR is higher than the mask level.

Table 5.5 Interrupts Selected in Each Interrupt Control Mode (2)

Interrupt Control Mode	Selected Interrupts
0	All interrupts
2	Highest priority-level (IPR) interrupt whose priority level is g than the mask level (IPR > I2 to I0)

Default Priority Determination: When an interrupt is selected by 8-level control, its pri
determined and a vector number is generated.

If the same value is set for IPR, acceptance of multiple interrupts is enabled, and so only
interrupt source with the highest priority according to the preset default priorities is selec
has a vector number generated. Interrupt sources with a lower priority than the accepted i
source are held pending.

Table 5.6 shows operations and control signal functions in each interrupt control mode.

X: No operation. (All interrupts enabled)
IM: Used as interrupt mask bit
PR: Sets priority
—: Not used
Notes: 1. Set to 1 when an interrupt is accepted.
2. Keep the initial setting.

5.5.2 Interrupt Control Mode 0

Enabling and disabling of IRQ interrupts and on-chip peripheral module interrupts can b
means of the I bit in the CPU's CCR. Interrupts are enabled when the I bit is cleared to 0
disabled when set to 1.

Figure 5.5 shows a flowchart of the interrupt acceptance operation in this case.

1. If an interrupt source occurs when the corresponding interrupt enable bit is set to 1, a
interrupt request is sent to the interrupt controller.
2. The I bit is referred to. If the I bit is cleared to 0, an interrupt request is accepted. If t
set to 1, only an NMI interrupt is accepted, and other interrupt requests are held pend
3. Interrupt requests are sent to the interrupt controller, the highest-ranked interrupt acc
the priority system is accepted, and other interrupt requests are held pending.
4. When the CPU accepts an interrupt request, it starts interrupt exception handling afte
execution of the current instruction has been completed.
5. The PC and CCR are saved to the stack area by interrupt exception handling. The PC
the stack shows the address of the first instruction to be executed after returning from
interrupt handling routine.
6. Next, the I bit in CCR is set to 1. This masks all interrupts except NMI.

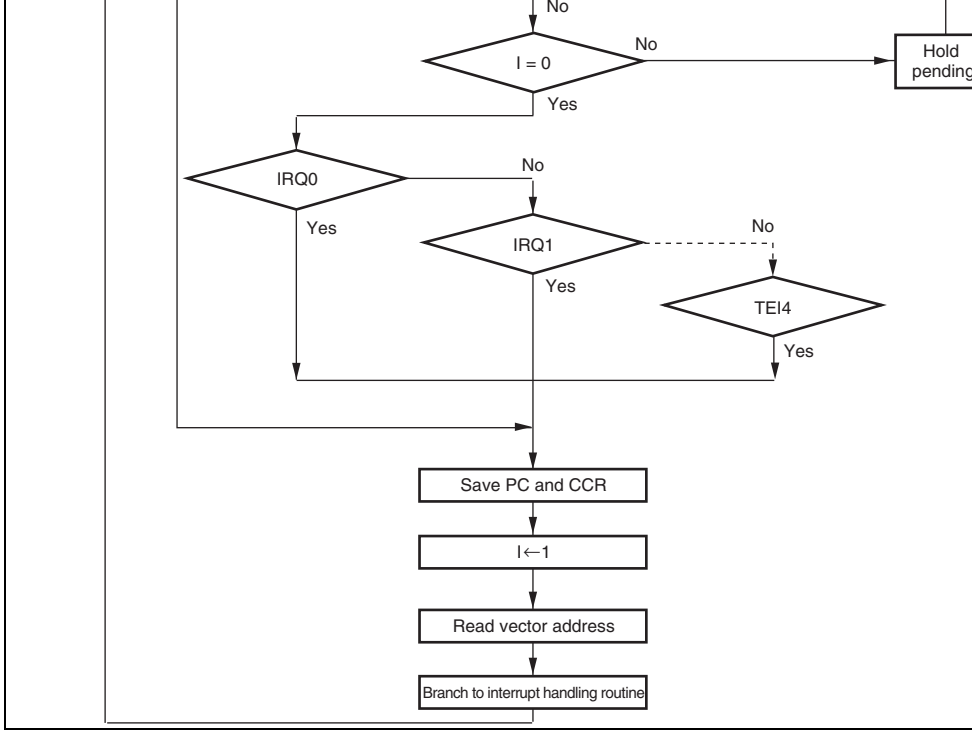


Figure 5.5 Flowchart of Procedure up to Interrupt Acceptance in Interrupt Control Mode 0

interrupt requests are held pending. If a number of interrupt requests with the same p
generated at the same time, the interrupt request with the highest priority according t
priority system shown in table 5.2 is selected.

3. Next, the priority of the selected interrupt request is compared with the interrupt mas
in EXR. An interrupt request with a priority no higher than the mask level set at that
held pending, and only an interrupt request with a priority higher than the interrupt m
is accepted.
4. When the CPU accepts an interrupt request, it starts interrupt exception handling afte
execution of the current instruction has been completed.
5. The PC, CCR, and EXR are saved to the stack area by interrupt exception handling.
saved on the stack shows the address of the first instruction to be executed after retu
the interrupt handling routine.
6. The T bit in EXR is cleared to 0. The interrupt mask level is rewritten with the priori
the accepted interrupt.
If the accepted interrupt is NMI, the interrupt mask level is set to H'7.
7. The CPU generates a vector address for the accepted interrupt and starts execution o
interrupt handling routine at the address indicated by the contents of the vector addre

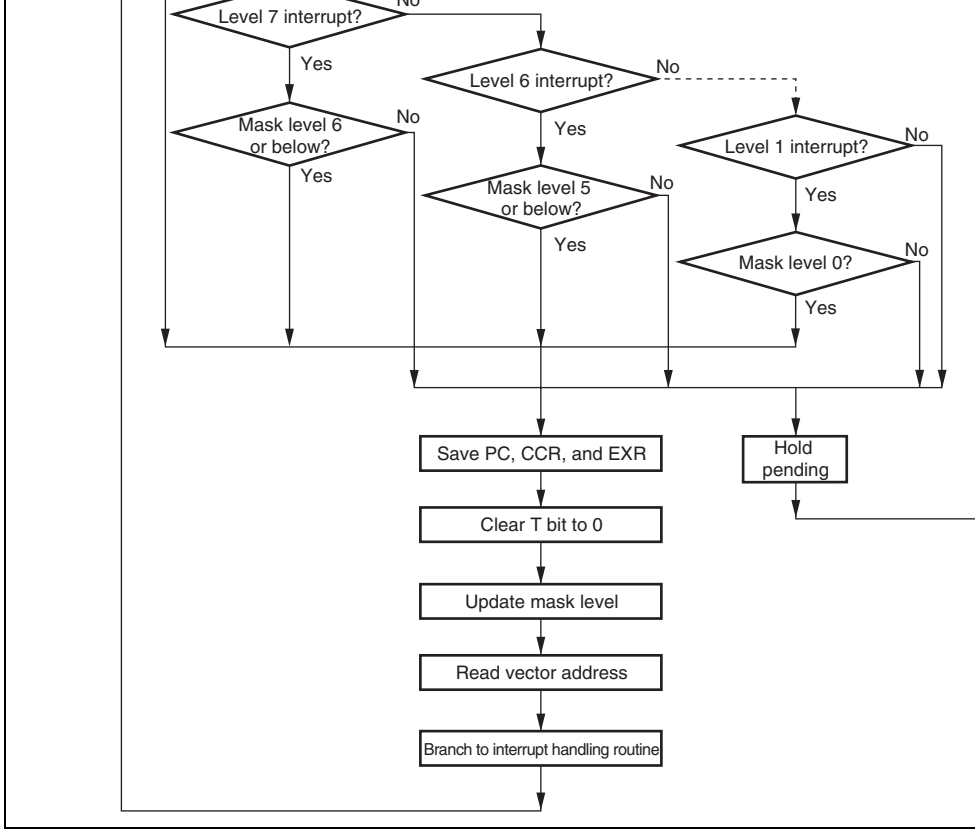


Figure 5.6 Flowchart of Procedure up to Interrupt Acceptance in Interrupt Control Mode 2

Figure 5.7 Interrupt Exception Handling

No.	Execution Status	INTM1 = 0	INTM1 = 1	INTM1 = 0	IN
1	Interrupt priority determination*1	3	3	3	3
2	Number of wait states until executing instruction ends*2	1 to 19 + $2 \cdot S_I$	1 to 19 + $2 \cdot S_I$	1 to 19 + $2 \cdot S_I$	1 $2 \cdot$
3	PC, CCR, EXR stack save	$2 \cdot S_K$	$3 \cdot S_K$	$2 \cdot S_K$	$3 \cdot$
4	Vector fetch	S_I	S_I	$2 \cdot S_I$	$2 \cdot$
5	Instruction fetch*3	$2 \cdot S_I$	$2 \cdot S_I$	$2 \cdot S_I$	$2 \cdot$
6	Internal processing*4	2	2	2	2
Total (using on-chip memory)		11 to 31	12 to 32	12 to 32	13

Notes: 1. Two states in case of internal interrupt.
2. Refers to MULXS and DIVXS instructions.
3. Prefetch after interrupt acceptance and interrupt handling routine prefetch.
4. Internal processing after interrupt acceptance and internal processing after ve
5. Not available in this LSI.

Legend:

m: Number of wait states in an external device access.

Note: * Not available in this LSI.

5.5.6 DTC Activation by Interrupt

The DTC can be activated by an interrupt. In this case, the following selections can be ma

1. Interrupt request to CPU
2. Activation request to DTC
3. Multiple selection of 1 and 2 above.

For details on interrupt request, which enables DTC activation, see section 8, Data Transf Controller (DTC). Figure 5.8 shows a block diagram of DTC and interrupt controller.

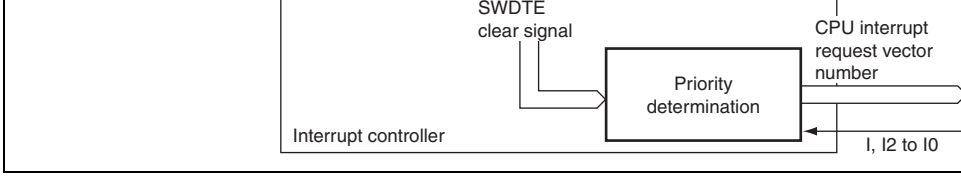


Figure 5.8 DTC and Interrupt Controller

Interrupt controller of DTC control has the following three main functions.

Interrupt Source Selection: For interruption source, select DTC activation request or C interruption request by the DTCE bits in DTCERA to DTCERG, and DTCERI of the DT DTC data transfer, the DTCE bit is cleared to 0, and an interrupt request to the CPU can by the setting of the DISEL bit in MRB of the DTC. When DTC performs data transfer prescribed number of times and transfer counter becomes 0, the DTCE bit should be cle and an interrupt request to the CPU is made after DTC data transfer.

Priority Determination: DTC activation source is selected according to priority of defa setting. Mask level and priority level do not affect the selection. For details, see section Location of Register Information and DTC Vector Table.

DTCE	DESEL	DTC	CPU
0	*	X	#
1	0	#	X
	1	O	#

Legend:

#: Corresponding interrupt is used. Interrupt source is cleared.
(The CPU should clear the source flag in the interrupt processing routine.)

O: Corresponding interrupt is used. Interrupt source is not cleared.

X: Corresponding interrupt cannot be used.

*: Don't care

Usage Note: Interrupt sources of the SCI and A/D converter are cleared when the DTC re writes prescribed register, and they do not depend on the DTCE or DISEL bit.

5.6 Usage Notes

5.6.1 Contention between Interrupt Generation and Disabling

When an interrupt enable bit is cleared to 0 to disable interrupt requests, the disabling bec effective after execution of the instruction.

When an interrupt enable bit is cleared to 0 by an instruction such as BCLR or MOV, and interrupt is generated during execution of the instruction, the interrupt concerned will stil enabled on completion of the instruction, and so interrupt exception handling for that inte be executed on completion of the instruction. However, if there is an interrupt request of priority than that interrupt, interrupt exception handling will be executed for the higher-pr interrupt, and the lower-priority interrupt will be ignored.

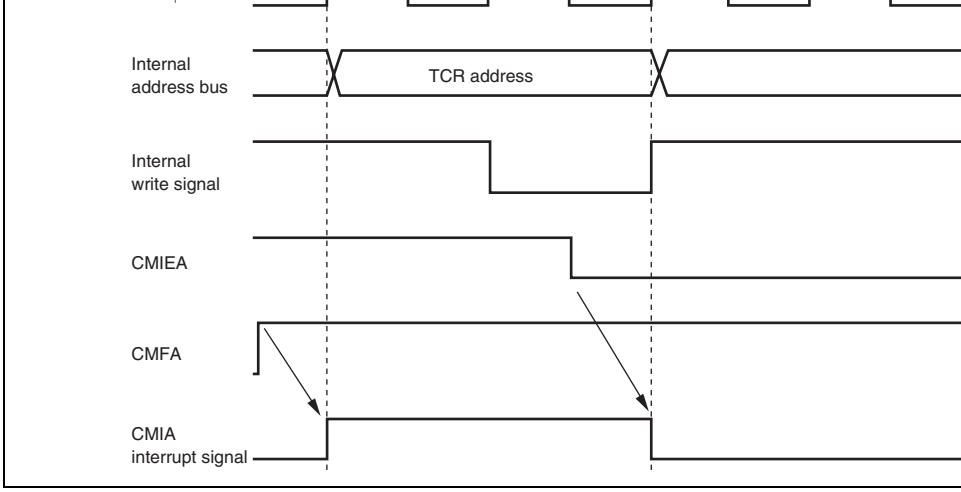


Figure 5.9 Contention between Interrupt Generation and Disabling

5.6.2 Instructions That Disable Interrupts

The instructions that disable interrupt requests directly after execution are LDC, ANDC, and XORC. After any of these instructions are executed, all interrupts including NMI ar and the next instruction is always executed.

When the I bit is set by one of these instructions, the new value becomes valid two state execution of the instruction ends.

5.6.3 When Interrupts Are Disabled

There are times when interrupt acceptance is disabled by the interrupt controller.

exception handling starts at a break in the transfer cycle. The PC value saved on the stack case is the address of the next instruction.

Therefore, if an interrupt is generated during execution of an EEPMOV.W instruction, the following coding should be used.

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

5.6.5 IRQ Interrupt

When operating by clock input, acceptance of input to an IRQ is synchronized with the cl software standby mode and watch mode, the input is accepted asynchronously. For detail input conditions, see section 24.4.3, Control Signal Timing.

5.6.6 NMI Interrupt Usage Notes

The NMI interrupt is part of the exception processing performed cooperatively by the LS internal interrupt controller and the CPU when the system is operating normally under the specified electrical conditions. No operations, including NMI interrupts, are guaranteed w operation is not normal (runaway status) due to software problems or abnormal input to th pins. In such cases, the LSI may be restored to the normal program execution state by app external reset.

- 24-bit break address
  - Bit masking possible
- Four types of break compare conditions
  - Instruction fetch
  - Data read
  - Data write
  - Data read/write
- Bus master
  - Either CPU or CPU/DTC can be selected
- The timing of PC break exception handling after the occurrence of a break condition follows:
  - Immediately before execution of the instruction fetched at the set address (instruc fetch)
  - Immediately after execution of the instruction that accesses data at the set addres access)
- Module stop mode can be set

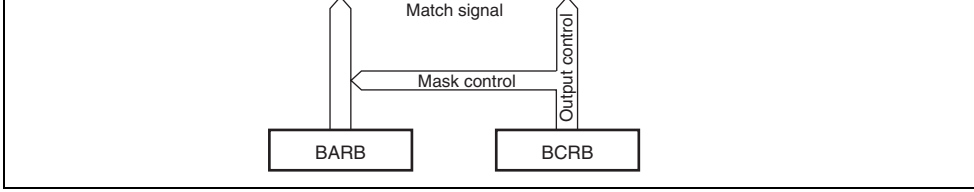


Figure 6.1 Block Diagram of PC Break Controller

6.2 Register Descriptions

The PC break controller has the following registers.

- Break address register A (BARA)
- Break address register B (BARB)
- Break control register A (BCRA)
- Break control register B (BCRB)

6.2.1 Break Address Register A (BARA)

BARA is a 32-bit readable/writable register that specifies the channel A break address.

Bit	Bit Name	Initial Value	R/W	Description
31 to 24	—	Undefined	—	Reserved These bits are read as an undefined valu cannot be modified.
23 to 0	BAA23 to BAA0	H'000000	R/W	These bits set the PC break address of c A.

				[Setting condition] When a condition set for channel A is satisfied [Clearing condition] When 0 is written to CMFA after reading*2 CMF
6	CDA	0	R/W	CPU Cycle/DTC Cycle Select A Selects the channel A break condition bus mast 0: CPU 1: CPU or DTC
5	BAMRA2	0	R/W	Break Address Mask Register A2 to A0
4	BAMRA1	0	R/W	These bits specify which bits of the break addre
3	BAMRA0	0	R/W	BARA are to be unmasked. 000: BAA23 to BAA0 (All bits are unmasked) 001: BAA23 to BAA1 (Lowest bit is masked) 010: BAA23 to BAA2 (Lower 2 bits are masked) 011: BAA23 to BAA3 (Lower 3 bits are masked) 100: BAA23 to BAA4 (Lower 4 bits are masked) 101: BAA23 to BAA8 (Lower 8 bits are masked) 110: BAA23 to BAA12 (Lower 12 bits are maske 111: BAA23 to BAA16 (Lower 16 bits are maske
2	CSELA1	0	R/W	Break Condition Select
1	CSELA0	0	R/W	These bits select the break condition of channel 00: Instruction fetch is used as the break conditi 01: Data read cycle is used as the break conditi 10: Data write cycle is used as the break conditi 11: Data read/write cycle is used as the break c

BCRB is the channel B break control register. The bit configuration is the same as for BC

6.3 Operation

The operation flow from break condition setting to PC break interrupt exception handling shown in section 6.3.1, PC Break Interrupt Due to Instruction Fetch, and section 6.3.2, PC Interrupt Due to Data Access, taking the example of channel A.

6.3.1 PC Break Interrupt Due to Instruction Fetch

1. Set the break address in BARA.

   For a PC break caused by an instruction fetch, set the address of the first instruction b break address.
2. Set the break conditions in BCR.

   Set bit 6 (CDA) to 0 to select the CPU because the bus master must be the CPU for a caused by an instruction fetch. Set the address bits to be masked to bits 5 to 3 (BAMR BAMRA0). Set bits 2 and 1 (CSELA1 and CSELA0) to 00 to specify an instruction fe the break condition. Set bit 0 (BIEA) to 1 to enable break interrupts.
3. When the instruction at the set address is fetched, a PC break request is generated imm before execution of the fetched instruction, and the condition match flag (CMFA) is s
4. After priority determination by the interrupt controller, PC break interrupt exception h is started.

3. After execution of the instruction that performs a data access on the set address, a PC
request is generated and the condition match flag (CMFA) is set.
4. After priority determination by the interrupt controller, PC break interrupt exception
is started.

6.3.3 Notes on PC Break Interrupt Handling

- When a PC break interrupt is generated at the transfer address of an EEPMOV.B ins
PC break exception handling is executed after all data transfers have been completed
EEPMOV.B instruction has ended.
- When a PC break interrupt is generated at a DTC transfer address
PC break exception handling is executed after the DTC has completed the specified
data transfers, or after data for which the DISEL bit is set to 1 has been transferred.

6.3.4 Operation in Transitions to Power-Down Modes

The operation when a PC break interrupt is set for an instruction fetch at the address afte
SLEEP instruction is shown below.

- When the SLEEP instruction causes a transition from high-speed (medium-speed) m
sleep mode
After execution of the SLEEP instruction, a transition is not made to sleep mode, and
exception handling is executed. After execution of PC break exception handling, the
instruction at the address after the SLEEP instruction is executed (figure 6.2 (A)).

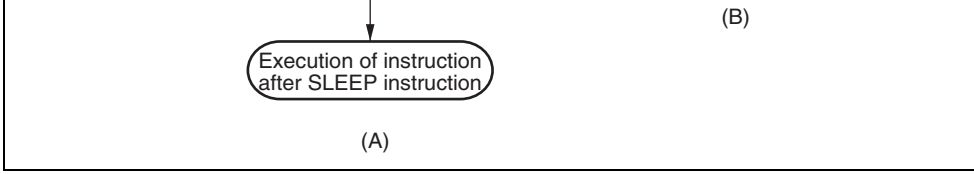


Figure 6.2 Operation in Power-Down Mode Transitions

6.3.5 When Instruction Execution Is Delayed by One State

While the break interrupt enable bit is set to 1, instruction execution in the following case state later than usual.

- For 1-word branch instructions (Bcc d:8, BSR, JSR, JMP, TRAPA, RTE, and RTS) in ROM or RAM
- When break interruption by instruction fetch is set, the set address indicates on-chip R RAM space, and that address is used for data access
- When break interruption by instruction fetch is set, if the instruction to be executed immediately before the set instruction has one of the addressing modes shown below, address indicates on-chip ROM or RAM

  Addressing modes: @ERn, @(d:16,ERn), @(d:32,ERn), @-ERn/ERn+, @aa:8, @aa: @aa:32, @(d:8,PC), @(d:16,PC), @@aa:8
- When break interruption by instruction fetch is set, if the instruction to be executed immediately before the set instruction is NOP or SLEEP, or has #xx, Rn as its address mode, and that instruction is located in on-chip ROM or RAM

The PC break interrupt is shared by channels A and B. The channel from which the requ
issued must be determined by the interrupt handler.

6.4.3 CMFA and CMFB

The CMFA and CMFB flags are not automatically cleared to 0, so 0 must be written to C
CMFB after first reading the flag while it is set to 1. If the flag is left set to 1, another in
will be requested after interrupt handling ends.

6.4.4 PC Break Interrupt when DTC Is Bus Master

A PC break interrupt generated when the DTC is the bus master is accepted after the bus
mastership has been transferred to the CPU by the bus controller.

6.4.5 PC Break Set for Instruction Fetch at Address Following BSR, JSR, JMP RTE, or RTS Instruction

Even if the instruction at the address following a BSR, JSR, JMP, TRAPA, RTE, or RTS
instruction is fetched, it is not executed, and so a PC break interrupt is not generated by
instruction fetch at the next address.

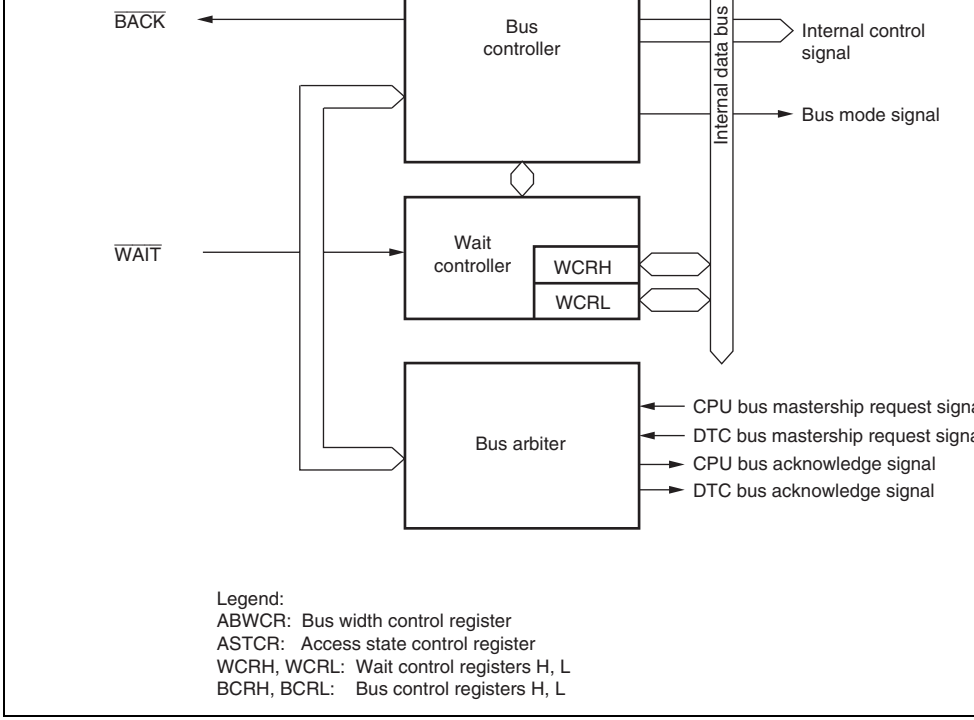
6.4.6 I Bit Set by LDC, ANDC, ORC, or XORC Instruction

When the I bit is set by an LDC, ANDC, ORC, or XORC instruction, a PC break interru
becomes valid two states after the end of the executing instruction. If a PC break interru
for the instruction following one of these instructions, since interrupts, including NMI, a
disabled for a 3-state period in the case of LDC, ANDC, ORC, and XOR, the next instru
always executed. For details, see section 5, Interrupt Controller.

destination is not executed.

— Bus specifications can be set independently for each area
— Burst ROM interface can be set

- Basic bus interface
  — H8S/2552 Group, H8S/2506 Group: Chip select signals ($\overline{CS0}$ to $\overline{CS7}$) can be out areas 0 to 7.
  — H8S/2556 Group: Chip select signals ($\overline{CS0}$, $\overline{CS3}$ to $\overline{CS7}$) can be output for areas 7.
  — 8-bit access or 16-bit access can be selected for each area
  — 2-state access or 3-state access can be selected for each area
  — Program wait states can be inserted for each area
- Burst ROM interface
  — Burst ROM interface can be selected for area 0
  — One or two states can be selected for the burst cycle
- Idle cycle insertion
  — Idle cycle can be inserted between consecutive read accesses to different externa
  — Idle cycle can be inserted before a write access to an external area immediately a access to an external area
- Bus mastership arbitration
  — The on-chip bus arbiter arbitrates the bus mastership among CPU and DTC.
- Other features
  — External bus mastership release function



Legend:
ABWCR: Bus width control register
ASTCR: Access state control register
WCRH, WCRL: Wait control registers H, L
BCRH, BCRL: Bus control registers H, L

Figure 7.1 Block Diagram of Bus Controller

High write	$\overline{\text{HWR}}$	Output	Strobe signal indicating that external addres to be written, and upper half (D15 to D8) of enabled.
Low write	$\overline{\text{LWR}}$	Output	Strobe signal indicating that external addres to be written, and lower half (D7 to D0) of da enabled.
Chip select 0 to 7	$\overline{\text{CS0}}$ to $\overline{\text{CS7}}$*	Output	Strobe signal indicating that areas 0 to 7 are
Wait	$\overline{\text{WAIT}}$	Input	Wait request signal when accessing externa access space.
Bus mastership request	$\overline{\text{BREQ}}$	Input	Request signal that releases bus to external
Bus mastership request acknowledge	$\overline{\text{BACK}}$	Output	Acknowledge signal indicating that bus has released.

Note: * $\overline{\text{CS1}}$ and $\overline{\text{CS2}}$ are not provided in the H8S/2556 Group.

7.3 Register Descriptions

The bus controller has the following registers.

- Bus width control register (ABWCR)
- Access state control register (ASTCR)
- Wait control register H (WCRH)
- Wait control register L (WCRL)
- Bus control register H (BCRH)
- Bus control register L (BCRL)
- Pin function control register (PFCR)

5	ABW5	1	R/W	designated for 8-bit access or 16-bit access.
4	ABW4	1	R/W	0: Area n is designated for 16-bit access
3	ABW3	1	R/W	1: Area n is designated for 8-bit access
2	ABW2	1	R/W	Note: n = 7 to 0
1	ABW1	1	R/W	
0	ABW0	1	R/W	

7.3.2 Access State Control Register (ASTCR)

ASTCR designates each area as either a 2-state access space or a 3-state access space.

ASTCR sets the number of access states for the external memory space. The number of a
states for on-chip memory and internal I/O registers is fixed regardless of the settings in A

Bit	Bit Name	Initial Value	R/W	Description
7	AST7	1	R/W	Area 7 to 0 Access State Control
6	AST6	1	R/W	These bits select whether the corresponding area
5	AST5	1	R/W	designated as a 2-state access space or a 3-state
4	AST4	1	R/W	space. Wait state insertion is enabled or disabled a same time.
3	AST3	1	R/W	0: Area n is designated for 2-state access
2	AST2	1	R/W	Wait state insertion in area n external space is d
1	AST1	1	R/W	1: Area n is designated for 3-state access
0	AST0	1	R/W	Wait state insertion in area n external space is e
				Note: n = 7 to 0

6	W70	1	R/W	These bits select the number of program wait sta area 7 in external address space is accessed wh AST7 bit in ASTCR is set to 1. 00: Program wait states are not inserted 01: 1 program wait state is inserted 10: 2 program wait states are inserted 11: 3 program wait states are inserted
5	W61	1	R/W	Area 6 Wait Control 1 and 0
4	W60	1	R/W	These bits select the number of program wait sta area 6 in external address space is accessed wh AST6 bit in ASTCR is set to 1. 00: Program wait states are not inserted 01: 1 program wait state is inserted 10: 2 program wait states are inserted 11: 3 program wait states are inserted
3	W51	1	R/W	Area 5 Wait Control 1 and 0
2	W50	1	R/W	These bits select the number of program wait sta area 5 in external address space is accessed wh AST5 bit in ASTCR is set to 1. 00: Program wait states are not inserted 01: 1 program wait state is inserted 10: 2 program wait states are inserted 11: 3 program wait states are inserted

- WCRL

Bit	Bit Name	Initial Value	R/W	Description
7	W31	1	R/W	Area 3 Wait Control 1 and 0
6	W30	1	R/W	These bits select the number of program wait state area 3 in external address space is accessed while AST3 bit in ASTCR is set to 1. 00: Program wait states are not inserted 01: 1 program wait state is inserted 10: 2 program wait states are inserted 11: 3 program wait states are inserted
5	W21	1	R/W	Area 2 Wait Control 1 and 0
4	W20	1	R/W	These bits select the number of program wait state area 2 in external address space is accessed while AST2 bit in ASTCR is set to 1. 00: Program wait states are not inserted 01: 1 program wait state is inserted 10: 2 program wait states are inserted 11: 3 program wait states are inserted

1	W01	1	R/W	Area 0 Wait Control 1 and 0
0	W00	1	R/W	These bits select the number of program wait sta area 0 in external address space is accessed wh AST0 bit in ASTCR is set to 1. 00: Program wait states are not inserted 01: 1 program wait state is inserted 10: 2 program wait states are inserted 11: 3 program wait states are inserted

				external read cycles in different areas 1: Idle cycle is inserted in case of consecutive ex read cycles in different areas
6	ICIS0	1	R/W	Idle Cycle Insertion 0 Selects whether or not one idle cycle state is to b inserted between bus cycles when consecutive e read and write cycles are performed. 0: Idle cycle is not inserted in case of consecutiv external read and write cycles 1: Idle cycle is inserted in case of consecutive ex read and write cycles
5	BRSTRM	0	R/W	Burst ROM Enable Selects whether area 0 is used as a burst ROM i 0: Area 0 is basic bus interface 1: Area 0 is burst ROM interface
4	BRSTS1	1	R/W	Burst Cycle Select 1 Selects the number of burst cycles for the burst I interface. 0: Burst cycle comprises 1 state 1: Burst cycle comprises 2 states
3	BRSTS0	0	R/W	Burst Cycle Select 0 Selects the number of words that can be access burst ROM interface burst access. 0: Max. 4 words in burst access 1: Max. 8 words in burst access

Bit	Bit Name	Initial Value	R/W	Description
7	BRLE	0	R/W	Bus Release Enable Enables or disables external bus release. 0: External bus release is disabled. $\overline{\text{BREQ}}$ and $\overline{\text{BA}}$ be used as I/O ports. 1: External bus release is enabled.
6	—	0	R/W	Reserved The write value should always be 0.
5	—	0	–	Reserved This bit is always read as 0 and cannot be modifi
4	—	0	R/W	Reserved The write value should always be 0.
3	—	1	R/W	Reserved The write value should always be 1.
2, 1	—	All 0	R/W	Reserved The write value should always be 0.
0	WAITE	0	R/W	WAIT Pin Enable Selects enabling or disabling of wait input by the 0: Wait input by the $\overline{\text{WAIT}}$ pin is disabled. The $\overline{\text{WA}}$ can be used as I/O port. 1: Wait input by the $\overline{\text{WAIT}}$ pin is enabled.

				WDT_1 selected by the PSS, CKS2 to CKS0 bits is BUZZ signal. 0: Functions as PF1 input/output pins 1: Functions as BUZZ output pins
4	—	0	R/W	Reserved The write value should always be 0.

0010: A8 and A9 output enabled. A10 to A23 outpu
disabled.

0011: A8 to A10 output enabled. A11 to A23 output

0100: A8 to A11 output enabled. A12 to A23 output

0101: A8 to A12 output enabled. A13 to A23 output

0110: A8 to A13 output enabled. A14 to A23 output

0111: A8 to A14 output enabled. A15 to A23 output

1000: A8 to A15 output enabled. A16 to A23 output

1001: A8 to A16 output enabled. A17 to A23 output

1010: A8 to A17 output enabled. A18 to A23 output

1011: A8 to A18 output enabled. A19 to A23 output

1100: A8 to A19 output enabled. A20 to A23 output

1101: A8 to A20 output enabled. A21 to A23 output

1110: A8 to A21 output enabled. A22 and A23 outp
disabled.

1111: A8 to A23 output enabled.

The chip select signals (CS0 to CS7) can be output for each area.

Note: * Not available in this LSI.

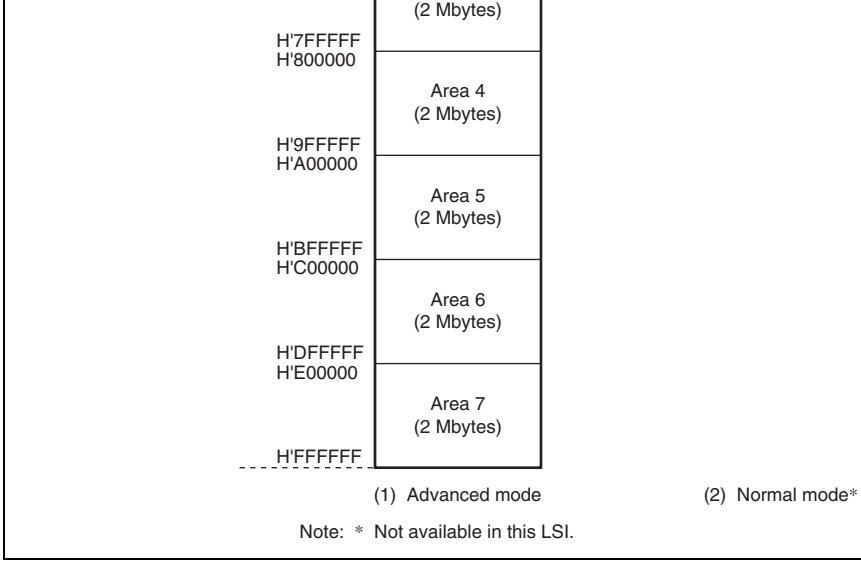


Figure 7.2 Overview of Area Divisions

7.4.2 Bus Specifications

The external address space bus specifications consist of three elements: bus width, numb
access states, and number of program wait states.

The bus width and number of access states for on-chip memory and internal I/O register
fixed, and are not affected by the bus controller.

Bus width: A bus width of 8 or 16 bits can be selected with ABWCR. An area for whic
bus is selected functions as an 8-bit access space, and an area for which a 16-bit bus is s
functions as a 16-bit access space.

When 2-state access space is designated, wait insertion is disabled.

Number of program wait states: When 3-state access space is designated by ASTCR, th number of program wait states to be inserted automatically is selected with WCRH and W From 0 to 3 program wait states can be selected.

Table 7.2 Bus Specifications for Each Area (Basic Bus Interface)

ABWCR	ASTCR	WCRH, WCRL		Bus Specifications (Basic Bus Interfac		
ABWn	ASTn	Wn1	Wn0	Bus Width	Number of Access States	Number of Pr Wait States
0	0	—	—	16	2	0
	1	0	0		3	0
			1			1
		1	0			2
			1			3
1	0	—	—	8	2	0
	1	0	0		3	0
			1			1
		1	0			2
			1			3

7.4.3 Bus Interface for Each Area

The initial state of each area is basic bus interface, 3-state access space. The initial bus wi selected according to the operating mode. The bus specifications described here cover bas only, and sections 7.6, Basic Bus Interface, and 7.7, Burst ROM Interface, on each memo interface should be referred to for further details.

space excluding on-chip RAM and internal I/O registers, is external address space. The
RAM is enabled when the RAME bit in the system control register (SYSCR) is set to 1;
RAME bit is cleared to 0, the on-chip RAM is disabled and the corresponding space bec
external address space.

When external address space of area 7 is accessed, the $\overline{CS7}$ signal can be output.

Only the basic bus interface can be used for area 7.

7.4.4 Chip Select Signals

This LSI can output chip select signals ($\overline{CS0}$ to $\overline{CS7}$) to areas 0 to 7, and these signals ar
low respectively when the corresponding external address space area is accessed. Figure
an example of $\overline{CSn}$ (n = 0 to 7) signal output timing. Enabling or disabling of the $\overline{CSn}$ si
performed by setting the data direction register (DDR) for the port corresponding to the
$\overline{CSn}$ pin.

In ROM-enabled extended mode, pins $\overline{CS0}$ to $\overline{CS7}$ are all placed in the input state after
on reset, and so the corresponding DDR should be set to 1 when outputting signals $\overline{CS0}$
For details, see section 9, I/O Ports.

Figure 7.3 CSn Signal Output Timing (n = 0 to 7)

word transfer. Figure 7.4 shows the on-chip memory access cycle. Figure 7.5 shows the

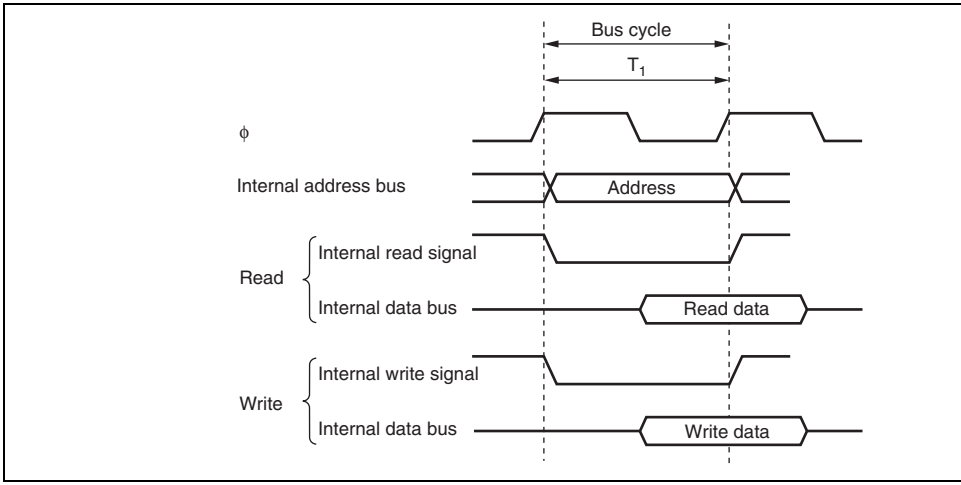


Figure 7.4 On-Chip Memory Access Cycle

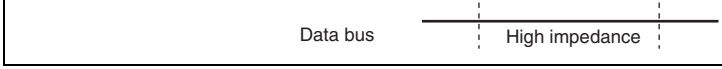


Figure 7.5 Pin States during On-Chip Memory Access

7.5.2 On-Chip Peripheral Module Access Timing

On-Chip Peripheral Module Access Timing Excluding Port H, Port J, IIC2, IEB, an HCAN: The on-chip peripheral modules are accessed in two states except for port H, por IEB, and HCAN. The data bus width is either 8 bits or 16 bits, depending on the particula I/O register being accessed. Figure 7.6 shows the access timing for the on-chip peripheral modules. Figure 7.7 shows the pin states.

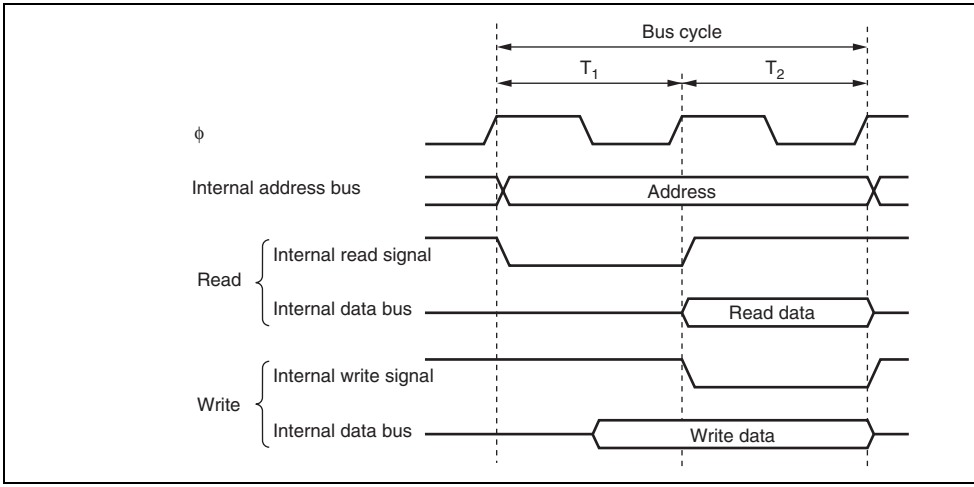


Figure 7.6 On-Chip Peripheral Module Access Cycle

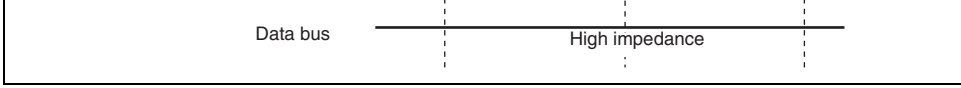


Figure 7.7 Pin States during On-Chip Peripheral Module Access

On-Chip Port H, Port J, and IIC2 Module Access Timing: On-chip port H, port J, an modules are accessed in four states. At this time, the data bus width is 8 bits. Figure 7.8 chip port H, port J, and IIC2 module access timing, and figure 7.9 shows the pin states.

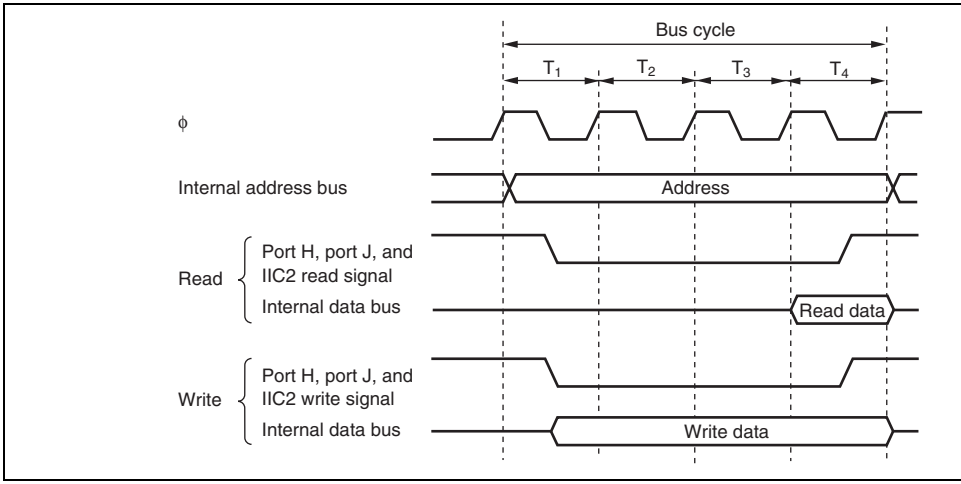


Figure 7.8 On-Chip Port H, Port J, and IIC2 Module Access Cycle

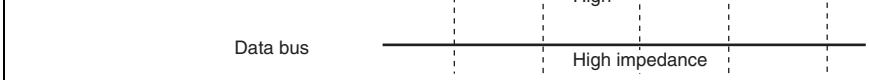


Figure 7.9 Pin States during On-Chip Port H, Port J, and IIC2 Module Acce

On-Chip IEB Module Access Timing (H8S/2552 Group Only): On-chip IEB module i accessed in four states. At this time, the data bus width is 8 bits. Figure 7.10 shows on-ch module access timing, and figure 7.11 shows the pin states.

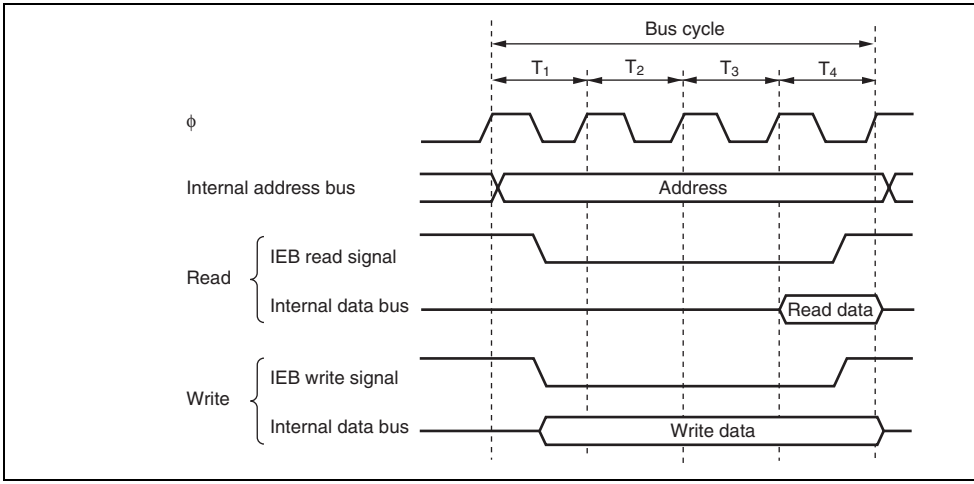


Figure 7.10 On-Chip IEB Module Access Cycle

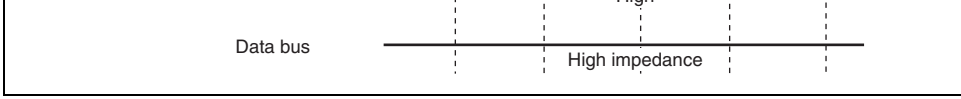


Figure 7.11 Pin States during On-Chip IEB Module Access

On-Chip HCAN Module Access Timing (H8S/2556 Group Only): On-chip HCAN m
accessed in five states. At this time, the data bus width is 16 bits. Figure 7.12 shows on-
HCAN module access timing, and figure 7.13 shows the pin states.

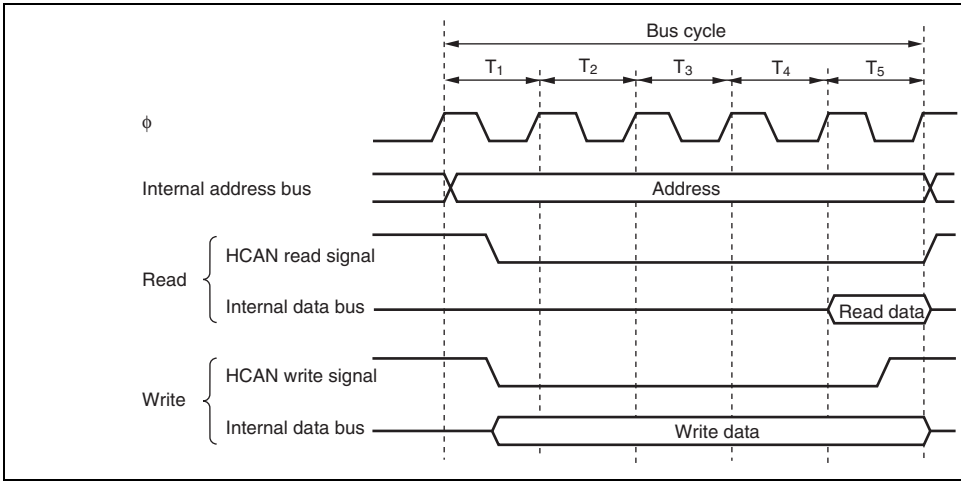


Figure 7.12 On-Chip HCAN Module Access Cycle

Figure 7.13 Pin States during On-Chip HCAN Module Access

7.5.3 External Address Space Access Timing

The external address space is accessed with an 8-bit or 16-bit data bus width in a two-stat
three-state bus cycle. In three-state access, wait states can be inserted. For further details,
section 7.6.3, Basic Timing.

size.

8-Bit Access Space: Figure 7.14 illustrates data alignment control for the 8-bit access sp
the 8-bit access space, the upper data bus (D15 to D8) is always used for accesses. The a
data that can be accessed at one time is one byte: a word transfer instruction is performe
byte accesses, and a longword transfer instruction, as four-byte accesses.

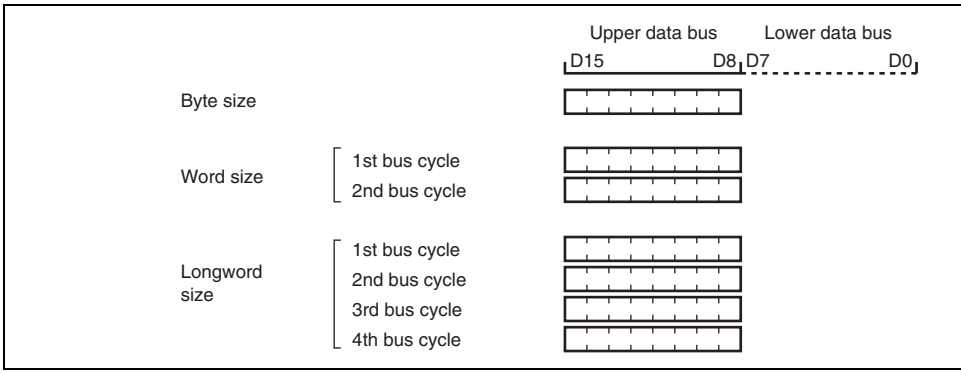


Figure 7.14 Access Sizes and Data Alignment Control (8-Bit Access Spac

16-Bit Access Space: Figure 7.15 illustrates data alignment control for the 16-bit access
With the 16-bit access space, the upper data bus (D15 to D8) and lower data bus (D7 to
used for accesses. The amount of data that can be accessed at one time is one byte or on
and a longword transfer instruction is performed as two-word accesses.

In byte access, whether the upper or lower data bus is used is determined by whether the
even or odd. The upper data bus is used for an even address, and the lower data bus for a
address.

7.6.2 Valid Strobes

Table 7.3 shows the data buses used and valid strobes for the access spaces.

In read access, the $\overline{RD}$ signal is valid without discrimination between the upper and lowe of the data bus.

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

Table 7.3 Data Buses Used and Valid Strobes

Area	Access Size	Read/ Write	Address	Valid Strobe	Upper Data Bus (D15 to D8)	Lower D (D7 to D
8-bit access space	Byte	Read	—	$\overline{RD}$	Valid	Invalid
		Write	—	$\overline{HWR}$	Valid	Hi-Z
16-bit access space	Byte	Read	Even	$\overline{RD}$	Valid	Invalid
			Odd	$\overline{RD}$	Invalid	Valid
		Write	Even	$\overline{HWR}$	Valid	Hi-Z
			Odd	$\overline{LWR}$	Hi-Z	Valid
	Word	Read	—	$\overline{RD}$	Valid	Valid
		Write	—	$\overline{HWR}$, $\overline{LWR}$	Valid	Valid

Legend:
Hi-Z: High impedance
Invalid: Input state; input value is ignored.

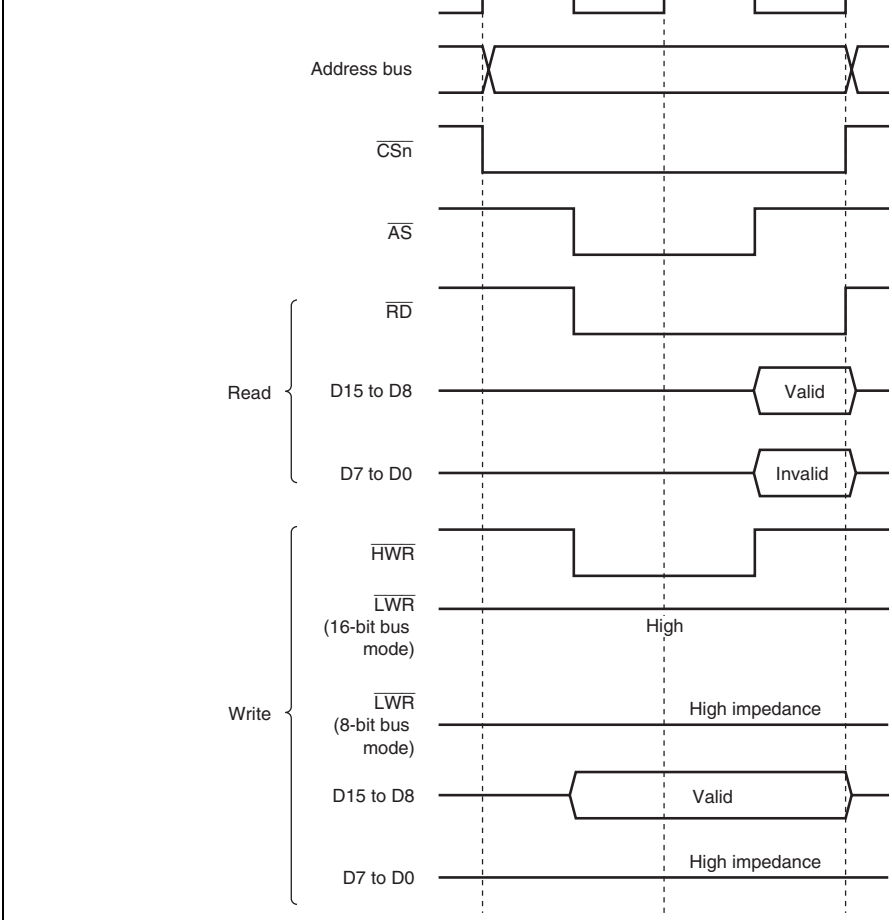


Figure 7.16 Bus Timing for 8-Bit 2-State Access Space

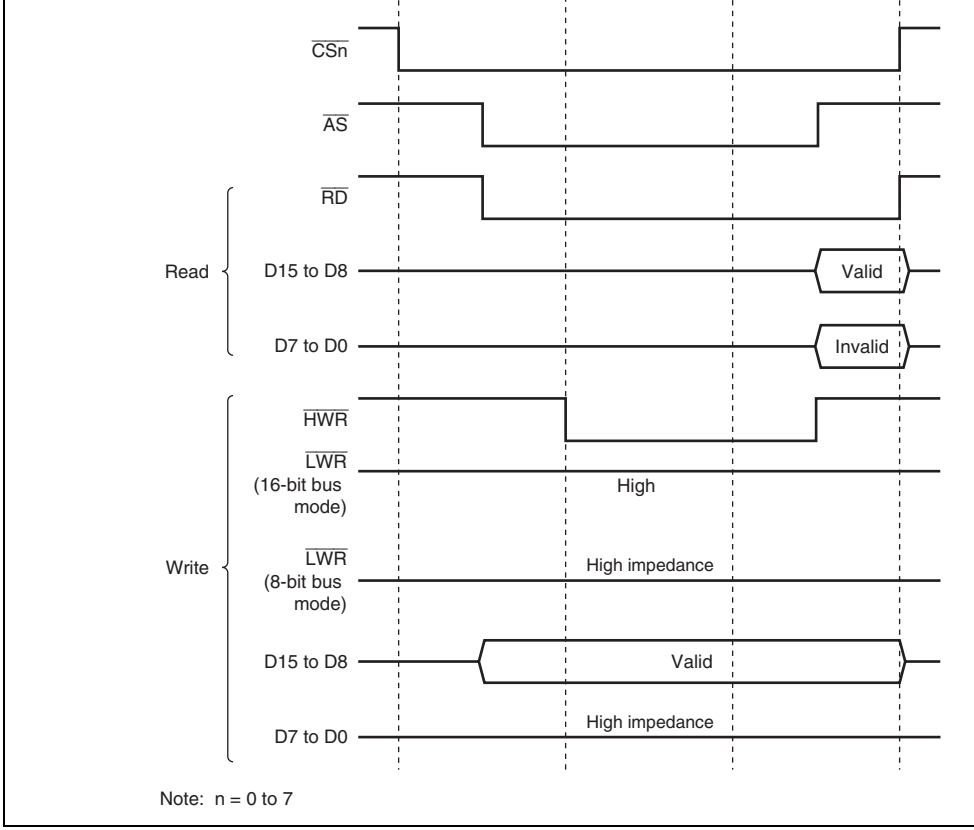


Figure 7.17 Bus Timing for 8-Bit 3-State Access Space

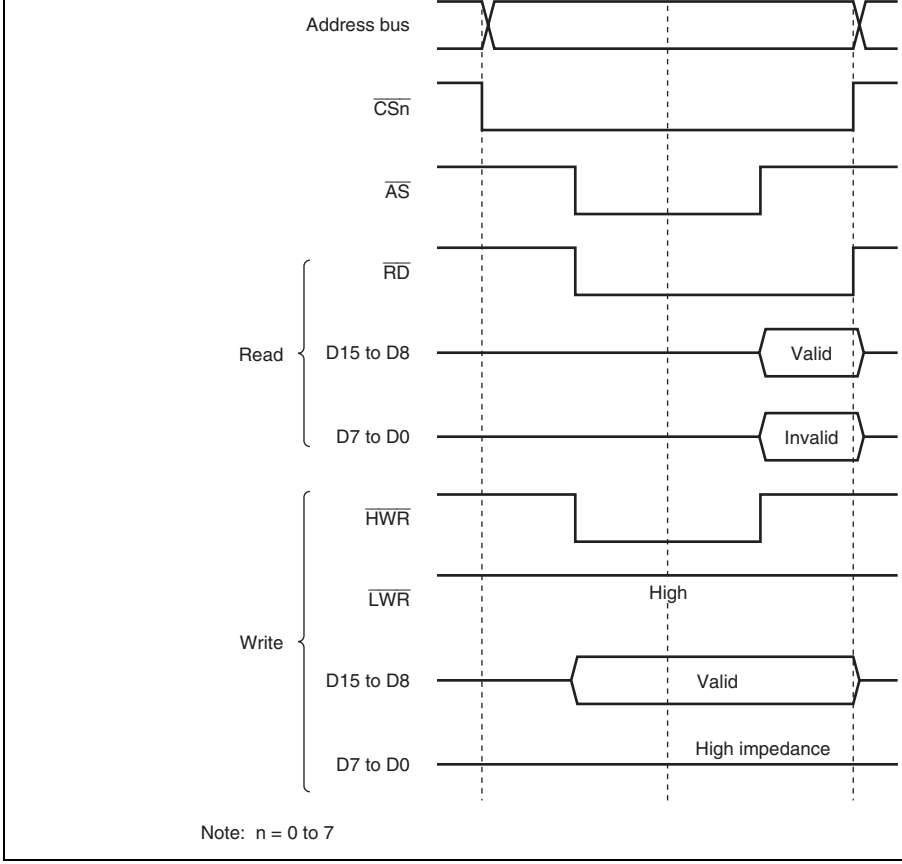


Figure 7.18 Bus Timing for 16-Bit 2-State Access Space (1) (Even Address Byte

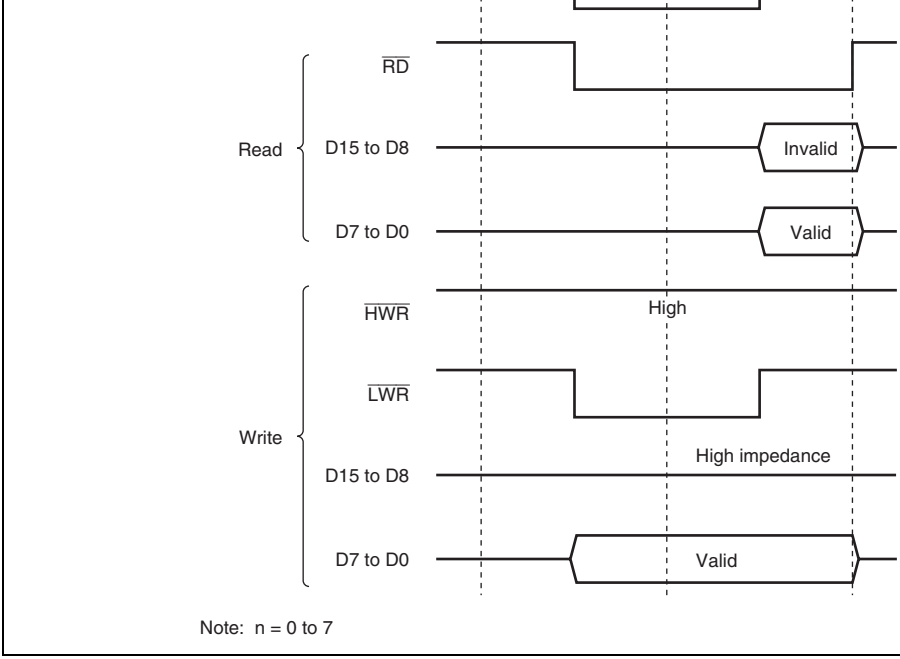


Figure 7.19 Bus Timing for 16-Bit 2-State Access Space (2) (Odd Address Byte A

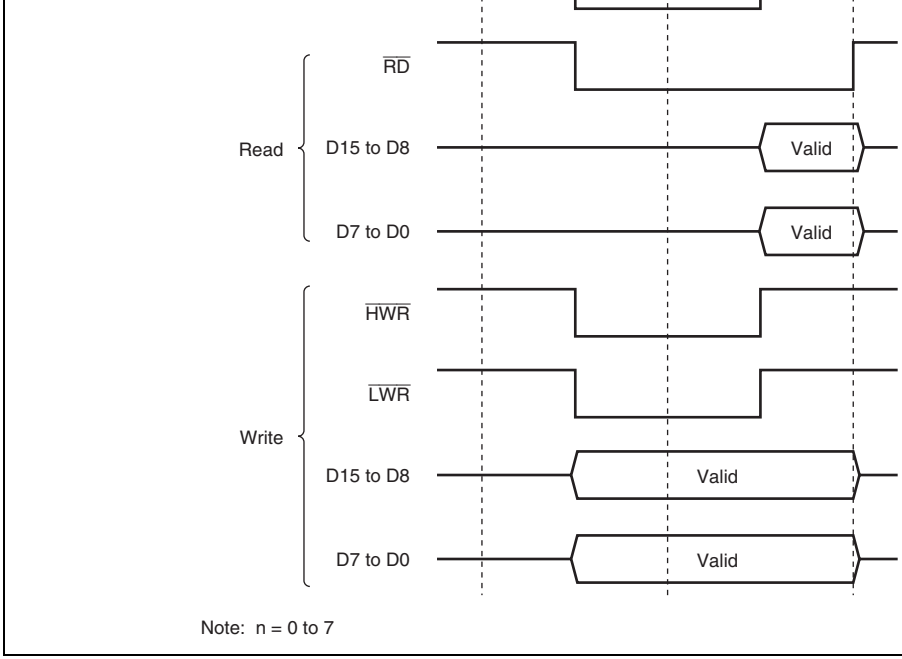


Figure 7.20 Bus Timing for 16-Bit 2-State Access Space (3) (Word Access

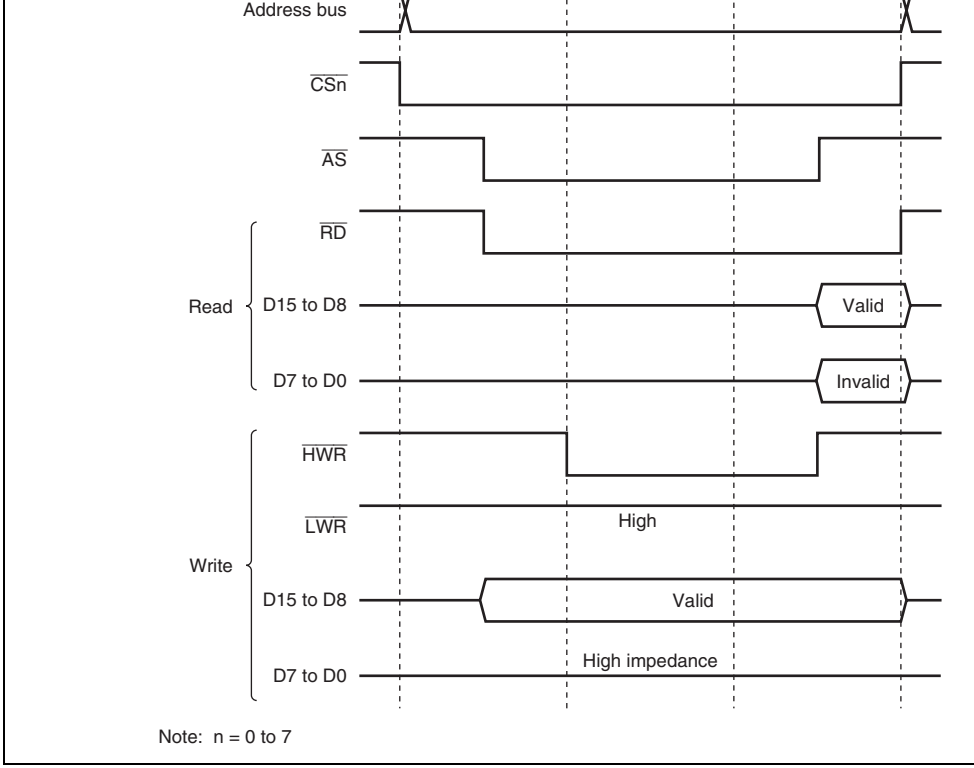


Figure 7.21 Bus Timing for 16-Bit 3-State Access Space (1) (Even Address Byte A

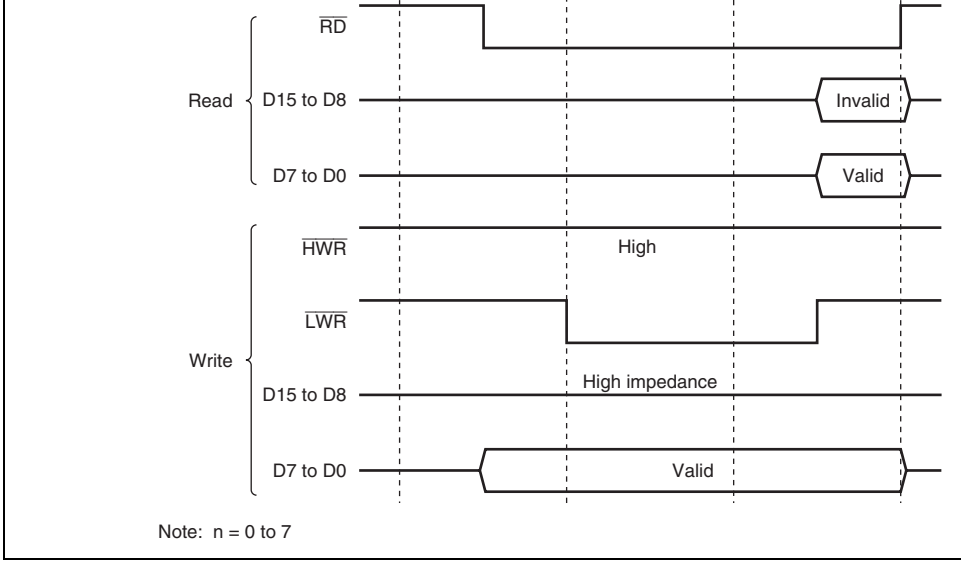


Figure 7.22 Bus Timing for 16-Bit 3-State Access Space (2) (Odd Address Byte

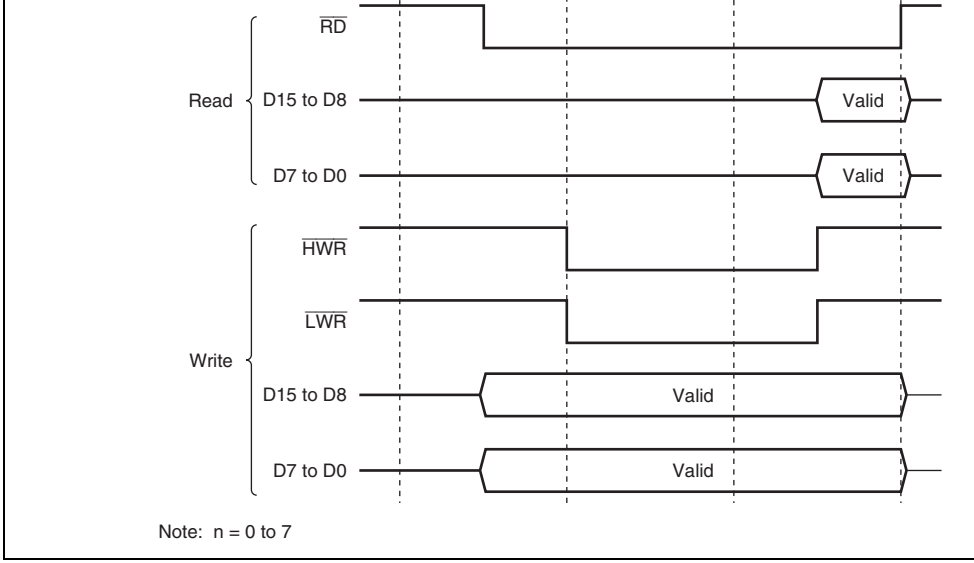


Figure 7.23 Bus Timing for 16-Bit 3-State Access Space (3) (Word Access)

carried out according to the settings of WCRH and WCRL. Then, if the $\overline{\text{WAIT}}$ pin is lo
falling edge of ϕ in the last T_2 or T_W state, a T_W state is inserted. If the $\overline{\text{WAIT}}$ pin is held
states are inserted until it goes high.

Figure 7.24 shows an example of wait state insertion timing.

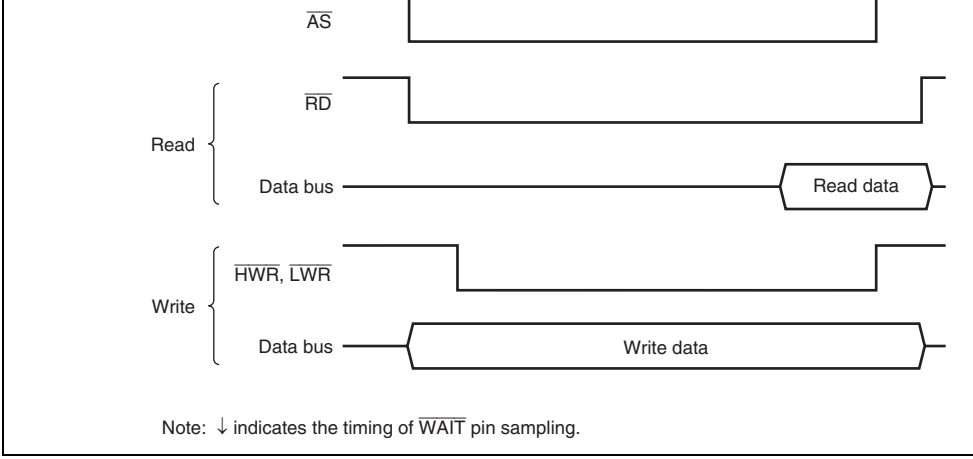


Figure 7.24 Example of Wait State Insertion Timing

7.7.1 Basic Timing

The number of access states in the initial cycle (full access) of the burst ROM interface i
accordance with the setting of the AST0 bit in ASTCR. Also, when the AST0 bit is set t
states can be inserted. One or two states can be selected for the burst cycle, according to
setting of the BRSTS1 bit in BCRH. Wait states cannot be inserted. When area 0 is desi
burst ROM space, it becomes 16-bit access space regardless of the setting of the ABW0
ABWCR.

When the BRSTS0 bit in BCRH is cleared to 0, burst access of up to 4 words is perform
the BRSTS0 bit is set to 1, burst access of up to 8 words is performed.

The basic access timing for burst ROM space is shown in figures 7.25 and 7.26. The tim
in figure 7.25 is for the case where the AST0 and BRSTS1 bits are both set to 1, and tha
7.26 is for the case where both these bits are cleared to 0.

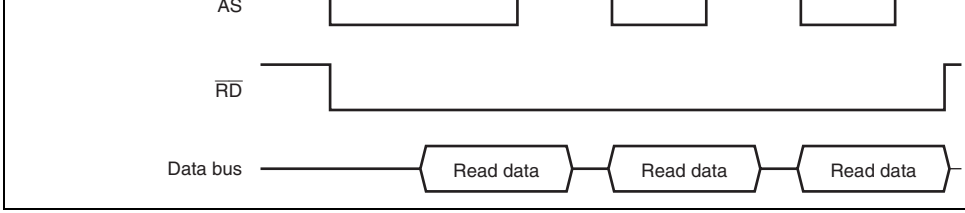


Figure 7.25 Example of Burst ROM Access Timing (When AST0 = BRSTS1 =

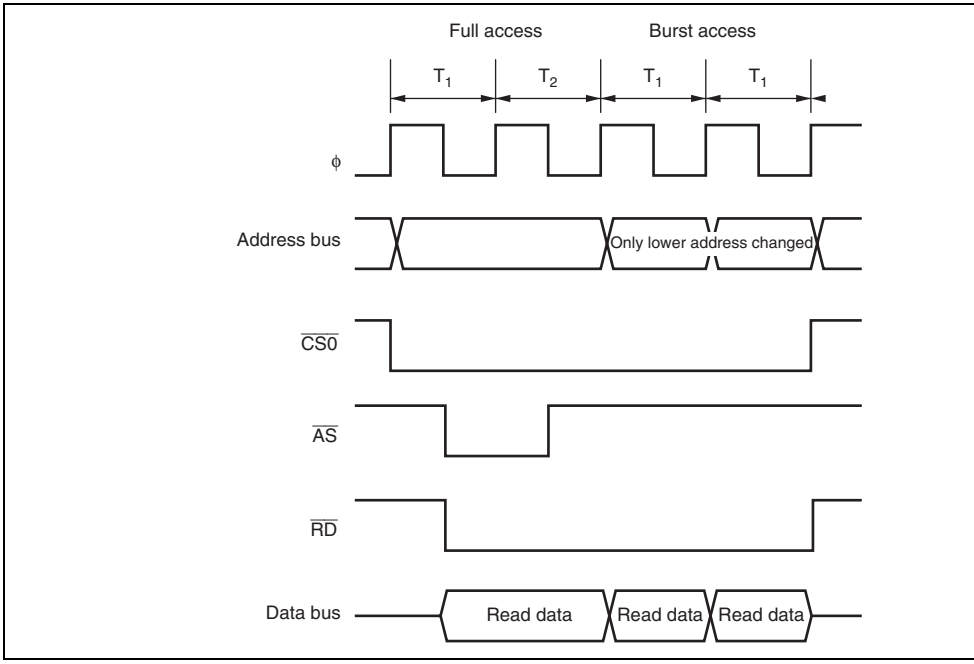


Figure 7.26 Example of Burst ROM Access Timing (When AST0 = BRSTS1 =

Figure 7.27 shows an example of the operation in this case. In this example, bus cycle A
cycle from ROM with a long output floating time, and bus cycle B is a read cycle from S
each being located in a different area. In (a), an idle cycle is not inserted, and a collision
cycle B between the read data from ROM and that from SRAM. In (b), an idle cycle is in
and a data collision is prevented.

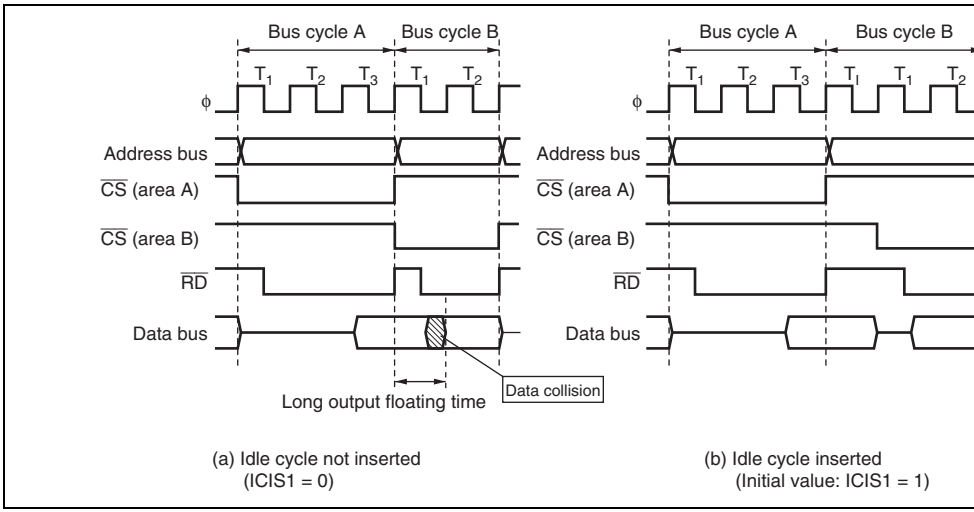


Figure 7.27 Example of Idle Cycle Operation (1)

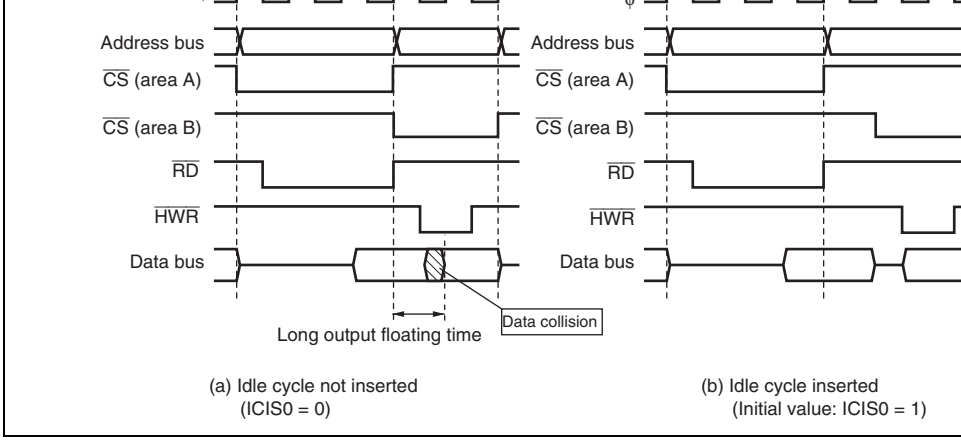


Figure 7.28 Example of Idle Cycle Operation (2)

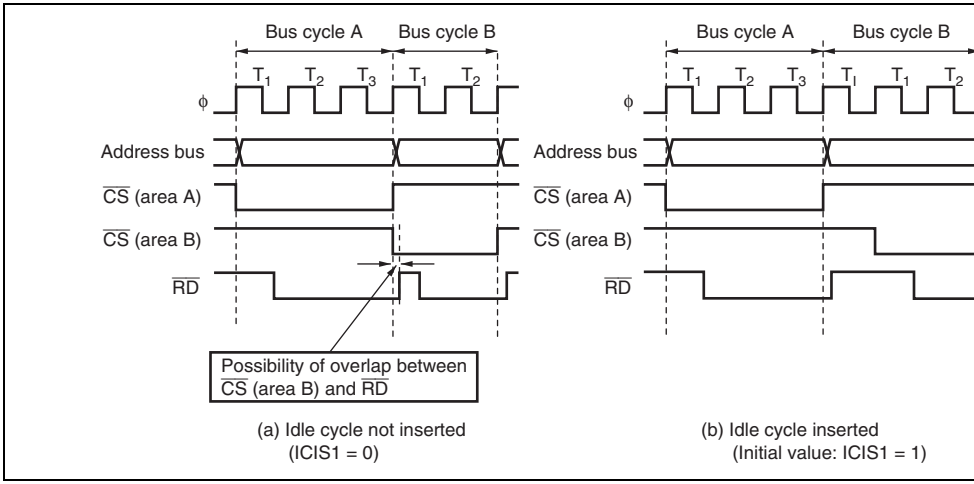


Figure 7.29 Relationship between Chip Select ($\overline{CS}$) and Read ($\overline{RD}$)

Table 7.4 shows the pin states in an idle cycle.

Table 7.4 Pin States in Idle Cycle

Pins	Pin State
A23 to A0	Contents of next bus cycle
D15 to D0	High impedance
$\overline{CSn}$	High
$\overline{AS}$	High
$\overline{RD}$	High
$\overline{HWR}$	High
$\overline{LWR}$	High

In the external bus mastership released state, an internal bus master can perform accesse
internal bus. When an internal bus master wants to make an external access, it temporari
activation of the bus cycle, and waits for the bus mastership request from the external bu
to be dropped.

When the $\overline{\text{BREQ}}$ pin is driven high, the $\overline{\text{BACK}}$ pin is driven high at the prescribed timin
external bus mastership released state is terminated.

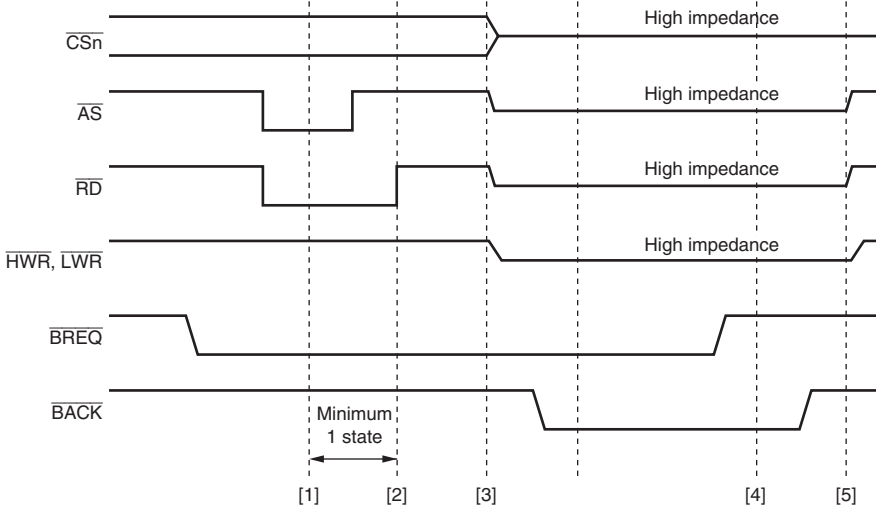
In the event of simultaneous external bus mastership release request and external access
generation, the order of priority is as follows:

(High) External bus mastership release > Internal bus master external access (L

Table 7.5 shows the pin states in the external bus mastership released state.

Table 7.5 Pin States in Bus Mastership Released State

Pins	Pin State
A23 to A0	High impedance
D15 to D0	High impedance
$\overline{\text{CSn}}$	High impedance
$\overline{\text{AS}}$	High impedance
$\overline{\text{RD}}$	High impedance
$\overline{\text{HWR}}$	High impedance
$\overline{\text{LWR}}$	High impedance



[1] The low level of the $\overline{\text{BREQ}}$ pin is sampled at the rise of T_2 state.
[2] The $\overline{\text{BACK}}$ pin is driven low at one state after the end of CPU read cycle, releasing bus mastership to external bus master.
[3] The $\overline{\text{BREQ}}$ pin state is still sampled in external bus mastership released state.
[4] The high level of the $\overline{\text{BREQ}}$ pin is sampled.
[5] The $\overline{\text{BACK}}$ pin is driven high, ending the bus mastership release cycles.

Note : n = 0 to 7

Figure 7.30 Bus Mastership Released State Transition Timing

7.9.1 Usage Note for Bus Mastership Release

When a transition to software standby mode or watch mode is made, external bus masters release function is aborted. In the state where MSTPCR is set to H'FFFFFF, and a transiti sleep mode is made, external bus mastership release function is aborted. When external b mastership release function is used in sleep mode, MSTPCR should not be set to H'FFFF

7.10.1 Operation

The bus arbiter detects the bus masters' bus mastership request signals, and if the bus m
is requested, sends a bus mastership request acknowledge signal to the bus master. If the
mastership requests from more than one bus master, the bus mastership request acknowl
signal is sent to the one with the highest priority. When a bus master receives the bus ma
request acknowledge signal, it takes possession of the bus until that signal is canceled.

The order of priority of the bus masters is as follows:

(High) DTC > CPU (Low)

An internal bus access by an internal bus master, and external bus mastership release, ca
executed in parallel.

In the event of simultaneous external bus mastership release request, and internal bus ma
external access request generation, the order of priority is as follows:

(High) External bus mastership release > Internal bus master external access (L

7.10.2 Bus Mastership Transfer Timing

Even if a bus mastership request is received from a bus master with a higher priority tha
the bus master that has acquired the bus and is currently operating, the bus mastership is
necessarily transferred immediately. There are specific times at which each bus master c
relinquish the bus mastership.

CPU: The CPU is the lowest-priority bus master, and if a bus mastership request is rece
the DTC, the bus arbiter transfers the bus mastership to the bus master that issued the re
timing for transfer of the bus mastership is as follows:

information write (3 states).

7.10.3 Usage Note for External Bus Mastership Release

External bus mastership release can be performed on completion of an external bus cycle signal remains low until the end of the external bus cycle. Therefore, when external bus mastership release is performed, the $\overline{CS}$ signal may change from the low level to the high impedance state.

7.11 Resets and the Bus Controller

In a power-on reset, this LSI, including the bus controller, enters the reset state at that poi an executing bus cycle is discontinued.

In a manual reset, the bus controller's registers and internal state are maintained, and an e external bus cycle is completed. In this case, $\overline{WAIT}$ input is ignored and write data is not guaranteed.

8.1 Features

- Transfer is possible over any number of channels
  - One activation source can trigger a number of data transfers (chain transfer)
- Three transfer modes
  - Normal, repeat, and block transfer modes are available
- The direct specification of 16-Mbyte address space is possible
- Transfer can be set in byte or word units
- A CPU interrupt can be requested for the interrupt that activated the DTC
- Activation by software is possible
- Module stop mode can be set

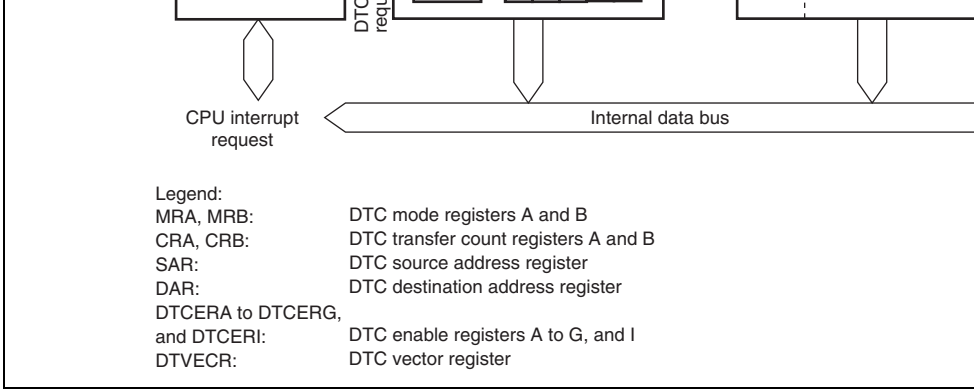


Figure 8.1 Block Diagram of DTC

These six registers cannot be directly accessed from the CPU. When activated, the DTC set of register information that is stored in on-chip RAM and transfers data to the corres DTC registers. After the data transfer, it writes a set of updated register inform ation bac RAM.

- DTC enable registers A to G, and I (DTCERA to DTCERG, and DTCERI)
- DTC vector register (DTVECR)

				10: SAR is incremented after a transfer (by +1 when Sz = 0; by +2 when Sz = 1) 11: SAR is decremented after a transfer (by –1 when Sz = 0; by –2 when Sz = 1)
5	DM1	Undefined	—	Destination Address Mode 1 and 0
4	DM0	Undefined	—	These bits specify a DAR operation after a data 0X: DAR is fixed 10: DAR is incremented after a transfer (by +1 when Sz = 0; by +2 when Sz = 1) 11: DAR is decremented after a transfer (by –1 when Sz = 0; by –2 when Sz = 1)
3	MD1	Undefined	—	DTC Mode 1 and 0
2	MD0	Undefined	—	These bits specify the DTC transfer mode. 00: Normal mode 01: Repeat mode 10: Block transfer mode 11: —
1	DTS	Undefined	—	DTC Transfer Mode Select Specifies whether the source side or the destina is set to be a repeat area or block area, in repea or block transfer mode. 0: Destination side is repeat area or block area 1: Source side is repeat area or block area

8.2.2 DTC Mode Register B (MRB)

MRB is an 8-bit register that selects the DTC operating mode.

Bit	Bit Name	Initial Value	R/W	Description
7	CHNE	Undefined	—	DTC Chain Transfer Enable This bit specifies a chain transfer. For details, section 8.5.4, Chain Transfer. In data transfer with CHNE set to 1, determina end of the specified number of transfers, clear interrupt source flag, and clearing of DTCER, a performed. 0: DTC data transfer completed (waiting for sta 1: DTC data transfer (reads new register inform and transfers data)
6	DISEL	Undefined	—	DTC Interrupt Select This bit specifies whether CPU interrupt is disa enabled after a data transfer. 0: Interrupt request is issued to the CPU when specified data transfer is completed. (The D the interrupt request flag that causes the ac 1: The DTC issues interrupt request to the CP data transfer. (The DTC does not clear the i request flag that causes the activation.)
5 to 0	—	Undefined	—	Reserved These bits have no effect on the DTC operatio write value should always be 0.

8.2.5 DTC Transfer Count Register A (CRA)

CRA is a 16-bit register that designates the number of times that data is transferred by the

In normal mode, the entire CRA functions as a 16-bit transfer counter (1 to 65536). It is decremented by 1 every time data is transferred, and transfer ends when the count reaches

In repeat mode or block transfer mode, the CRA is divided into two parts; the upper 8 bits (CRAH) and the lower 8 bits (CRAL). In repeat mode, CRAH holds the number of transf CRAL functions as an 8-bit transfer counter (1 to 256). In block transfer mode, CRAH ho block size while CRAL functions as an 8-bit block size counter (1 to 256). CRAL is decre by 1 every time data is transferred, and the contents of CRAH are sent when the count rea H'00. These operations are repeated.

8.2.6 DTC Transfer Count Register B (CRB)

CRB is a 16-bit register that designates the number of times that data is transferred by the block transfer mode. It functions as a 16-bit transfer counter (1 to 65536) that is decremer every time data is transferred, and transfer ends when the count reaches H'0000.

8.2.7 DTC Enable Registers A to G, and I (DTCERA to DTCERG, and DTCER

DTCER is a set of registers to specify the DTC activation interrupt source, and comprised registers; DTCERA to DTCERG, and DTCERI. The correspondence between interrupt so and DTCE bits, and vector numbers generated by the interrupt controller are shown in tab For DTCE bit setting, use bit manipulation instructions such as BSET and BCLR for read writing. When multiple activation sources are to be set at one time, only at the initial setti writing data is enabled after executing a dummy read on the relevant register with all the being masked.

				[Retaining condition]
				• When the DISEL bit is 0 and the specified nur transfers have not been completed

(n = A to G, and I)

8.2.8 DTC Vector Register (DTVECR)

DTVECR is an 8-bit readable/writable register that enables or disables DTC activation b software, and sets a vector number for the software activation interrupt.

Bit	Bit Name	Initial Value	R/W	Description
7	SWDTE	0	R/W	DTC Software Activation Enable
				Enables or disables the DTC software activation
				0: Disables the DTC software activation.
				1: Enables the DTC software activation.
				[Clearing conditions]
				• When the DISEL bit is 0 and the specified nur transfers have not ended
				• When 0 is written after a software-activated da transfer end interrupt (SWDTEND) request ha sent to the CPU
				[Retaining conditions]
				• When the DISEL bit is 1 and data transfer has
				• When the specified number of transfers have
				• When the software-activated data transfer is i

8.3 Activation Sources

The DTC operates when activated by an interrupt request or by a write to DTVECR by so
An activation interrupt request is specified by DTCER. When the corresponding bit is set
becomes DTC activation source and when it is cleared to 0, it becomes CPU interrupt sou
the end of a data transfer (or the last consecutive transfer in the case of chain transfer), th
interrupt flag of activation source or corresponding DTCER bit is cleared. Table 8.1 show
relationship between the activation source and DTCER clearing. The activation source fla
case of RXI0, for example, is the RDRF flag in SCI_0.

Since there are a number of DTC activation sources, transferring the last byte (or word) d
clear the flag of its activation source. Take appropriate steps at each interrupt processing.

When an interrupt has been designated as a DTC activation source, the existing CPU mas
and interrupt controller priorities have no effect. If there are more than one activation sou
the same time, the DTC operates in accordance with the default priority of the interrupt so

Figure 8.2 shows a block diagram of the DTC activation source control. For details, see s
Interrupt Controller.

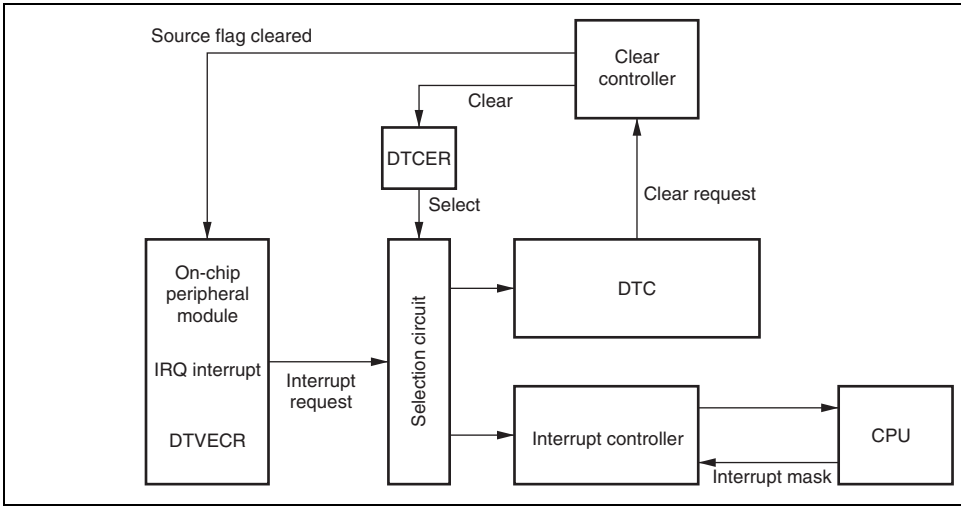


Figure 8.2 Block Diagram of DTC Activation Source Control

information from the vector address set for each activation source, and then reads the regi
information from that start address.

When the DTC is activated by software, the vector address is obtained from: H'0400 + (DTVECR[6:0] × 2). For example, if DTVECR is H'10, the vector address is H'0420.

The configuration of the vector address is the same in both normal* and advanced modes unit being used in both cases. These two bytes specify the lower bits of the register inform start address.

Note: Normal mode is not supported in this LSI.

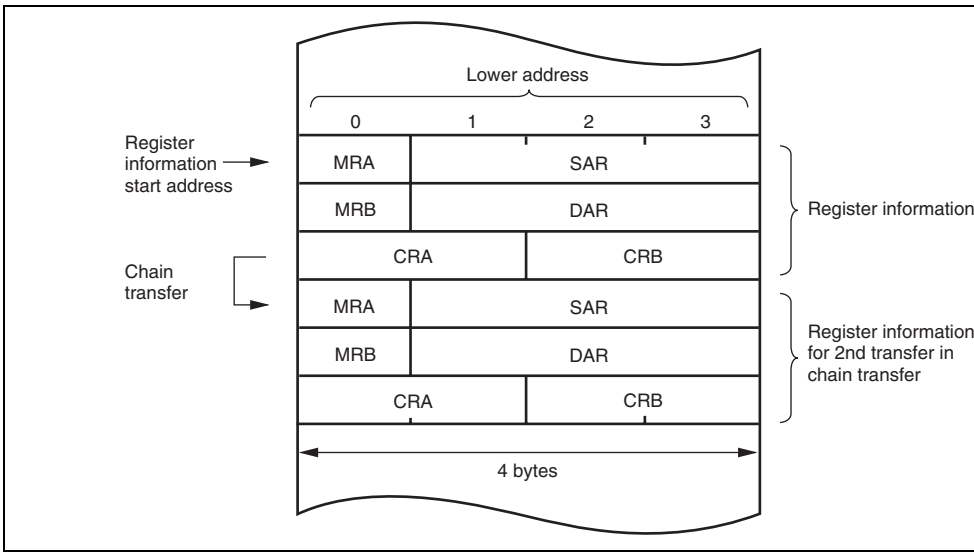


Figure 8.3 Location of DTC Register Information in Address Space

Figure 8.4 Correspondence between DTC Vector Address and Register Inform

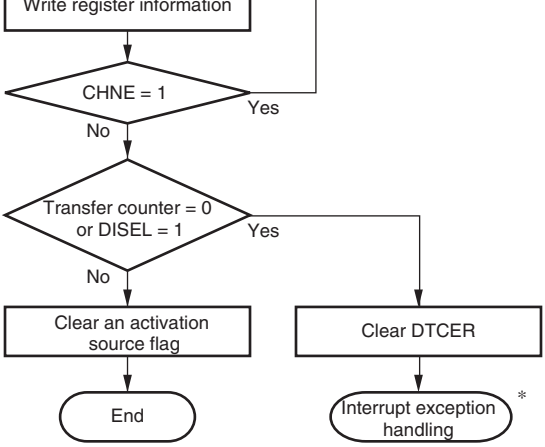
	IRQ4	20	H'0428	DTCEA3
	IRQ5	21	H'042A	DTCEA2
	IRQ6	22	H'042C	DTCEA1
	IRQ7	23	H'042E	DTCEA0
A/D converter	ADI (A/D conversion end)	28	H'0438	DTCEB6
TPU channel 0	TGI0A	32	H'0440	DTCEB5
	TGI0B	33	H'0442	DTCEB4
	TGI0C	34	H'0444	DTCEB3
	TGI0D	35	H'0446	DTCEB2
TPU channel 1	TGI1A	40	H'0450	DTCEB1
	TGI1B	41	H'0452	DTCEB0
TPU channel 2	TGI2A	44	H'0458	DTCEC7
	TGI2B	45	H'045A	DTCEC6
TPU channel 3	TGI3A	48	H'0460	DTCEC5
	TGI3B	49	H'0462	DTCEC4
	TGI3C	50	H'0464	DTCEC3
	TGI3D	51	H'0466	DTCEC2
TPU channel 4	TGI4A	56	H'0470	DTCEC1
	TGI4B	57	H'0472	DTCEC0
TPU channel 5	TGI5A	60	H'0478	DTCED5
	TGI5B	61	H'047A	DTCED4
8-bit timer channel 0	CMIA0	64	H'0480	DTCED3
	CMIB0	65	H'0482	DTCED2

	TXI2	90	H'04B4	DTCEF6
8-bit timer channel 2	CMIA2	92	H'04B8	DTCEF5
	CMIB2	93	H'04BA	DTCEF4
8-bit timer channel 3	CMIA3	96	H'04C0	DTCEF3
	CMIB3	97	H'04C2	DTCEF2
IEB (H8S/2552 Group only)	IERxI	105	H'04D2	DTCEG6
	IETxI	106	H'04D4	DTCEG5
HCAN (H8S/2556 Group only)	RM0	109	H'04DA	DTCEG2
SCI channel 3	RXI3	121	H'04F2	DTCEI7
	TXI3	122	H'04F4	DTCEI6
SCI channel 4	RXI4	125	H'04FA	DTCEI5
	TXI4	126	H'04FC	DTCEI4

Note: * The DTCE bits with no corresponding interrupt are reserved, and should be w
0.

The 24-bit SAR designates the DTC transfer source address, and the 24-bit DAR designa
transfer destination address. After each transfer, SAR and DAR are independently increm
decremented, or left fixed depending on its register information.

Figure 8.5 shows the flowchart of DTC operation.



Note: * For details, see the section related to each peripheral module.

Figure 8.5 Flowchart of DTC Operation

8.5.1 Normal Mode

In normal mode, one operation transfers one byte or one word of data.

From 1 to 65,536 transfers can be specified. Once the specified number of transfers have completed, a CPU interrupt can be requested.

Table 8.3 lists the register function in normal mode. Figure 8.6 shows the memory mapp normal mode.

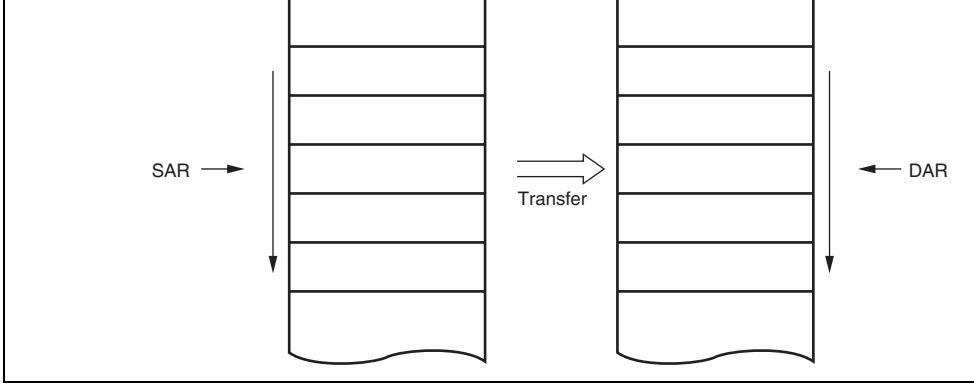


Figure 8.6 Memory Mapping in Normal Mode

8.5.2 Repeat Mode

In repeat mode, one operation transfers one byte or one word of data.

From 1 to 256 transfers can be specified. Once the specified number of transfers have enc
initial state of the transfer counter and the address register specified as the repeat area is r
and transfer is repeated. In repeat mode, the transfer counter value does not reach H'00, a
therefore CPU interrupts cannot be requested when DISEL = 0.

Table 8.4 lists the register function in repeat mode. Figure 8.7 shows the memory mappin
repeat mode.

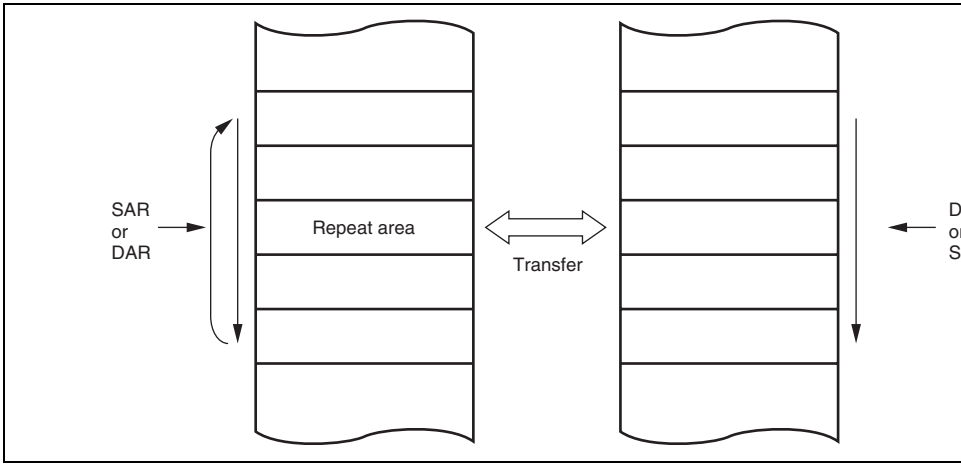


Figure 8.7 Memory Mapping in Repeat Mode

8.5.3 Block Transfer Mode

In block transfer mode, one operation transfers one block of data. Either the transfer sou
transfer destination is designated as a block area.

The block size can be between 1 to 256. When the transfer of one block ends, the initial
the block size counter and the address register specified as the block area is restored. Th
address register is then incremented, decremented, or left fixed depending on its register
information.

From 1 to 65,536 transfers can be specified. Once the specified numbers of transfers hav
completed, a CPU interrupt is requested.

DTC transfer count register B	CRB	Designates transfer count

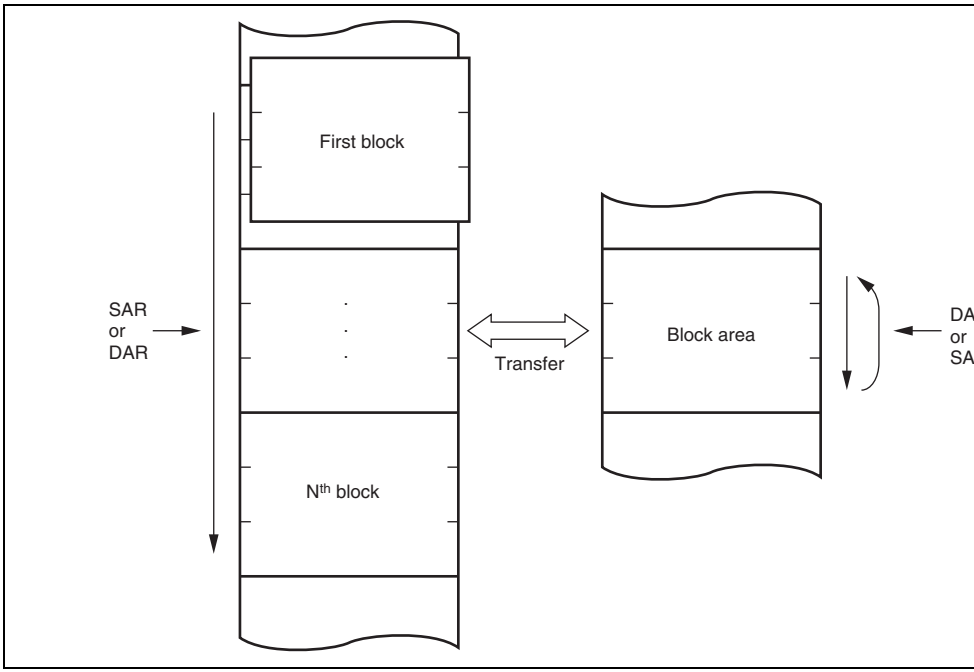


Figure 8.8 Memory Mapping in Block Transfer Mode

bit is cleared to 0.

In the case of transfer with CHNE set to 1, an interrupt request to the CPU is not genera
end of the specified number of transfers or by setting of the DISEL bit to 1, and the inte
request flag for the activation source is not affected.

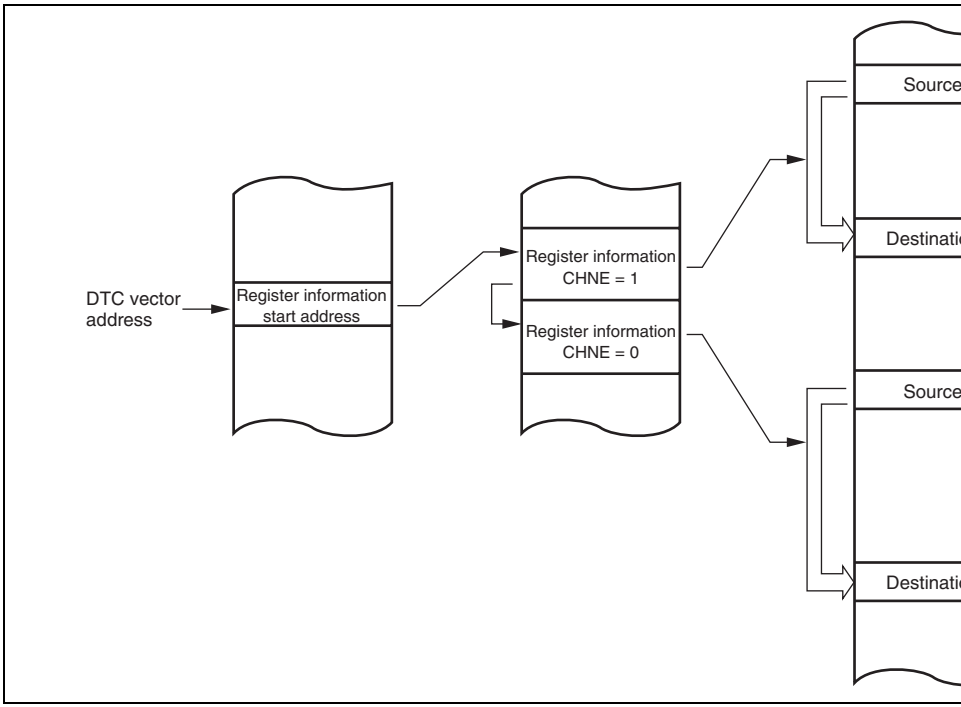


Figure 8.9 Chain Transfer Operation

transfers have been completed, the SWDTE bit is held at 1 and an SWDTEND interrupt r
generated after the end of data transfer. The interrupt handling routine will then clear the
bit to 0.

When the DTC is activated by software, an SWDTEND interrupt is not generated during
transfer wait or during data transfer even if the SWDTE bit is set to 1.

8.5.6 Operation Timing

Figures 8.10 to 8.12 show the DTC operation timing.

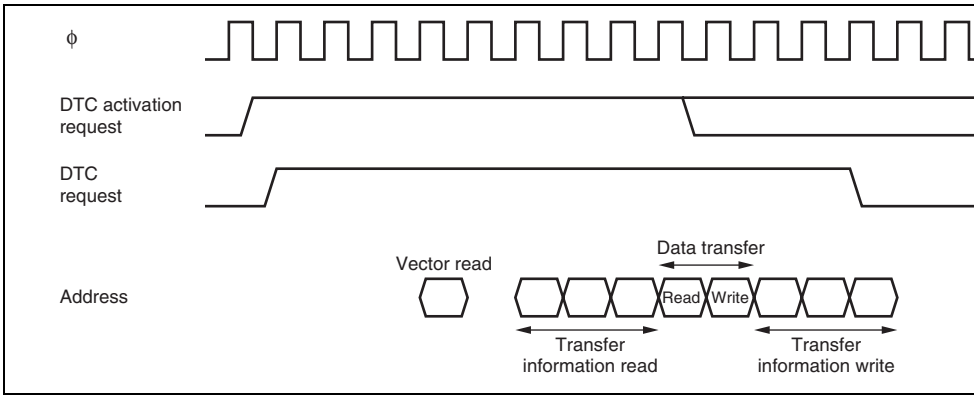


Figure 8.10 DTC Operation Timing (Example in Normal Mode or Repeat Mo

Information read Information write

Figure 8.11 DTC Operation Timing (Example in Block Transfer Mode, with Block Size of 2)

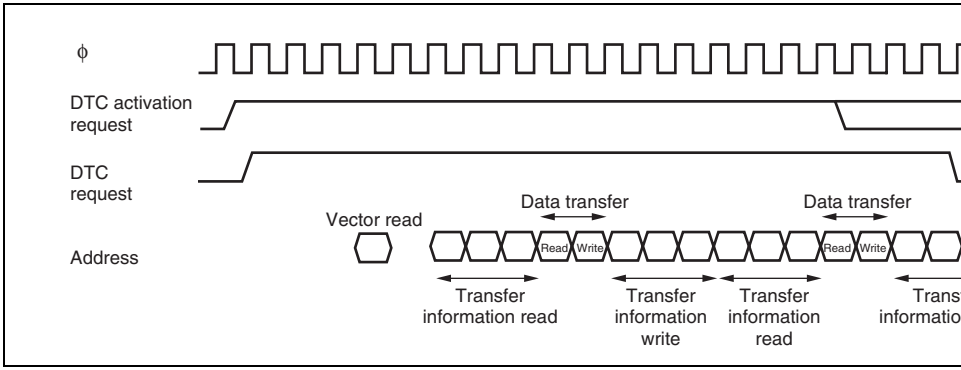


Figure 8.12 DTC Operation Timing (Example of Chain Transfer)

8.5.7 Number of DTC Execution States

Table 8.6 lists execution status for a single DTC data transfer, and table 8.7 lists the nun states required for each execution status.

Table 8.7 Number of States Required for Each Execution Status

Object to be Accessed		On-Chip RAM	On-Chip ROM	Internal I/O Registers				External Devices		
				IEB*1	HCAN*2	Other than IEB and HCAN				
Bus width		32	16	8	16	8	16	8	8	16
Access states		1	1	4	5	2	2	2	3	2
Execution Status	Vector read S_I	—	1	—	—	—	—	4	6 + 2m	2
	Register information read/write S_J	1	—	—	—	—	—	—	—	—
	Byte data read S_K	1	1	4	5	2	2	2	3 + m	2
	Word data read S_K	1	1	—	5	4	2	4	6 + 2m	2
	Byte data write S_L	1	1	4	5	2	2	2	3 + m	2
	Word data write S_L	1	1	—	5	4	2	4	6 + 2m	2
	Internal operation S_M	1	1	1	1	1	1	1	1	1

Legend:

m: The number of wait states for accessing external devices.

Notes: 1. H8S/2552 Group only.

2. H8S/2556 Group only.

The number of execution states is calculated from using the formula below. Note that Σ is of all transfers activated by one activation source (the number in which the CHNE bit is s plus 1).

Number of execution states = $I \cdot S_I + \Sigma (J \cdot S_J + K \cdot S_K + L \cdot S_L) + M \cdot S_M$

1. Set the MRA, MRB, SAR, DAR, CRA, and CRB register information in on-chip R
2. Set the start address of the register information in the DTC vector address.
3. Set the corresponding bit in DTCER to 1.
4. Set the enable bits for the interrupt sources to be used as the activation sources to 1. is activated when an interrupt used as an activation source is generated.
5. After one data transfer has been completed, or after the specified number of data tra been completed, the DTCE bit is cleared to 0 and a CPU interrupt is requested. If th to continue transferring data, set the DTCE bit to 1.

8.6.2 Activation by Software

The procedure for using the DTC with software activation is as follows:

1. Set the MRA, MRB, SAR, DAR, CRA, and CRB register information in on-chip R
2. Set the start address of the register information in the DTC vector address.
3. Check that the SWDTE bit is 0.
4. Write 1 to SWDTE bit and the vector number to DTVECR.
5. Check the vector number written to DTVECR.
6. After one data transfer has been completed, if the DISEL bit is 0 and a CPU interru requested, the SWDTE bit is cleared to 0. If the DTC is to continue transferring dat SWDTE bit to 1. When the DISEL bit is 1, or after the specified number of data tra been completed, the SWDTE bit is held at 1 and a CPU interrupt is requested.

be received in, and CRA sets 128 (H'0080). CRB can be set to any value.

2. Set the start address of the register information in the DTC vector address.
3. Set the corresponding bit in DTCER to 1.
4. Set the SCI to the appropriate receive mode. Set the RIE bit in SCR to 1 to enable the reception complete (RXI) interrupt. Since the generation of a receive error during the reception operation will disable subsequent reception, the CPU should be enabled to receive error interrupts.
5. Each time the reception of one byte of data has been completed on the SCI, the RDR SSR is set to 1, an RXI interrupt is generated, and the DTC is activated. The receive transferred from RDR to RAM by the DTC. DAR is incremented and CRA is decrem The RDRF flag is automatically cleared to 0.
6. When CRA becomes 0 after the 128 data transfers have been completed, the RDRF f held at 1, the DTCE bit is cleared to 0, and an RXI interrupt request is sent to the CP interrupt handling routine will perform wrap-up processing.

8.7.2 Software Activation

An example is shown in which the DTC is used to transfer a block of 128 bytes of data by of software activation. The transfer source address is H'1000 and the transfer destination a H'2000. The vector number is H'60, so the vector address is H'04C0.

1. MRA sets the source address increment (SM1 = 1, SM0 = 0), destination address inc (DM1 = 1, DM0 = 0), block transfer mode (MD1 = 1, MD0 = 0), and byte size (Sz = DTS bit can be set to any value. MRB performs one block transfer by one interrupt (0). SAR sets the transfer source address (H'1000), DAR sets the transfer destination a (H'2000), and CRA sets 128 (H'8080). CRB sets 1 (H'0001).
2. Set the start address of the register information at the DTC vector address (H'04C0).

the SWDTE bit to 0 and perform other wrap-up processing.

8.8 Usage Notes

8.8.1 Module Stop Mode Setting

The DTC operation can be disabled or enabled using the module stop control register. T
setting is for the DTC operation to be enabled. Register access is disabled by setting mo
mode. Module stop mode cannot be set during the DTC operation. For details, see sectio
Power-Down Modes.

8.8.2 On-Chip RAM

The MRA, MRB, SAR, DAR, CRA, and CRB registers are all located in on-chip RAM.
DTC is used, the RAME bit in SYSCR should not be cleared to 0.

8.8.3 DTCE Bit Setting

For DTCE bit setting, use bit manipulation instructions such as BSET and BCLR for rea
writing. When multiple activation sources are to be set at one time, only at the initial set
writing data is enabled after executing a dummy read on the relevant register with all the
being masked.

Ports 3 and A include an open-drain control register (ODR) that controls the on/off state output buffer PMOS.

All the I/O ports can drive a single TTL load and a 30 pF capacitive load.

The P34 and P35 pins on port 3 are NMOS push pull outputs.

The $\overline{\text{IRQ}}$ pin is a schmitt trigger input.

		P11/TIOCB0	
		P10/TIOCA0	
Port 2	General I/O port also functioning as TPU I/O pins	P27/TIOCB5	
		P26/TIOCA5	
		P25/TIOCB4	
		P24/TIOCA4	
		P23/TIOCD3	
		P22/TIOCC3	
		P21/TIOCB3	
		P20/TIOCA3	
Port 3	General I/O port also functioning as I^2C bus interface 2 I/O pins, SCI I/O pins, and interrupt input pins	P37/TxD4	Open dra enabled
		P36/RxD4	
		P35/SCK1/SCK4/SCL0/$\overline{\text{IRQ5}}$	Schmitt t input ($\overline{\text{IRQ5}}$, $\overline{\text{IR}}$
		P34/RxD1/SDA0	
		P33/TxD1/SCL1	
		P32/SCK0/SDA1/$\overline{\text{IRQ4}}$	NMOS pu output (P SCK1, SC
		P31/RxD0	
		P30/TxD0	

Port 5	General I/O port also functioning as SCI I/O pins	P52/SCK2	
		P51/RxD2	
		P50/TxD2	
Port 7	General I/O port also functioning as SCI I/O pins, TMR I/O pins, bus control output pins, and manual reset input pins	P77/TxD3	
		P76/RxD3	
		P75/TMO3/SCK3	
		P74/TMO2/$\overline{\text{MRES}}$	
		P73/TMO1/$\overline{\text{CS7}}$	P73/TMO1
		P72/TMO0/$\overline{\text{CS6}}$	P72/TMO0
		P71/TMRI23/TMCI23/$\overline{\text{CS5}}$	P71/TMRI23/TMCI23
		P70/TMRI01/TMCI01/$\overline{\text{CS4}}$	P70/TMRI0/TMCI01
Port 9	General input port also functioning as A/D converter analog input and D/A converter analog output pins	P97/AN15/DA1	
		P96/AN14/DA0	
		P95/AN13	
		P94/AN12	
		P93/AN11	
		P92/AN10	
		P91/AN9	
		P90/AN8	

Port B	General I/O port also functioning as address output pins	PB7/A15	PB7	Built-in in up MOS
		PB6/A14	PB6	
		PB5/A13	PB5	
		PB4/A12	PB4	
		PB3/A11	PB3	
		PB2/A10	PB2	
		PB1/A9	PB1	
		PB0/A8	PB0	
Port C	General I/O port also functioning as address output pins	PC7/A7	PC7	Built-in in up MOS
		PC6/A6	PC6	
		PC5/A5	PC5	
		PC4/A4	PC4	
		PC3/A3	PC3	
		PC2/A2	PC2	
		PC1/A1	PC1	
		PC0/A0	PC0	
Port D	General I/O port also functioning as data I/O pins	D15	PD7	Built-in in up MOS
		D14	PD6	
		D13	PD5	
		D12	PD4	
		D11	PD3	
		D10	PD2	
		D9	PD1	
		D8	PD0	

Port F	General I/O port also functioning as system clock output pins, interrupt input pins, bus control I/O pins, A/D converter input pins, and BUZZ output pins	PF7/φ	PF7/φ	Schmitt input (IRQ3, I
		AS	PF6	
		RD	PF5	
		HWR	PF4	
		PF3/LWR/ADTRG/IRQ3	PF3/ADTRG/IRQ3	
		PF2/WAIT	PF2	
		PF1/BACK/BUZZ	PF1/BUZZ	
		PF0/BREQ/IRQ2	PF0/IRQ2	
Port G	General I/O port also functioning as bus control output pins, interrupt input pins, and IEB I/O pins*1	PG4/CS0	PG4	Schmitt input (IRQ7, I
		PG3/Rx*1/CS1*2	PG3/Rx*1*2	
		PG2/Tx*1/CS2*2	PG2/Tx*1*2	
		PG1/CS3/IRQ7	PG1/IRQ7	
		PG0/IRQ6	PG0/IRQ6	
Port H	General I/O port	PH7		
		PH6		
		PH5		
		PH4		
		PH3		
		PH2		
		PH1		
		PH0		

Notes: 1. The $\overline{Rx}$ and $\overline{Tx}$ of IEB are valid only in the H8S/2552 Group.
2. The PG3/$\overline{Rx}$/$\overline{CS1}$ and PG2/$\overline{Tx}$/$\overline{CS2}$ pins are not available in the H8S/2556 Gro

P1DDR specifies input or output of the port 1 pins using the individual bits. P1DDR can read; if it is, the read value is undefined. Since this register is a write-only register, bit-manipulation instructions should not be used when writing. See section 2.9.4, Access M Registers with Write-only Bits.

Bit	Bit Name	Initial Value	R/W	Description
7	P17DDR	0	W	When a pin is specified as a general purpo port, setting this bit to 1 makes the corresp port 1 pin an output pin. Clearing this bit to the pin an input pin.
6	P16DDR	0	W	
5	P15DDR	0	W	
4	P14DDR	0	W	
3	P13DDR	0	W	
2	P12DDR	0	W	
1	P11DDR	0	W	
0	P10DDR	0	W	

2	P12DR	0	R/W	
1	P11DR	0	R/W	
0	P10DR	0	R/W	

9.1.3 Port 1 Register (PORT1)

PORT1 shows port 1 pin states. This register cannot be modified.

Bit	Bit Name	Initial Value	R/W	Description
7	P17	—*	R	If a port 1 read is performed while P1DDR I
6	P16	—*	R	set to 1, the P1DR values are read. If a por
5	P15	—*	R	is performed while P1DDR bits are cleared pin states are read.
4	P14	—*	R	
3	P13	—*	R	
2	P12	—*	R	
1	P11	—*	R	
0	P10	—*	R	

Note: * Determined by the states of pins P17 to P10.

9.1.4 Pin Functions

Port 1 pins also function as TPU I/O pins and interrupt input pins. Port 1 pin functions ar below.

2. This pin functions as TIOCB2 input when TPU channel 2 timer operating mod normal operation or phase counting mode and IOB3 in TIOR_2 is set to 1.
3. This pin functions as TCLKD input when TPSC2 to TPSC0 in TCR_0 or TCR to 111. This pin also functions as TCLKD input when channel 2 or 4 is set to counting mode.

- P16/TIOCA2/$\overline{\text{IRQ1}}$

  The pin function is switched as shown below according to the combination of the TP 2 setting and the P16DDR bit.

TPU Channel 2 Setting*[1]	Output	Input or Initial Value	
P16DDR	—	0	1
Pin function	TIOCA2 output	P16 input	P16 o
		TIOCA2 input*[2]	
	$\overline{\text{IRQ1}}$ input*[3]		

Notes: 1. For the setting of the TPU channel, see section 10, 16-Bit Timer Pulse Unit (T
2. This pin functions as TIOCA2 input when TPU channel 2 timer operating mod normal operation or phase counting mode and IOA3 in TIOR_2 is set to 1.
3. When this pin is used as an external interrupt pin, do not specify other functio

Notes: 1. For the setting of the TPU channel, see section 10, 16-Bit Timer Pulse Unit (TP
2. This pin functions as TIOCB1 input when TPU channel 1 timer operating mode
normal operation or phase counting mode and IOB3 to IOB0 in TIOR_1 are se
3. This pin functions as TCLKC input when TPSC2 to TPSC0 in TCR_0 or TCR_
to 110, or when TPSC2 to TPSC0 in TCR_4 or TCR_5 are set to 101. This pin
functions as TCLKC input when channel 2 or 4 is set to phase counting mode.

- P14/TIOCA1/$\overline{\text{IRQ0}}$

  The pin function is switched as shown below according to the combination of the TPU
  1 setting and the P14DDR bit.

TPU Channel 1 Setting*[1]	Output	Input or Initial Value	
P14DDR	—	0	1
Pin function	TIOCA1 output	P14 input	P14 ou
		TIOCA1 input*[2]	
	$\overline{\text{IRQ0}}$ input*[3]		

Notes: 1. For the setting of the TPU channel, see section 10, 16-Bit Timer Pulse Unit (TP
2. This pin functions as TIOCA1 input when TPU channel 1 timer operating mode
normal operation or phase counting mode and IOA3 to IOA0 in TIOR_1 are se
3. When this pin is used as an external interrupt pin, do not specify other function

2. This pin functions as TIOCD0 input when TPU channel 0 timer operating mod
normal operation and IOD3 to IOD0 in TIORL_0 are set to 10xx.
3. This pin functions as TCLKB input when TPSC2 to TPSC0 are set to 101 in a
TCR_0, TCR_1, and TCR_2. This pin also functions as TCLKB input when ch
5 is set to phase counting mode.

- P12/TIOCC0/TCLKA

The pin function is switched as shown below according to the combination of the TP
0 setting, the TPSC2 to TPSC0 bits in TCR_0, TCR_1, TCR_2, TCR_3, TCR_4, or
and the P12DDR bit.

TPU Channel 0 Setting*[1]	Output	Input or Initial Value	
P12DDR	—	0	1
Pin function	TIOCC0 output	P12 input	P12 o
		TIOCC0 input*[2]	
	TCLKA input*[3]		

Notes: 1. For the setting of the TPU channel, see section 10, 16-Bit Timer Pulse Unit (T
2. This pin functions as TIOCC0 input when TPU channel 0 timer operating mod
normal operation and IOC3 to IOC0 in TIORL_0 are set to 10xx.
3. This pin functions as TCLKA input when TPSC2 to TPSC0 are set to 100 in a
TCR_0, TCR_1, TCR_2, TCR_3, TCR_4, and TCR_5. This pin also functions
TCLKA input when channel 1 or 5 is set to phase counting mode.

normal operation and IOB3 to IOB0 in TIORH_0 are set to 10xx.

- P10/TIOCA0

  The pin function is switched as shown below according to the combination of the TPU
  0 setting and the P10DDR bit.

TPU Channel 0 Setting*[1]	Output	Input or Initial Value	
P10DDR	—	0	1
Pin function	TIOCA0 output	P10 input	P10 ou
		TIOCA0 input*[2]	

Notes: 1. For the setting of the TPU channel, see section 10, 16-Bit Timer Pulse Unit (TP
2. This pin functions as TIOCA0 input when TPU channel 0 timer operating mode
normal operation and IOA3 to IOA0 in TIORH_0 are set to 10xx.

P2DDR specifies input or output of the port 2 pins using the individual bits. P2DDR can read; if it is, the read value is undefined. Since this register is a write-only register, bit-manipulation instructions should not be used when writing. See section 2.9.4, Access M Registers with Write-only Bits.

Bit	Bit Name	Initial Value	R/W	Description
7	P27DDR	0	W	When a pin is specified as a general purpo port, setting this bit to 1 makes the corresp port 2 pin an output pin. Clearing this bit to the pin an input pin.
6	P26DDR	0	W	
5	P25DDR	0	W	
4	P24DDR	0	W	
3	P23DDR	0	W	
2	P22DDR	0	W	
1	P21DDR	0	W	
0	P20DDR	0	W	

2	P22DR	0	R/W	
1	P21DR	0	R/W	
0	P20DR	0	R/W	

9.2.3 Port 2 Register (PORT2)

PORT2 shows port 2 pin states. This register cannot be modified.

Bit	Bit Name	Initial Value	R/W	Description
7	P27	—*	R	If a port 2 read is performed while P2DDR b
6	P26	—*	R	set to 1, the P2DR values are read. If a port
5	P25	—*	R	performed while P2DDR bits are cleared to
4	P24	—*	R	states are read.
3	P23	—*	R	
2	P22	—*	R	
1	P21	—*	R	
0	P20	—*	R	

Note: * Determined by the states of pins P27 to P20.

Pin function	TIOCB5 output	P27 input	P27 o
		TIOCB5 input*2	

Notes: 1. For the setting of the TPU channel, see section 10, 16-Bit Timer Pulse Unit (T
2. This pin functions as TIOCB5 input when TPU channel 5 timer operating mod
normal operation or phase counting mode and IOB3 in TIOR_5 is set to 1.

- P26/TIOCA5

  The pin function is switched as shown below according to the combination of the TP
  5 setting and the P26DDR bit.

TPU Channel 5 Setting*1	Output	Input or Initial Value	
P26DDR	—	0	1
Pin function	TIOCA5 output	P26 input	P26 o
		TIOCA5 input*2	

Notes: 1. For the setting of the TPU channel, see section 10, 16-Bit Timer Pulse Unit (T
2. This pin functions as TIOCA5 input when TPU channel 5 timer operating mod
normal operation or phase counting mode and IOA3 in TIOR_5 is set to 1.

normal operation or phase counting mode and IOB3 to IOB0 in TIOR_4 are se

- P24/TIOCA4

  The pin function is switched as shown below according to the combination of the TPU 4 setting and the P24DDR bit.

TPU Channel 4 Setting*[1]	Output	Input or Initial Value	
P24DDR	—	0	1
Pin function	TIOCA4 output	P24 input	P24 ou
		TIOCA4 input*[2]	

Notes: 1. For the setting of the TPU channel, see section 10, 16-Bit Timer Pulse Unit (TP
2. This pin functions as TIOCA4 input when TPU channel 4 timer operating mode normal operation or phase counting mode and IOA3 to IOA0 in TIOR_4 are se

- P23/TIOCD3

  The pin function is switched as shown below according to the combination of the TPU 3 setting and the P23DDR bit.

TPU Channel 3 Setting*[1]	Output	Input or Initial Value	
P23DDR	—	0	1
Pin function	TIOCD3 output	P23 input	P23 ou
		TIOCD3 input*[2]	

Notes: 1. For the setting of the TPU channel, see section 10, 16-Bit Timer Pulse Unit (TP
2. This pin functions as TIOCD3 input when TPU channel 3 timer operating mode normal operation and IOD3 to IOD0 in TIORL_3 are set to 10xx.

normal operation and IOC3 to IOC0 in TIORL_3 are set to 10xx.

- P21/TIOCB3

  The pin function is switched as shown below according to the combination of the TP
  3 setting and the P21DDR bit.

TPU Channel 3 Setting*[1]	Output	Input or Initial Value	
P21DDR	—	0	1
Pin function	TIOCB3 output	P21 input	P21 o
		TIOCB3 input*[2]	

Notes: 1. For the setting of the TPU channel, see section 10, 16-Bit Timer Pulse Unit (T
2. This pin functions as TIOCB3 input when TPU channel 3 timer operating mod
normal operation and IOB3 to IOB0 in TIORH_3 are set to 10xx.

- P20/TIOCA3

  The pin function is switched as shown below according to the combination of the TP
  3 setting and the P20DDR bit.

TPU Channel 3 Setting*[1]	Output	Input or Initial Value	
P20DDR	—	0	1
Pin function	TIOCA3 output	P20 input	P20 o
		TIOCA3 input*[2]	

Notes: 1. For the setting of the TPU channel, see section 10, 16-Bit Timer Pulse Unit (T
2. This pin functions as TIOCA3 input when TPU channel 3 timer operating mod
normal operation and IOA3 to IOA0 in TIORH_3 are set to 10xx.

9.3.1 Port 3 Data Direction Register (P3DDR)

P3DDR specifies input or output of the port 3 pins using the individual bits. P3DDR cannr read; if it is, the read value is undefined. Since this register is a write-only register, bit-manipulation instructions should not be used when writing. See section 2.9.4, Access Me Registers with Write-only Bits.

Bit	Bit Name	Initial Value	R/W	Description
7	P37DDR	0	W	When a pin is specified as a general purpos port, setting this bit to 1 makes the correspo port 3 pin an output port. Clearing this bit to the pin an input port.
6	P36DDR	0	W	
5	P35DDR	0	W	
4	P34DDR	0	W	
3	P33DDR	0	W	
2	P32DDR	0	W	
1	P31DDR	0	W	
0	P30DDR	0	W	

2	P32DR	0	R/W	
1	P31DR	0	R/W	
0	P30DR	0	R/W	

9.3.3 Port 3 Register (PORT3)

PORT3 shows port 3 pin states. This register cannot be modified.

Bit	Bit Name	Initial Value	R/W	Description
7	P37	—*	R	If a port 3 read is performed while P3DDR set to 1, the P3DR values are read. If a por performed while P3DDR bits are cleared to states are read.
6	P36	—*	R	
5	P35	—*	R	
4	P34	—*	R	
3	P33	—*	R	
2	P32	—*	R	
1	P31	—*	R	
0	P30	—*	R	

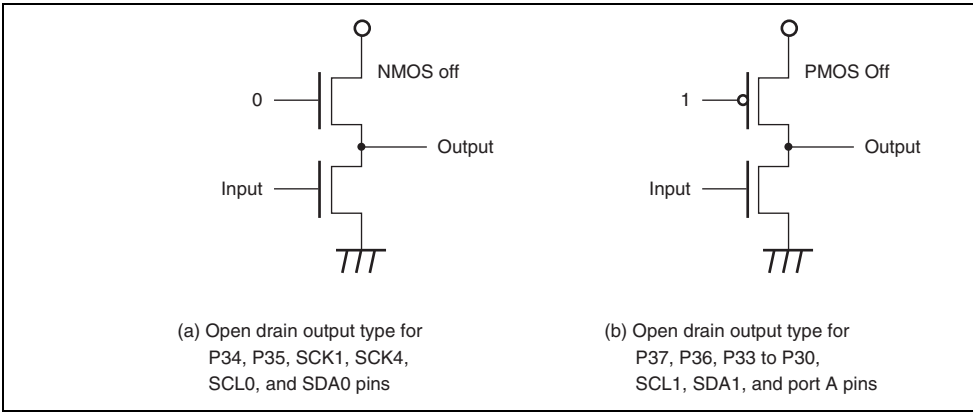
Note: * Determined by the states of pins P37 to P30.

2	P32ODR	0	R/W	the corresponding pins P33 and P34 functio NMOS open drain outputs. When cleared to corresponding pins function as NMOS push outputs.
1	P31ODR	0	R/W	
0	P30ODR	0	R/W	

9.3.5 Pin Functions

The port 3 pins also function as SCI I/O pins, I2C bus interface 2 I/O pins, and interrupt i
pins.

As shown in figure 9.1, when the pin P34, P35, SCK1, SCK4, SCL0, or SDA0 type open
output is used, a bus line is not affected even if the power supply for this LSI fails. Use (a
open drain output when using a bus line having a state in which the power is not supplied
LSI.



(a) Open drain output type for P34, P35, SCK1, SCK4, SCL0, and SDA0 pins

(b) Open drain output type for P37, P36, P33 to P30, SCL1, SDA1, and port A pins

Figure 9.1 Types of Open Drain Outputs

the specification in table 24.3, Permissible Output Currents.

- P37/TxD4

  The pin function is switched as shown below according to the combination of the TE
  SCR_4 of SCI_4 and the P37DDR bit.

TE	0		1
P37DDR	0	1	—
Pin function	P37 input	P37 output*	TxD4 ou

Note: * When P37ODR is set to 1, this pin functions as NMOS open drain output.

- P36/RxD4

  The pin function is switched as shown below according to the combination of the RE
  SCR_4 of SCI_4 and the P36DDR bit.

RE	0		1
P36DDR	0	1	—
Pin function	P36 input	P36 output*	RxD4 i

Note: * When P36ODR is set to 1, this pin functions as NMOS open drain output.

P35DDR	0	1	—	—	—	
Pin function	P35 input	P35 output*[1]	SCK1/ SCK4 output*[1]	SCK1/ SCK4 output*[1]	SCK1/SCK4 input	SC
	$\overline{\text{IRQ5}}$ input*[2]					

Notes: 1. When P35ODR is set to 1, this pin functions as NMOS open drain output. Whe to 0, this pin functions as NMOS push pull output.

2. When this pin is used as an external interrupt pin, do not specify other function

- P34/RxD1/SDA0

  The pin function is switched as shown below according to the combination of the ICE ICCR1_0 of IIC2_0, the RE bit in SCR_1 of SCI_1, and the P34DDR bit.

ICE	0			1
RE	0		1	—
P34DDR	0	1	—	—
Pin function	P34 input	P34 output*	RxD1 input	SDA0 inpu

Note: * When P34ODR is set to 1, this pin functions as NMOS open drain output. Whe to 0, this pin functions as NMOS push pull output.

- P32/SCK0/SDA1/IRQ4

  The pin function is switched as shown below according to the combination of the IC
  ICCR1_1 of IIC2_1, the C/$\overline{A}$ bit in SMR_0 of SCI_0, the CKE0 and CKE1 bits in S
  the P32DDR bit.

ICE	0					
CKE1	0				1	
C/$\overline{A}$	0			1	—	
CKE0	0		1	—	—	
P32DDR	0	1	—	—	—	
Pin function	P32 input	P32 output*[1]	SCK0 output*[1]	SCK0 output*[1]	SCK0 input	SI
	$\overline{IRQ4}$ Input*[2]					

Notes: 1. When P32ODR is set to 1, this pin functions as NMOS open drain output.
2. When this pin is used as an external interrupt pin, do not specify other functio

- P31/RxD0

  The pin function is switched as shown below according to the combination of the RE
  SCR_0 of SCI_0 and the P31DDR bit.

RE	0		1
P31DDR	0	1	—
Pin function	P31 input	P31 output*	RxD0 i

Note: * When P31ODR is set to 1, this pin functions as NMOS open drain output.

Port 4 is an 8-bit input-only port and has the following register.

- Port 4 register (PORT4)

9.4.1 Port 4 Register (PORT4)

PORT4 shows port 4 pin states. This register cannot be modified.

Bit	Bit Name	Initial Value	R/W	Description
7	P47	—*	R	The pin states are always read when a port is performed.
6	P46	—*	R	
5	P45	—*	R	
4	P44	—*	R	
3	P43	—*	R	
2	P42	—*	R	
1	P41	—*	R	
0	P40	—*	R	

Note: * Determined by the states of pins P47 to P40.

9.4.2 Pin Functions

Port 4 pins also function as A/D converter analog input pins (AN0 to AN7).

read; if it is, the read value is undefined. Since this register is a write-only register, bit-manipulation instructions should not be used when writing. See section 2.9.4, Access M Registers with Write-only Bits.

Bit	Bit Name	Initial Value	R/W	Description
7 to 3	—	Undefined	—	Reserved These bits are always read as undefined v cannot be modified.
2	P52DDR	0	W	When a pin is specified as a general purpo port, setting this bit to 1 makes the corresp port 5 pin an output pin. Clearing this bit to the pin an input pin.
1	P51DDR	0	W	
0	P50DDR	0	W	

9.5.2 Port 5 Data Register (P5DR)

P5DR stores output data for port 5 pins.

Bit	Bit Name	Initial Value	R/W	Description
7 to 3	—	Undefined	—	Reserved These bits are always read as undefined v cannot be modified.
2	P52DR	0	R/W	Output data for a pin is stored when the pi specified as a general purpose output port
1	P51DR	0	R/W	
0	P50DR	0	R/W	

				pin states are read.

Note: * Determined by the states of pins P52 to P50.

9.5.4 Pin Functions

Port 5 pins also function as SCI I/O pins. Port 5 pin functions are shown below.

- P52/SCK2

  The pin function is switched as shown below according to the combination of the C/$\overline{A}$ SMR_2 of SCI_2, the CKE0 and CKE1 bits in SCR_2, and the P52DDR bit.

CKE1	0				
C/$\overline{A}$	0			1	
CKE0	0		1	—	
P52DDR	0	1	—	—	
Pin function	P52 input	P52 output	SCK2 output	SCK2 output	SCK

- P51/RxD2

  The pin function is switched as shown below according to the combination of the RE SCR_2 of SCI_2 and the P51DDR bit.

RE	0		1
P51DDR	0	1	—
Pin function	P51 input	P51 output	RxD2 inp

Port 7 is an 8-bit I/O port and has the following registers.

- Port 7 data direction register (P7DDR)
- Port 7 data register (P7DR)
- Port 7 register (PORT7)

9.6.1 Port 7 Data Direction Register (P7DDR)

P7DDR specifies input or output of the port 7 pins using the individual bits. P7DDR can read; if it is, the read value is undefined. Since this register is a write-only register, bit-manipulation instructions should not be used when writing. See section 2.9.4, Access M Registers with Write-only Bits.

Bit	Bit Name	Initial Value	R/W	Description
7	P77DDR	0	W	When a pin is specified as a general purpo setting this bit to 1 makes the correspondin an output pin. Clearing this bit to 0 makes t input pin.
6	P76DDR	0	W	
5	P75DDR	0	W	
4	P74DDR	0	W	
3	P73DDR	0	W	
2	P72DDR	0	W	
1	P71DDR	0	W	
0	P70DDR	0	W	

2	P72DR	0	R/W	
1	P71DR	0	R/W	
0	P70DR	0	R/W	

9.6.3 Port 7 Register (PORT7)

PORT7 shows port 7 pin states. This register cannot be modified.

Bit	Bit Name	Initial Value	R/W	Description
7	P77	—*	R	If a port 7 read is performed while P7DDR b
6	P76	—*	R	set to 1, the P7DR values are read. If a port
5	P75	—*	R	performed while P7DDR bits are cleared to
4	P74	—*	R	states are read.
3	P73	—*	R	
2	P72	—*	R	
1	P71	—*	R	
0	P70	—*	R	

Note: * Determined by the states of pins P77 to P70.

Pin function	P77 input	P77 output	TxD3 ou

- P76/RxD3

  The pin function is switched as shown below according to the combination of the RE SCR_3 of SCI_3 and the P76DDR bit.

RE	0		1
P76DDR	0	1	—
Pin function	P76 input	P76 output	RxD3 in

- P75/TMO3/SCK3

  The pin function is switched as shown below according to the combination of the OS bits in TCSR_3 of TMR_3, the CKE1 and CKE0 bits in SCR_3 of SCI_3, the C/$\overline{A}$ b SMR_3, and the P75DDR bit.

OS3 to OS0	All bits are 0					A
CKE1	0				1	
C/$\overline{A}$	0			1	—	
CKE0	0		1	—	—	
P75DDR	0	1	—	—	—	
Pin function	P75 input	P75 output	SCK3 output	SCK3 output	SCK3 input	TM

The pin function is switched as shown below according to the combination of the ope
mode, the OS3 to OS0 bits in TCSR_1 of TMR_1, and the P73DDR bit.

Operating mode	Mode 6			Mode 7		
OS3 to OS0	All bits are 0		Any bit is 1	All bits are 0		An
P73DDR	0	1	—	0	1	
Pin function	P73 input	$\overline{CS7}$ output	TMO1 output	P73 input	P73 output	TM

- P72/TMO0/$\overline{CS6}$

  The pin function is switched as shown below according to the combination of the ope
  mode, the OS3 to OS0 bits in TCSR_0 of TMR_0, and the P72DDR bit.

Operating mode	Mode 6			Mode 7		
OS3 to OS0	All bits are 0		Any bit is 1	All bits are 0		An
P72DDR	0	1	—	0	1	
Pin function	P72 input	$\overline{CS6}$ output	TMO0 output	P72 input	P72 output	TM

The pin function is switched as shown below according to the combination of the op
mode and the P70DDR bit.

Operating mode	Mode 6		Mode 7	
P70DDR	0	1	0	
Pin function	P70 input	$\overline{CS4}$ output	P70 input	P70
	TMRI01/TMCI01 input			

7	P97	—*	R	The pin states are always read when a port performed.
6	P96	—*	R	
5	P95	—*	R	
4	P94	—*	R	
3	P93	—*	R	
2	P92	—*	R	
1	P91	—*	R	
0	P90	—*	R	

Note: * Determined by the states of pins P97 and P90.

9.7.2 Pin Functions

Port 9 pins also function as A/D converter analog input pins (AN15 and AN8) and D/A c
analog output pins (DA0 and DA1).

- P97/AN15/DA1

  The pin function is switched as shown below according to the combination of the DA
  the DAOE1 bit in DACR of D/A converter.

DAOE1	0		1
DAE	0	1	—
Pin function	P97 input	DA1 output	DA1 outpu
	AN15 input		

Pin function	P95, P94, P93, P92, P91, P90 input pin
	AN13, AN12, AN11, AN10, AN9, AN8 input

9.8.1 Port A Data Direction Register (PADDR)

PADDR specifies input or output the port A pins using the individual bits. PADDR cannot read; if it is, the read value is undefined. Since this register is a write-only register, bit-manipulation instructions should not be used when writing. See section 2.9.4, Access Methods for Registers with Write-only Bits.

Bit	Bit Name	Initial Value	R/W	Description
7	PA7DDR	0	W	When a pin is specified as a general purpose I/O port, setting this bit to 1 makes the corresponding port A pin an output pin. Clearing this bit to 0 makes the pin an input pin.
6	PA6DDR	0	W	
5	PA5DDR	0	W	
4	PA4DDR	0	W	
3	PA3DDR	0	W	
2	PA2DDR	0	W	
1	PA1DDR	0	W	
0	PA0DDR	0	W	

2	PA2DR	0	R/W
1	PA1DR	0	R/W
0	PA0DR	0	R/W

9.8.3 Port A Register (PORTA)

PORTA shows port A pin states. This register cannot be modified.

Bit	Bit Name	Initial Value	R/W	Description
7	PA7	—*	R	If a port A read is performed while PADDR
6	PA6	—*	R	set to 1, the PADR values are read. If a po
5	PA5	—*	R	is performed while PADDR bits are cleared pin states are read.
4	PA4	—*	R	
3	PA3	—*	R	
2	PA2	—*	R	
1	PA1	—*	R	
0	PA0	—*	R	

Note: * Determined by the states of pins PA7 to PA0.

2	PA2PCR	0	R/W	
1	PA1PCR	0	R/W	
0	PA0PCR	0	R/W	

9.8.5 Port A Open Drain Control Register (PAODR)

PAODR selects the output type for port A pins.

Bit	Bit Name	Initial Value	R/W	Description
7	PA7ODR	0	R/W	When these bits are set to 1, the correspond
6	PA6ODR	0	R/W	function as NMOS open drain outputs. Whe
5	PA5ODR	0	R/W	to 0, the corresponding pins function as CM
4	PA4ODR	0	R/W	outputs.
3	PA3ODR	0	R/W	
2	PA2ODR	0	R/W	
1	PA1ODR	0	R/W	
0	PA0ODR	0	R/W	

PA7DDR	—	0	1	0	
Pin function	A23 output	PA7 input	PA7 output*	PA7 input	PA

Note: * When PA7ODR in PAODR is set to 1, this pin functions as NMOS open drain

- PA6/A22

  The pin function is switched as shown below according to the combination of the op mode, the AE3 to AE0 bits in PFCR, and the PA6DDR bit.

Operating mode	Mode 6			Mode 7	
AE3 to AE0	B'1111	Other than B'1111		—	
PA6DDR	—	0	1	0	
Pin function	A22 output	PA6 input	PA6 output*	PA6 input	PA

Note: * When PA6ODR in PAODR is set to 1, this pin functions as NMOS open drain

- PA5/A21

  The pin function is switched as shown below according to the combination of the op mode, the AE3 to AE0 bits in PFCR, and the PA5DDR bit.

Operating mode	Mode 6			Mode 7	
AE3 to AE0	B'111x	Other than B'111x		—	
PA5DDR	—	0	1	0	
Pin function	A21 output	PA5 input	PA5 output*	PA5 input	PA

Legend:

x: Don't care

Note: * When PA5ODR in PAODR is set to 1, this pin functions as NMOS open drain

- PA3/A19

  The pin function is switched as shown below according to the combination of the ope
  mode, the AE3 to AE0 bits in PFCR, and the PA3DDR bit.

Operating mode	Mode 6			Mode 7	
AE3 to AE0	B'11xx	Other than B'11xx		—	
PA3DDR	—	0	1	0	
Pin function	A19 output	PA3 input	PA3 output*	PA3 input	PA3

Legend:

x: Don't care

Note: * When PA3ODR in PAODR is set to 1, this pin functions as NMOS open drain o

- PA2/A18

  The pin function is switched as shown below according to the combination of the ope
  mode, the AE3 to AE0 bits in PFCR, and the PA2DDR bit.

Operating mode	Mode 6			Mode 7	
AE3 to AE0	B'1011 or B'11xx	Other than B'1011 or B'11xx		—	
PA2DDR	—	0	1	0	
Pin function	A18 output	PA2 input	PA2 output*	PA2 input	PA2

Legend:

x: Don't care

Note: * When PA2ODR in PAODR is set to 1, this pin functions as NMOS open drain o

x: Don't care

Note: * When PA1ODR in PAODR is set to 1, this pin functions as NMOS open drain

- PA0/A16

  The pin function is switched as shown below according to the combination of the op mode, the AE3 to AE0 bits in PFCR, and the PA0DDR bit.

Operating mode	Mode 6			Mode 7	
AE3 to AE0	Other than B'0xxx or B'1000	B'0xxx or B'1000		—	
PA0DDR	—	0	1	0	
Pin function	A16 output	PA0 input	PA0 output*	PA0 input	PA

Legend:

x: Don't care

Note: * When PA0ODR in PAODR is set to 1, this pin functions as NMOS open drain

Port input	ON/OFF

Legend:
OFF: Input pull-up MOS is always off.
ON/OFF: On when PADDR = 0 and PAPCR = 1; otherwise off.

9.9 Port B

Port B is an 8-bit I/O port and has the following registers.

- Port B data direction register (PBDDR)
- Port B data register (PBDR)
- Port B register (PORTB)
- Port B pull-up MOS control register (PBPCR)

5	PB5DDR	0	W	the pin an input pin.
4	PB4DDR	0	W	
3	PB3DDR	0	W	
2	PB2DDR	0	W	
1	PB1DDR	0	W	
0	PB0DDR	0	W	

9.9.2 Port B Data Register (PBDR)

PBDR stores output data for port B pins.

Bit	Bit Name	Initial Value	R/W	Description
7	PB7DR	0	R/W	Output data for a pin is stored when the pin
6	PB6DR	0	R/W	specified as a general purpose output port.
5	PB5DR	0	R/W	
4	PB4DR	0	R/W	
3	PB3DR	0	R/W	
2	PB2DR	0	R/W	
1	PB1DR	0	R/W	
0	PB0DR	0	R/W	

2	PB2	—*	R
1	PB1	—*	R
0	PB0	—*	R

Note: * Determined by the states of pins PB7 to PB0.

9.9.4 Port B Pull-Up MOS Control Register (PBPCR)

PBPCR controls on/off state of the input pull-up MOS for port B pins.

Bit	Bit Name	Initial Value	R/W	Description
7	PB7PCR	0	R/W	When a pin is specified as an input port, set corresponding bit to 1 turns on the input pull for that pin.
6	PB6PCR	0	R/W	
5	PB5PCR	0	R/W	
4	PB4PCR	0	R/W	
3	PB3PCR	0	R/W	
2	PB2PCR	0	R/W	
1	PB1PCR	0	R/W	
0	PB0PCR	0	R/W	

PB7DDR	—	0	1	0	
Pin function	A15 output	PB7 input	PB7 output	PB7 input	PB7

Legend:
x: Don't care

- PB6/A14

  The pin function is switched as shown below according to the combination of the op
  mode, the AE3 to AE0 bits in PFCR, and the PB6DDR bit.

Operating mode	Mode 6			Mode 7	
AE3 to AE0	B'0111 or B'1xxx	Other than B'0111 or B'1xxx		—	
PB6DDR	—	0	1	0	
Pin function	A14 output	PB6 input	PB6 output	PB6 input	PB

Legend:
x: Don't care

x: Don't care

- PB4/A12

  The pin function is switched as shown below according to the combination of the ope
  mode, the AE3 to AE0 bits in PFCR, and the PB4DDR bit.

Operating mode	Mode 6			Mode 7	
AE3 to AE0	Other than B'0100 or B'00xx	B'0100 or B'00xx		—	
PB4DDR	—	0	1	0	
Pin function	A12 output	PB4 input	PB4 output	PB4 input	PB4

Legend:

x: Don't care

- PB3/A11

  The pin function is switched as shown below according to the combination of the ope
  mode, the AE3 to AE0 bits in PFCR, and the PB3DDR bit.

Operating mode	Mode 6			Mode 7	
AE3 to AE0	Other than B'00xx	B'00xx		—	
PB3DDR	—	0	1	0	
Pin function	A11 output	PB3 input	PB3 output	PB3 input	PB3

Legend:

x: Don't care

x: Don't care

- PB1/A9

  The pin function is switched as shown below according to the combination of the op
  mode, the AE3 to AE0 bits in PFCR, and the PB1DDR bit.

Operating mode	Mode 6			Mode 7	
AE3 to AE0	Other than B'000x	B'000x		—	
PB1DDR	—	0	1	0	
Pin function	A9 output	PB1 input	PB1 output	PB1 input	PB

Legend:
x: Don't care

- PB0/A8

  The pin function is switched as shown below according to the combination of the op
  mode, the AE3 to AE0 bits in PFCR, and the PB0DDR bit.

Operating mode	Mode 6			Mode 7	
AE3 to AE0	Other than B'0000	B'0000		—	
PB0DDR	—	0	1	0	
Pin function	A8 output	PB0 input	PB0 output	PB0 input	PB

Legend:
x: Don't care

Port input	ON/OFF

Legend:

OFF: Input pull-up MOS is always off.

ON/OFF: On when PBDDR = 0 and PBPCR = 1; otherwise off.

PCDDR specifies input or output the port C pins using the individual bits. PCDDR cann
if it is, the read value is undefined. Since this register is a write-only register, bit-manipu
instructions should not be used when writing. See section 2.9.4, Access Method for Reg
Write-only Bits.

Bit	Bit Name	Initial Value	R/W	Description
7	PC7DDR	0	W	When a pin is specified as a general purpo port, setting this bit to 1 makes the corresp port C pin an output pin. Clearing this bit to the pin an input pin.
6	PC6DDR	0	W	
5	PC5DDR	0	W	
4	PC4DDR	0	W	
3	PC3DDR	0	W	
2	PC2DDR	0	W	
1	PC1DDR	0	W	
0	PC0DDR	0	W	

2	PC2DR	0	R/W	
1	PC1DR	0	R/W	
0	PC0DR	0	R/W	

9.10.3 Port C Register (PORTC)

PORTC shows port C pin states. This register cannot be modified.

Bit	Bit Name	Initial Value	R/W	Description
7	PC7	—*	R	If a port C read is performed while PCDDR b
6	PC6	—*	R	set to 1, the PCDR values are read. If a port
5	PC5	—*	R	is performed while PCDDR bits are cleared t
4	PC4	—*	R	pin states are read.
3	PC3	—*	R	
2	PC2	—*	R	
1	PC1	—*	R	
0	PC0	—*	R	

Note: * Determined by the states of pins PC7 to PC0.

2	PC2PCR	0	R/W
1	PC1PCR	0	R/W
0	PC0PCR	0	R/W

9.10.5 Pin Functions

Port C pins also function as address output pins. Port C pin functions are shown below.

- PC7/A7, PC6/A6, PC5/A5, PC4/A4, PC3/A3, PC2/A2, PC1/A1, PC0/A0
  The pin function is switched as shown below according to the combination of the op
  mode and the PCnDDR bit.

Operating mode	Mode 6		Mode 7	
PCnDDR	0	1	0	
Pin function	PCn input	Address output	PCn input	PCn

Note: n = 7 to 0

Port input	ON/OFF

Legend:

OFF: Input pull-up MOS is always off.

ON/OFF: On when PCDDR = 0 and PCPCR = 1; otherwise off.

PDDDR specifies input or output the port D pins using the individual bits. PDDDR cann read; if it is, the read value is undefined. Since this register is a write-only register, bit-manipulation instructions should not be used when writing. See section 2.9.4, Access M Registers with Write-only Bits.

Bit	Bit Name	Initial Value	R/W	Description
7	PD7DDR	0	W	When a pin is specified as a general purpo port, setting this bit to 1 makes the corresp port D pin an output pin. Clearing this bit to the pin an input pin.
6	PD6DDR	0	W	
5	PD5DDR	0	W	
4	PD4DDR	0	W	
3	PD3DDR	0	W	
2	PD2DDR	0	W	
1	PD1DDR	0	W	
0	PD0DDR	0	W	

2	PD2DR	0	R/W	
1	PD1DR	0	R/W	
0	PD0DR	0	R/W	

9.11.3 Port D Register (PORTD)

PORTD shows port D pin states. This register cannot be modified.

Bit	Bit Name	Initial Value	R/W	Description
7	PD7	—*	R	If a port D read is performed while PDDDR b
6	PD6	—*	R	set to 1, the PDDR values are read. If a port
5	PD5	—*	R	is performed while PDDDR bits are cleared t
4	PD4	—*	R	pin states are read.
3	PD3	—*	R	
2	PD2	—*	R	
1	PD1	—*	R	
0	PD0	—*	R	

Note: * Determined by the states of pins PD7 to PD0.

2	PD2PCR	0	R/W
1	PD1PCR	0	R/W
0	PD0PCR	0	R/W

9.11.5 Pin Functions

Port D pins also function as data I/O pins. Port D pin functions are shown below.

- PD7/D15, PD6/D14, PD5/D13, PD4/D12, PD3/D11, PD2/D10, PD1/D9, PD0/D8
  The pin function is switched as shown below according to the combination of the op
  mode and the PDnDDR bit.

Operating mode	Mode 6	Mode 7	
PDnDDR	—	0	1
Pin function	Data input/output	PDn input	PDn ou

Note: n = 7 to 0

port output (mode 7)	
Port input (mode 7)	ON/OFF

Legend:

OFF: Input pull-up MOS is always off.

ON/OFF: On when PDDDR = 0 and PDPCR = 1; otherwise off.

PEDDR specifies input or output the port E pins using the individual bits. PEDDR cann
if it is, the read value is undefined. Since this register is a write-only register, bit-manipu
instructions should not be used when writing. See section 2.9.4, Access Method for Reg
Write-only Bits.

Bit	Bit Name	Initial Value	R/W	Description
7	PE7DDR	0	W	When a pin is specified as a general purpo port, setting this bit to 1 makes the corresp port E pin an output pin. Clearing this bit to the pin an input pin.
6	PE6DDR	0	W	
5	PE5DDR	0	W	
4	PE4DDR	0	W	
3	PE3DDR	0	W	
2	PE2DDR	0	W	
1	PE1DDR	0	W	
0	PE0DDR	0	W	

2	PE2DR	0	R/W	
1	PE1DR	0	R/W	
0	PE0DR	0	R/W	

9.12.3 Port E Register (PORTE)

PORTE shows port E pin states. This register cannot be modified.

Bit	Bit Name	Initial Value	R/W	Description
7	PE7	—*	R	If a port E read is performed while PEDDR b
6	PE6	—*	R	set to 1, the PEDR values are read. If a port
5	PE5	—*	R	is performed while PEDDR bits are cleared t
4	PE4	—*	R	pin states are read.
3	PE3	—*	R	
2	PE2	—*	R	
1	PE1	—*	R	
0	PE0	—*	R	

Note: * Determined by the states of pins PE7 to PE0.

2	PE2PCR	0	R/W
1	PE1PCR	0	R/W
0	PE0PCR	0	R/W

9.12.5 Pin Functions

Port E pins also function as data I/O pins. Port E pin functions are shown below.

- PE7/D7, PE6/D6, PE5/D5, PE4/D4, PE3/D3, PE2/D2, PE1/D1, PE0/D0
  The pin function is switched as shown below according to the combination of the op
  mode and the PEnDDR bit.

Operating mode	Mode 6			Mode 7	
Bus mode	8-bit bus mode		16-bit bus mode	—	
PEnDDR	0	1	—	0	
Pin function	PEn input	PEn output	Data input/output	PEn input	PI

Note: n = 7 to 0

bus in mode 6, mode 7)	
Port input (8-bit bus in mode 6, mode 7)	ON/OFF

Legend:

OFF: Input pull-up MOS is always off.

ON/OFF: On when PEDDR = 0 and PEPCR = 1; otherwise off.

PFDDR specifies input or output the port F pins using the individual bits. PFDDR canno
if it is, the read value is undefined. Since this register is a write-only register, bit-manipu
instructions should not be used when writing. See section 2.9.4, Access Method for Reg
Write-only Bits.

Bit	Bit Name	Initial Value	R/W	Description
7	PF7DDR	0/1*	W	When a pin is specified as a general purpo port, setting this bit to 1 makes the corresp port F pin an output pin. Clearing this bit to the pin an input pin.
6	PF6DDR	0	W	
5	PF5DDR	0	W	
4	PF4DDR	0	W	
3	PF3DDR	0	W	
2	PF2DDR	0	W	
1	PF1DDR	0	W	
0	PF0DDR	0	W	

Note: * PF7DDR is initialized to 1 in mode 6 and 0 in mode 7.

2	PF2DR	0	R/W	
1	PF1DR	0	R/W	
0	PF0DR	0	R/W	

Note: * The value of PF7DR is not output on pin PF7 when the PF7DDR bit is set to 1 ϕ signal is output.

9.13.3 Port F Register (PORTF)

PORTF shows port F pin states. This register cannot be modified.

Bit	Bit Name	Initial Value	R/W	Description
7	PF7	—*	R	If a port F read is performed while PFDDR b set to 1, the PFDR values are read. If a port performed while PFDDR bits are cleared to states are read.
6	PF6	—*	R	
5	PF5	—*	R	
4	PF4	—*	R	
3	PF3	—*	R	
2	PF2	—*	R	
1	PF1	—*	R	
0	PF0	—*	R	

Note: * Determined by the states of pins PF7 to PF0.

- PF6/$\overline{AS}$

  The pin function is switched as shown below according to the combination of the op
  mode and the PF6DDR bit.

Operating mode	Mode 6	Mode 7	
PF6DDR	—	0	1
Pin function	$\overline{AS}$ output	PF6 input	PF6 o

- PF5/$\overline{RD}$

  The pin function is switched as shown below according to the combination of the op
  mode and the PF5DDR bit.

Operating mode	Mode 6	Mode 7	
PF5DDR	—	0	1
Pin function	$\overline{RD}$ output	PF5 input	PF5 o

- PF4/$\overline{HWR}$

  The pin function is switched as shown below according to the combination of the op
  mode and the PF4DDR bit.

Operating mode	Mode 6	Mode 7	
PF4DDR	—	0	1
Pin function	$\overline{HWR}$ output	PF4 input	PF4 o

		$\overline{\text{IRQ3}}$ input*2

Notes: 1. When TRGS0 = TRGS1 = 1, port F is used as the $\overline{\text{ADTRG}}$ input pin.
2. When this port is used as an external interrupt pin, do not specify other functio

- PF2/$\overline{\text{WAIT}}$

  The pin function is switched as shown below according to the combination of the ope mode, the WAITE bit, and the PF2DDR bit.

Operating mode	Mode 6			Mode 7	
WAITE	0		1	—	
PF2DDR	0	1	—	0	
Pin function	PF2 input	PF2 output	$\overline{\text{WAIT}}$ input	PF2 input	PF

- PF1/$\overline{\text{BACK}}$/BUZZ

  The pin function is switched as shown below according to the combination of the ope mode, the BRLE bit, the BUZZ bit in PFCR, and the PF1DDR bit.

Operating mode	Mode 6				Mode 7	
BRLE	0			1	—	
BUZZE	0		1	—	0	
PF1DDR	0	1	—	—	0	1
Pin function	PF1 input	PF1 output	BUZZ output	$\overline{\text{BACK}}$ output	PF1 input	PF1 output

9.14 Port G

Port G is a 5-bit I/O port and has the following registers.

- Port G data direction register (PGDDR)
- Port G data register (PGDR)
- Port G register (PORTG)

9.14.1 Port G Data Direction Register (PGDDR)

PGDDR specifies input or output the port G pins using the individual bits. PGDDR cann read; if it is, the read value is undefined. Since this register is a write-only register, bit-manipulation instructions should not be used when writing. See section 2.9.4, Access M Registers with Write-only Bits.

Bit	Bit Name	Initial Value	R/W	Description
7 to 5	—	Undefined	—	Reserved These bits are always read as undefined v cannot be modified.
4	PG4DDR	0	W	When a pin is specified as a general purpo port, setting these bits to 1 makes the corre port G pin an output pin. Clearing this bit to the pin an input pin.
3	PG3DDR*	0	W	
2	PG2DDR*	0	W	
1	PG1DDR	0	W	
0	PG0DDR	0	W	

Note: * Reserved in the H8S/2556 Group. This bit is set to 0.

2	PG2DR*	0	R/W	
1	PG1DR	0	R/W	
0	PG0DR	0	R/W	

Note: * Reserved in the H8S/2556 Group. This bit is set to 0.

9.14.3 Port G Register (PORTG)

PORTG shows port G pin states. This register cannot be modified.

Bit	Bit Name	Initial Value	R/W	Description
7 to 5	—	Undefined	—	Reserved This bit is always read as undefined value a cannot be modified.
4	PG4	—*[1]	R	If these bits are read while the correspondin PGDDR bits are set to 1, the PGDR value is these bits are read while PGDDR bits are cl 0, the pin states are read.
3	PG3*[2]	—*[1]	R	
2	PG2*[2]	—*[1]	R	
1	PG1	—*[1]	R	
0	PG0	—*[1]	R	

Notes: 1. Determined by the states of pins PG4 to PG0.
2. Reserved in the H8S/2556 Group. An undefined value will be read.

Pin function	PG4 input	$\overline{CS0}$ output	PG4 input	PG4

- PG3/$\overline{Rx}$/$\overline{CS1}$

  In the H8S/2552 and H8S/2506 Groups, the pin function is switched as shown below according to the combination of the IEE bit in IECTR of IEB*, the operating mode, a PG3DDR bit. This pin is not available in the H8S/2556 Group.

IEE	0				
Operating mode	Mode 6		Mode 7		
PG3DDR	0	1	0	1	
Pin function	PG3 input	$\overline{CS1}$ output	PG3 input	PG3 output	

Note: * IEB is supported only by the H8S/2552 Group.

- PG2/$\overline{Tx}$/$\overline{CS2}$

  In the H8S/2552 and H8S/2506 Groups, the pin function is switched as shown below according to the combination of the IEE bit in IECTR of IEB*, the operating mode, a PG2DDR bit. This pin is not available in the H8S/2556 Group.

IEE	0				
Operating mode	Mode 6		Mode 7		
PG2DDR	0	1	0	1	
Pin function	PG2 input	$\overline{CS2}$ output	PG2 input	PG2 output	$\overline{T}$

Note: * IEB is supported only by the H8S/2552 Group.

- PG0/IRQ6

  The pin function is switched as shown below according to the PG0DDR bit.

PG0DDR	0	1
Pin function	PG0 input	PG0 output
	$\overline{\text{IRQ6}}$ input*	

Note: * When this port is used as an external interrupt pin, do not specify other functio

9.15 Port H

Port H is an 8-bit I/O port and has the following registers.

- Port H data direction register (PHDDR)
- Port H data register (PHDR)
- Port H register (PORTH)

5	PH5DDR	0	W	the pin an input pin.
4	PH4DDR	0	W	
3	PH3DDR	0	W	
2	PH2DDR	0	W	
1	PH1DDR	0	W	
0	PH0DDR	0	W	

9.15.2 Port H Data Register (PHDR)

PHDR stores output data for port H pins.

Bit	Bit Name	Initial Value	R/W	Description
7	PH7DR	0	R/W	Output data for a pin is stored when the pin specified as a general purpose output port.
6	PH6DR	0	R/W	
5	PH5DR	0	R/W	
4	PH4DR	0	R/W	
3	PH3DR	0	R/W	
2	PH2DR	0	R/W	
1	PH1DR	0	R/W	
0	PH0DR	0	R/W	

2	PH2	—*	R
1	PH1	—*	R
0	PH0	—*	R

Note: * Determined by the states of pins PH7 to PH0.

9.15.4 Pin Functions

Port H pins also function as general purpose I/O pins. Port H pin functions are shown bel

- PH7, PH6, PH5, PH4, PH3, PH2, PH1, PH0
  The pin function is switched as shown below according to the PHnDDR bit.

PHnDDR	0	1
Pin function	PHn input	PHn output

Note: n = 7 to 0

PJDDR specifies input or output the port J pins using the individual bits. PJDDR cannot it is, the read value is undefined. Since this register is a write-only register, bit-manipula instructions should not be used when writing. See section 2.9.4, Access Method for Reg Write-only Bits.

Bit	Bit Name	Initial Value	R/W	Description
7	PJ7DDR	0	W	When a pin is specified as a general purpo port, setting this bit to 1 makes the corresp port J pin an output pin. Clearing this bit to the pin an input pin.
6	PJ6DDR	0	W	
5	PJ5DDR	0	W	
4	PJ4DDR	0	W	
3	PJ3DDR	0	W	
2	PJ2DDR	0	W	
1	PJ1DDR	0	W	
0	PJ0DDR	0	W	

2	PJ2DR	0	R/W	
1	PJ1DR	0	R/W	
0	PJ0DR	0	R/W	

9.16.3 Port J Register (PORTJ)

PORTJ shows port J pin states. This register cannot be modified.

Bit	Bit Name	Initial Value	R/W	Description
7	PJ7	—*	R	If a port J read is performed while PJDDR bi
6	PJ6	—*	R	set to 1, the PJDR values are read. If a port
5	PJ5	—*	R	performed while PJDDR bits are cleared to 0
4	PJ4	—*	R	states are read.
3	PJ3	—*	R	
2	PJ2	—*	R	
1	PJ1	—*	R	
0	PJ0	—*	R	

Note: * Determined by the states of pins PJ7 to PJ0.

7 to 4		All 0		Reserved This bit is readable/writable, but the write val should always be 0.
3	BUFGC2	0	R/W	Buffer Gain Control 2 Controls drivability of output ports of which p supplied by $P2V_{CC}$. This bit should be set acc the voltage of $P2V_{CC}$ when a port is used as a port. If the bit setting is not appropriate, it ma malfunction or characteristics described in se Electrical Characteristics cannot be satisfied power supply pin, see table 1.1. 0: 4.5 V ≤ $P2V_{CC}$ ≤ 5.5 V 1: 3.0 V ≤ $P2V_{CC}$ ≤ 3.6 V
2	BUFGC1	0	R/W	Buffer Gain Control 1 Controls drivability of output ports of which p supplied by $P1V_{CC}$. This bit should be set acc the voltage of $P1V_{CC}$ when a port is used as a port. If the bit setting is not appropriate, it ma malfunction or characteristics described in se Electrical Characteristics cannot be satisfied power supply pin, see table 1.1. 0: 4.5 V ≤ $P1V_{CC}$ ≤ 5.5 V 1: 3.0 V ≤ $P1V_{CC}$ ≤ 3.6 V
1, 0	—	All 0	R/W	Reserved These bits are readable/writable, but the writ should always be 0.

Port 2	
Port 3	
Port 4	Connect each pin to AVcc (pull-up) or to AVss (pull-down) via a resistor.
Port 5	Connect each pin to P1Vcc (pull-up) or to Vss (pull-down) via a resistor.
Port 7	Connect each pin to P2Vcc (pull-up) or to Vss (pull-down) via a resistor.
Port 9	Connect each pin to AVcc (pull-up) or to AVss (pull-down) via a resistor.
Port A	Connect each pin to P1Vcc (pull-up) or to Vss (pull-down) via a resistor.
Port B	
Port C	
Port D	
Port E	
Port F	
Port G	
Port H	
Port J	

- Selection of 8 counter input clocks for each channel
- The following operations can be set for each channel:
  - Waveform output at compare match
  - Input capture function
  - Counter clear operation
  - Synchronous operation:
    Multiple timer counters (TCNT) can be written to simultaneously
    Simultaneous clearing by compare match and input capture is possible
    Register simultaneous input/output is possible by synchronous counter operation
  - A maximum 15-phase PWM output is possible in combination with synchronous
- Buffer operation settable for channels 0 and 3
- Phase counting mode settable independently for each of channels 1, 2, 4, and 5
- Cascaded operation
- Fast access via internal 16-bit bus
- 26 interrupt sources
- Automatic transfer of register data
- A/D converter conversion start trigger can be generated
- Module stop mode can be set

(TGR)		TGRB_0	TGRB_1	TGRB_2	TGRB_3	TGRB_4	TG
General registers/ buffer registers		TGRC_0 TGRD_0	—	—	TGRC_3 TGRD_3	—	—
I/O pins		TIOCA0 TIOCB0 TIOCC0 TIOCD0	TIOCA1 TIOCB1	TIOCA2 TIOCB2	TIOCA3 TIOCB3 TIOCC3 TIOCD3	TIOCA4 TIOCB4	TI TI
Counter clear function		TGR compare match or input capture	TGR compare match or input capture	TGR compare match or input capture	TGR compare match or input capture	TGR compare match or input capture	TG co ma inp ca
Compare match output	0 output	○	○	○	○	○	○
	1 output	○	○	○	○	○	○
	Toggle output	○	○	○	○	○	○
Input capture function		○	○	○	○	○	○
Synchronous operation		○	○	○	○	○	○
PWM mode		○	○	○	○	○	○
Phase counting mode		—	○	○	—	○	○
Buffer operation		○	—	—	○	—	—

Interrupt sources	8 sources	4 sources	4 sources	8 sources	4 sources	4 so
	• Compare match or input capture 0A	• Compare match or input capture 1A	• Compare match or input capture 2A	• Compare match or input capture 3A	• Compare match or input capture 4A	•
	• Compare match or input capture 0B	• Compare match or input capture 1B	• Compare match or input capture 2B	• Compare match or input capture 3B	• Compare match or input capture 4B	•
	• Compare match or input capture 0C	• Overflow • Underflow	• Overflow • Underflow	• Compare match or input capture 3C	• Overflow • Underflow	•
	• Compare match or input capture 0D			• Compare match or input capture 3D		
	• Overflow			• Overflow		

Legend:

○: Possible

—: Not possible

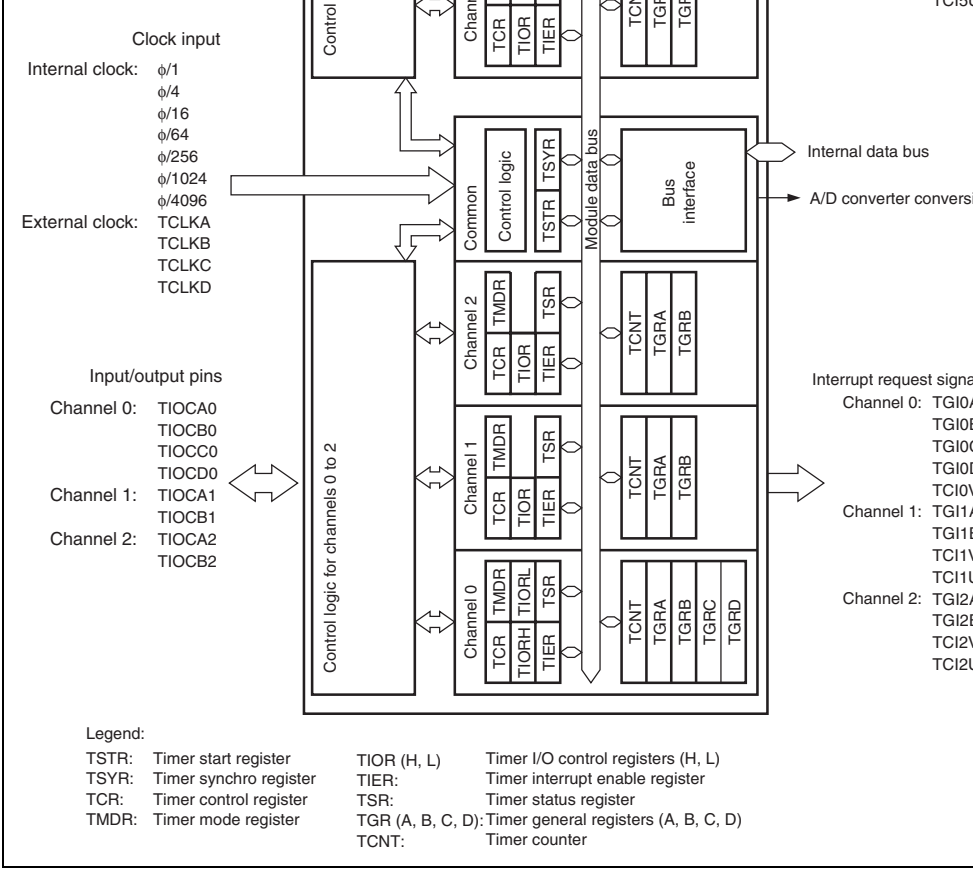


Figure 10.1 Block Diagram of TPU

	TCLKD	Input	External clock D input pin (Channel 2 and 4 phase counting mode B phase input)
0	TIOCA0	I/O	TGRA_0 input capture input/output compare output/PWM output pin
	TIOCB0	I/O	TGRB_0 input capture input/output compare output/PWM output pin
	TIOCC0	I/O	TGRC_0 input capture input/output compare output/PWM output pin
	TIOCD0	I/O	TGRD_0 input capture input/output compare output/PWM output pin
1	TIOCA1	I/O	TGRA_1 input capture input/output compare output/PWM output pin
	TIOCB1	I/O	TGRB_1 input capture input/output compare output/PWM output pin
2	TIOCA2	I/O	TGRA_2 input capture input/output compare output/PWM output pin
	TIOCB2	I/O	TGRB_2 input capture input/output compare output/PWM output pin
3	TIOCA3	I/O	TGRA_3 input capture input/output compare output/PWM output pin
	TIOCB3	I/O	TGRB_3 input capture input/output compare output/PWM output pin
	TIOCC3	I/O	TGRC_3 input capture input/output compare output/PWM output pin
	TIOCD3	I/O	TGRD_3 input capture input/output compare output/PWM output pin

- Timer mode register_0 (TMDR_0)
- Timer I/O control register H_0 (TIORH_0)
- Timer I/O control register L_0 (TIORL_0)
- Timer interrupt enable register_0 (TIER_0)
- Timer status register_0 (TSR_0)
- Timer counter_0 (TCNT_0)
- Timer general register A_0 (TGRA_0)
- Timer general register B_0 (TGRB_0)
- Timer general register C_0 (TGRC_0)
- Timer general register D_0 (TGRD_0)

Channel 1

- Timer control register_1 (TCR_1)
- Timer mode register_1 (TMDR_1)
- Timer I/O control register _1 (TIOR_1)
- Timer interrupt enable register_1 (TIER_1)
- Timer status register_1 (TSR_1)
- Timer counter_1 (TCNT_1)
- Timer general register A_1 (TGRA_1)
- Timer general register B_1 (TGRB_1)

Channel 3

- Timer control register_3 (TCR_3)
- Timer mode register_3 (TMDR_3)
- Timer I/O control register H_3 (TIORH_3)
- Timer I/O control register L_3 (TIORL_3)
- Timer interrupt enable register_3 (TIER_3)
- Timer status register_3 (TSR_3)
- Timer counter_3 (TCNT_3)
- Timer general register A_3 (TGRA_3)
- Timer general register B_3 (TGRB_3)
- Timer general register C_3 (TGRC_3)
- Timer general register D_3 (TGRD_3)

Channel 4

- Timer control register_4 (TCR_4)
- Timer mode register_4 (TMDR_4)
- Timer I/O control register _4 (TIOR_4)
- Timer interrupt enable register_4 (TIER_4)
- Timer status register_4 (TSR_4)
- Timer counter_4 (TCNT_4)
- Timer general register A_4 (TGRA_4)
- Timer general register B_4 (TGRB_4)

Common Registers

- Timer start register (TSTR)
- Timer synchro register (TSYR)

5	CCLR0	0	R/W	
4	CKEG1	0	R/W	Clock Edge 1 and 0
3	CKEG0	0	R/W	These bits select the input clock edge. When the clock is counted using both edges, the input clock halved (e.g. ϕ/4 both edges = ϕ/2 rising edge). If p counting mode is used on channels 1, 2, 4, and 5 setting is ignored and the phase counting mode s has priority. Internal clock edge selection is valid w input clock is ϕ/4 or slower. This setting is ignored input clock is ϕ/1, or when overflow/underflow of a channel is selected. (The clock is counted at the f edge when ϕ/1 is selected.) 00: Count at rising edge 01: Count at falling edge 1X: Count at both edges Legend: X: Don't care
2	TPSC2	0	R/W	Time Prescaler 2 to 0
1	TPSC1	0	R/W	These bits select the TCNT counter clock. The clo
0	TPSC0	0	R/W	source can be selected independently for each ch See tables 10.5 to 10.10 for details.

				synchronous operation*1
	1	0	0	TCNT clearing disabled
			1	TCNT cleared by TGRC compare mat capture*2
		1	0	TCNT cleared by TGRD compare mat capture*2
			1	TCNT cleared by counter clearing for channel performing synchronous clea synchronous operation*1

Notes: 1. Synchronous operation is set by setting the SYNC bit in TSYR to 1.
2. When TGRC or TGRD is used as a buffer register, TCNT is not cleared beca buffer register setting has priority, and compare match/input capture does not

Table 10.4 CCLR0 to CCLR2 (channels 1, 2, 4, and 5)

Channel	Bit 7 Reserved*2	Bit 6 CCLR1	Bit 5 CCLR0	Description
1, 2, 4, 5	0	0	0	TCNT clearing disabled
			1	TCNT cleared by TGRA compare mat capture
		1	0	TCNT cleared by TGRB compare mat capture
			1	TCNT cleared by counter clearing for channel performing synchronous clea synchronous operation*1

Notes: 1. Synchronous operation is selected by setting the SYNC bit in TSYR to 1.
2. Bit 7 is reserved in channels 1, 2, 4, and 5. It is always read as 0 and cannot modified.

1	0	0	External clock: counts on TCLKC pin in
		1	External clock: counts on TCLKD pin in

Table 10.6 TPSC0 to TPSC2 (channel 1)

Channel	Bit 2 TPSC2	Bit 1 TPSC1	Bit 0 TPSC0	Description
1	0	0	0	Internal clock: counts on ϕ/1
			1	Internal clock: counts on ϕ/4
		1	0	Internal clock: counts on ϕ/16
			1	Internal clock: counts on ϕ/64
	1	0	0	External clock: counts on TCLKA pin in
			1	External clock: counts on TCLKB pin in
		1	0	Internal clock: counts on ϕ/256
			1	Counts on TCNT2 overflow/underflow

Note: This setting is ignored when channel 1 is in phase counting mode.

		1	0	External clock: counts on TCLKC pin i
			1	Internal clock: counts on ϕ/1024

Note: This setting is ignored when channel 2 is in phase counting mode.

Table 10.8 TPSC0 to TPSC2 (channel 3)

Channel	Bit 2 TPSC2	Bit 1 TPSC1	Bit 0 TPSC0	Description
3	0	0	0	Internal clock: counts on ϕ/1
			1	Internal clock: counts on ϕ/4
		1	0	Internal clock: counts on ϕ/16
			1	Internal clock: counts on ϕ/64
	1	0	0	External clock: counts on TCLKA pin i
			1	Internal clock: counts on ϕ/1024
		1	0	Internal clock: counts on ϕ/256
			1	Internal clock: counts on ϕ/4096

		1	0	Internal clock: counts on ϕ/1024
			1	Counts on TCNT5 overflow/underflow

Note: This setting is ignored when channel 4 is in phase counting mode.

Table 10.10 TPSC0 to TPSC2 (channel 5)

Channel	Bit 2 TPSC2	Bit 1 TPSC1	Bit 0 TPSC0	Description
5	0	0	0	Internal clock: counts on ϕ/1
			1	Internal clock: counts on ϕ/4
		1	0	Internal clock: counts on ϕ/16
			1	Internal clock: counts on ϕ/64
	1	0	0	External clock: counts on TCLKA pin in
			1	External clock: counts on TCLKC pin in
		1	0	Internal clock: counts on ϕ/256
			1	External clock: counts on TCLKD pin in

Note: This setting is ignored when channel 5 is in phase counting mode.

5	BFB	0	R/W	Buffer Operation B Specifies whether TGRB is to operate in the norm or TGRB and TGRD are to be used together for b operation. When TGRD is used as a buffer regist input capture/output compare is not generated. In channels 1, 2, 4, and 5, which have no TGRD, reserved. It is always read as 0 and cannot be m 0: TGRB operates normally 1: TGRB and TGRD are used together for buffer
4	BFA	0	R/W	Buffer Operation A Specifies whether TGRA is to operate in the norm or TGRA and TGRC are to be used together for b operation. When TGRC is used as a buffer regist input capture/output compare is not generated. In channels 1, 2, 4, and 5, which have no TGRC, reserved. It is always read as 0 and cannot be m 0: TGRA operates normally 1:TGRA and TGRC are used together for buffer o
3	MD3	0	R/W	Modes 3 to 0
2	MD2	0	R/W	These bits are used to set the timer operating mo
1	MD1	0	R/W	MD3 is a reserved bit. In a write, it should always
0	MD0	0	R/W	written with 0. See table 10.11 for details.

		1	0	Phase counting mode 3
			1	Phase counting mode 4
1	X	X	X	—

Legend:

X: Don't care

Notes: 1. MD3 is a reserved bit. In a write, it should always be written with 0.

2. Phase counting mode cannot be set for channels 0 and 3. In this case, 0 shoul be written to MD2.

10.3.3 Timer I/O Control Register (TIOR)

The TIOR registers control the TGR registers. The TPU has eight TIOR registers, two ea channels 0 and 3, and one each for channels 1, 2, 4, and 5. Care is required as TIOR is aff the TMDR setting.

The initial output specified by TIOR is valid when the counter is stopped (the CST bit in cleared to 0). Note also that, in PWM mode 2, the output at the point at which the counter cleared to 0 is specified.

When TGRC or TGRD is designated for buffer operation, this setting is invalid and the re operates as a buffer register.

0	IOA0	0	R/W	

- TIORL_0, TIORL_3

Bit	Bit Name	Initial value	R/W	Description
7	IOD3	0	R/W	I/O Control D3 to D0
6	IOD2	0	R/W	Specify the function of TGRD. See tables 10.13, for details.
5	IOD1	0	R/W	
4	IOD0	0	R/W	
3	IOC3	0	R/W	I/O Control C3 to C0
2	IOC2	0	R/W	Specify the function of TGRC. See tables 10.21, for details.
1	IOC1	0	R/W	
0	IOC0	0	R/W	

					Toggle output at compare match
	1	0	0		Output disabled
			1		Initial output is 1 0 output at compare match
		1	0		Initial output is 1 1 output at compare match
			1		Initial output is 1 Toggle output at compare match
1	0	0	0	Input capture register	Capture input source is TIOCB0 pin Input capture at rising edge
			1		Capture input source is TIOCB0 pin Input capture at falling edge
		1	X		Capture input source is TIOCB0 pin Input capture at both edges
	1	X	X		Capture input source is channel 1/count Input capture at TCNT_1 count-up/count

Legend:

X: Don't care

Note: * When the TPSC0 to TPSC2 bits in TCR_1 are set to B'000 and ϕ/1 is used as TCNT_1 count clock, this setting is invalid and input capture is not generated.

					Toggle output at compare match
	1	0	0		Output disabled
			1		Initial output is 1 0 output at compare match
		1	0		Initial output is 1 1 output at compare match
			1		Initial output is 1 Toggle output at compare match
1	0	0	0	Input capture register*2	Capture input source is TIOCD0 pin Input capture at rising edge
			1		Capture input source is TIOCD0 pin Input capture at falling edge
		1	X		Capture input source is TIOCD0 pin Input capture at both edges
	1	X	X		Capture input source is channel 1/coun Input capture at TCNT_1 count-up/cou

Legend:

X: Don't care

Notes: 1. When the TPSC0 to TPSC2 bits in TCR_1 are set to B'000 and ϕ/1 is used as TCNT_1 count clock, this setting is invalid and input capture is not generated

2. When the BFB bit in TMDR_0 is set to 1 and TGRD_0 is used as a buffer reg setting is invalid and input capture/output compare is not generated.

					Toggle output at compare match
	1	0	0		Output disabled
			1		Initial output is 1 0 output at compare match
		1	0		Initial output is 1 1 output at compare match
			1		Initial output is 1 Toggle output at compare match
1	0	0	0	Input capture register	Capture input source is TIOCB1 pin Input capture at rising edge
			1		Capture input source is TIOCB1 pin Input capture at falling edge
		1	X		Capture input source is TIOCB1 pin Input capture at both edges
	1	X	X		TGRC_0 compare match/ input capture Input capture at generation of TGRC_0 c match/input capture

Legend:

X: Don’t care

					Toggle output at compare match
	1	0	0		Output disabled
			1		Initial output is 1 0 output at compare match
		1	0		Initial output is 1 1 output at compare match
			1		Initial output is 1 Toggle output at compare match
1	X	0	0	Input capture register	Capture input source is TIOCB2 pin Input capture at rising edge
			1		Capture input source is TIOCB2 pin Input capture at falling edge
		1	X		Capture input source is TIOCB2 pin Input capture at both edges

Legend:

X: Don't care

					Toggle output at compare match
	1	0	0		Output disabled
			1		Initial output is 1 0 output at compare match
		1	0		Initial output is 1 1 output at compare match
			1		Initial output is 1 Toggle output at compare match
1	0	0	0	Input capture register	Capture input source is TIOCB3 pin Input capture at rising edge
			1		Capture input source is TIOCB3 pin Input capture at falling edge
		1	X		Capture input source is TIOCB3 pin Input capture at both edges
	1	X	X		Capture input source is channel 4/count c Input capture at TCNT_4 count-up/count-

Legend:

X: Don't care

Note: * When the TPSC0 to TPSC2 bits in TCR_4 are set to B'000 and ϕ/1 is used as TCNT_4 count clock, this setting is invalid and input capture is not generated.

					Toggle output at compare match
	1	0	0		Output disabled
			1		Initial output is 1 0 output at compare match
		1	0		Initial output is 1 1 output at compare match
			1		Initial output is 1 Toggle output at compare match
1	0	0	0	Input capture register*2	Capture input source is TIOCD3 pin Input capture at rising edge
			1		Capture input source is TIOCD3 pin Input capture at falling edge
		1	X		Capture input source is TIOCD3 pin Input capture at both edges
	1	X	X		Capture input source is channel 4/count Input capture at TCNT_4 count-up/coun

Legend:

X: Don't care

Notes: 1. When the TPSC0 to TPSC2 bits in TCR_4 are set to B'000 and ϕ/1 is used as TCNT_4 count clock, this setting is invalid and input capture is not generated

2. When the BFB bit in TMDR_3 is set to 1 and TGRD_3 is used as a buffer reg setting is invalid and input capture/output compare is not generated.

					Toggle output at compare match
	1	0	0		Output disabled
			1		Initial output is 1 0 output at compare match
		1	0		Initial output is 1 1 output at compare match
			1		Initial output is 1 Toggle output at compare match
1	0	0	0	Input capture register	Capture input source is TIOCB4 pin Input capture at rising edge
			1		Capture input source is TIOCB4 pin Input capture at falling edge
		1	X		Capture input source is TIOCB4 pin Input capture at both edges
	1	X	X		Capture input source is TGRC_3 compar match/input capture Input capture at generation of TGRC_3 c match/input capture

Legend:

X: Don’t care

					Toggle output at compare match
	1	0	0		Output disabled
			1		Initial output is 1 0 output at compare match
		1	0		Initial output is 1 1 output at compare match
			1		Initial output is 1 Toggle output at compare match
1	X	0	0	Input capture register	Capture input source is TIOCB5 pin Input capture at rising edge
			1		Capture input source is TIOCB5 pin Input capture at falling edge
		1	X		Capture input source is TIOCB5 pin Input capture at both edges

Legend:

X: Don’t care

					Toggle output at compare match
	1	0	0		Output disabled
			1		Initial output is 1 0 output at compare match
		1	0		Initial output is 1 1 output at compare match
			1		Initial output is 1 Toggle output at compare match
1	0	0	0	Input capture register	Capture input source is TIOCA0 pin Input capture at rising edge
			1		Capture input source is TIOCA0 pin Input capture at falling edge
		1	X		Capture input source is TIOCA0 pin Input capture at both edges
	1	X	X		Capture input source is channel 1/count c Input capture at TCNT_1 count-up/count-

Legend:

X: Don't care

					Toggle output at compare match
	1	0	0		Output disabled
			1		Initial output is 1 0 output at compare match
		1	0		Initial output is 1 1 output at compare match
			1		Initial output is 1 Toggle output at compare match
1	0	0	0	Input capture register*	Capture input source is TIOCC0 pin Input capture at rising edge
			1		Capture input source is TIOCC0 pin Input capture at falling edge
		1	X		Capture input source is TIOCC0 pin Input capture at both edges
	1	X	X		Capture input source is channel 1/count Input capture at TCNT_1 count-up/coun

Legend:

X: Don't care

Note: * When the BFA bit in TMDR_0 is set to 1 and TGRC_0 is used as a buffer reg setting is invalid and input capture/output compare is not generated.

					Toggle output at compare match
	1	0	0		Output disabled
			1		Initial output is 1 0 output at compare match
		1	0		Initial output is 1 1 output at compare match
			1		Initial output is 1 Toggle output at compare match
1	0	0	0	Input capture register	Capture input source is TIOCA1 pin Input capture at rising edge
			1		Capture input source is TIOCA1 pin Input capture at falling edge
		1	X		Capture input source is TIOCA1 pin Input capture at both edges
	1	X	X		Capture input source is TGRA_0 compar match/input capture Input capture at generation of channel 0/
					compare match/input capture

Legend:

X: Don’t care

					Toggle output at compare match
	1	0	0		Output disabled
			1		Initial output is 1 0 output at compare match
		1	0		Initial output is 1 1 output at compare match
			1		Initial output is 1 Toggle output at compare match
1	X	0	0	Input capture register	Capture input source is TIOCA2 pin Input capture at rising edge
			1		Capture input source is TIOCA2 pin Input capture at falling edge
		1	X		Capture input source is TIOCA2 pin Input capture at both edges

Legend:

X: Don't care

					Toggle output at compare match
	1	0	0		Output disabled
			1		Initial output is 1 0 output at compare match
		1	0		Initial output is 1 1 output at compare match
			1		Initial output is 1 Toggle output at compare match
1	0	0	0	Input capture register	Capture input source is TIOCA3 pin Input capture at rising edge
			1		Capture input source is TIOCA3 pin Input capture at falling edge
		1	X		Capture input source is TIOCA3 pin Input capture at both edges
	1	X	X		Capture input source is channel 4/count c Input capture at TCNT_4 count-up/count-

Legend:

X: Don't care

					Toggle output at compare match
	1	0	0		Output disabled
			1		Initial output is 1 0 output at compare match
		1	0		Initial output is 1 1 output at compare match
			1		Initial output is 1 Toggle output at compare match
1	0	0	0	Input capture register*	Capture input source is TIOCC3 pin Input capture at rising edge
			1		Capture input source is TIOCC3 pin Input capture at falling edge
		1	X		Capture input source is TIOCC3 pin Input capture at both edges
	1	X	X		Capture input source is channel 4/count Input capture at TCNT_4 count-up/coun

Legend:

X: Don't care

Note: * When the BFA bit in TMDR_3 is set to 1 and TGRC_3 is used as a buffer reg setting is invalid and input capture/output compare is not generated.

					Toggle output at compare match
	1	0	0		Output disabled
			1		Initial output is 1 0 output at compare match
		1	0		Initial output is 1 1 output at compare match
			1		Initial output is 1 Toggle output at compare match
1	0	0	0	Input capture register	Capture input source is TIOCA4 pin Input capture at rising edge
			1		Capture input source is TIOCA4 pin Input capture at falling edge
		1	X		Capture input source is TIOCA4 pin Input capture at both edges
	1	X	X		Capture input source is TGRA_3 compar match/input capture Input capture at generation of TGRA_3 c match/input capture

Legend:

X: Don't care

					Toggle output at compare match
	1	0	0		Output disabled
			1		Initial output is 1 0 output at compare match
		1	0		Initial output is 1 1 output at compare match
			1		Initial output is 1 Toggle output at compare match
1	X	0	0	Input capture register	Capture input source is TIOCA5 pin Input capture at rising edge
			1		Capture input source is TIOCA5 pin Input capture at falling edge
		1	X		Capture input source is TIOCA5 pin Input capture at both edges

Legend:

X: Don't care

				1: A/D conversion start request generation enable
6	—	1	—	Reserved This bit is always read as 1 and cannot be modifie
5	TCIEU	0	R/W	Underflow Interrupt Enable Enables or disables interrupt requests (TCIU) by t flag when the TCFU flag in TSR is set to 1 in chan 4, and 5. In channels 0 and 3, bit 5 is reserved. It is always and cannot be modified. 0: Interrupt requests (TCIU) by TCFU disabled 1: Interrupt requests (TCIU) by TCFU enabled
4	TCIEV	0	R/W	Overflow Interrupt Enable Enables or disables interrupt requests (TCIV) by t flag when the TCFV flag in TSR is set to 1. 0: Interrupt requests (TCIV) by TCFV disabled 1: Interrupt requests (TCIV) by TCFV enabled
3	TGIED	0	R/W	TGR Interrupt Enable D Enables or disables interrupt requests (TGID) by t bit when the TGFD bit in TSR is set to 1 in channe 3. In channels 1, 2, 4, and 5, bit 3 is reserved. It is al read as 0 and cannot be modified. 0: Interrupt requests (TGID) by TGFD bit disabled 1: Interrupt requests (TGID) by TGFD bit enabled

				Enables or disables interrupt requests (TGIB) by bit when the TGFB bit in TSR is set to 1. 0: Interrupt requests (TGIB) by TGFB bit disabled 1: Interrupt requests (TGIB) by TGFB bit enabled
0	TGIEA	0	R/W	TGR Interrupt Enable A Enables or disables interrupt requests (TGIA) by bit when the TGFA bit in TSR is set to 1. 0: Interrupt requests (TGIA) by TGFA bit disabled 1: Interrupt requests (TGIA) by TGFA bit enabled

				and cannot be modified. 0: TCNT counts down 1: TCNT counts up
6	—	1	—	Reserved This bit is always read as 1 and cannot be modifie
5	TCFU	0	R/(W)*	Underflow Flag Status flag that indicates that TCNT underflow has occurred when channels 1, 2, 4, and 5 are set to p counting mode. Only 0 can be written, for flag clea In channels 0 and 3, bit 5 is reserved. It is always and cannot be modified. [Setting condition] When the TCNT value underflows (changes from H'FFFF) [Clearing condition] When 0 is written to TCFU after reading TCFU = 1
4	TCFV	0	R/(W)*	Overflow Flag Status flag that indicates that TCNT overflow has Only 0 can be written, for flag clearing. [Setting condition] When the TCNT value overflows (changes from H H'0000) [Clearing condition] When 0 is written to TCFV after reading TCFV = 1

				capture signal and TGRD is functioning as inp capture register [Clearing conditions] • When DTC is activated by TGID interrupt, the of MRB in DTC is 0 with the transfer counter 0 • When 0 is written to TGFD after reading TGF
2	TGFC	0	R/(W)*	Input Capture/Output Compare Flag C Status flag that indicates the occurrence of TGR capture or compare match in channels 0 and 3. C be written, for flag clearing. In channels 1, 2, 4, a is reserved. It is always read as 0 and cannot be [Setting conditions] • When TCNT = TGRC and TGRC is functionin output compare register • When TCNT value is transferred to TGRC by capture signal and TGRC is functioning as inp capture register [Clearing conditions] • When DTC is activated by TGIC interrupt, the of MRB in DTC is 0 with the transfer counter 0 • When 0 is written to TGFC after reading TGF

				capture register [Clearing conditions] • When DTC is activated by TGIB interrupt, the I of MRB in DTC is 0 with the transfer counter o 0 • When 0 is written to TGFB after reading TGFB
0	TGFA	0	R/(W)*	Input Capture/Output Compare Flag A Status flag that indicates the occurrence of TGRA capture or compare match. Only 0 can be written, clearing. [Setting conditions] • When TCNT = TGRA and TGRA is functioning output compare register • When TCNT value is transferred to TGRA by i capture signal and TGRA is functioning as inpu capture register [Clearing conditions] • When DTC is activated by TGIA interrupt, the I of MRB in DTC is 0 with the transfer counter o 0 • When 0 is written to TGFA after reading TGFA

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

The TGR registers are dual function 16-bit readable/writable registers, functioning as eit
compare or input capture registers. The TPU has 16 TGR registers, four each for channe
and two each for channels 1, 2, 4, and 5. TGRC and TGRD for channels 0 and 3 can also
designated for operation as buffer registers. The TGR registers cannot be accessed in 8-b
they must always be accessed as a 16-bit unit. TGR buffer register combinations are TG
TGRC and TGRB—TGRD.

10.3.8 Timer Start Register (TSTR)

TSTR specifies whether to operate or stop TCNT for channels 0 to 5. When setting the c
mode in TMDR or setting the count clock in TCR, first stop the TCNT counter.

Bit	Bit Name	Initial value	R/W	Description
7, 6	—	All 0	—	Reserved Only 0 should be written to these bits.
5	CST5	0	R/W	Counter Start 5 to 0
4	CST4	0	R/W	These bits specify whether to operate or stop TC
3	CST3	0	R/W	If 0 is written to the CST bit during operation with
2	CST2	0	R/W	pin designated for output, the counter stops but t
1	CST1	0	R/W	pin output compare output level is retained. If TIO
0	CST0	0	R/W	written to when the CST bit is cleared to 0, the pi
				level will be changed to the set initial output value
				0: TCNT_0 to TCNT_5 count operation is stoppe
				1: TCNT_0 to TCNT_5 performs count operation

4	SYNC4	0	R/W	These bits are used to select the independent or synchronized operation with other channels. When synchronous operation is selected, the TCN synchronous presetting of multiple channels, and synchronous clearing by counter clearing on anoth channel, are possible. To set synchronous operation, the SYNC bits for a two channels must be set to 1. To set synchronous clearing, in addition to the SYNC bit , the TCNT cl source must also be set by means of the CCLR0 t bits in TCR. 0: TCNT_0 to TCNT_5 operates independently (T presetting /clearing is unrelated to other channe 1: TCNT_0 to TCNT_5 performs synchronous ope TCNT synchronous presetting/synchronous clea possible.
3	SYNC3	0	R/W	
2	SYNC2	0	R/W	
1	SYNC1	0	R/W	
0	SYNC0	0	R/W	

periodic counter, for example.

1. Example of count operation setting procedure

   Figure 10.2 shows an example of the count operation setting procedure.

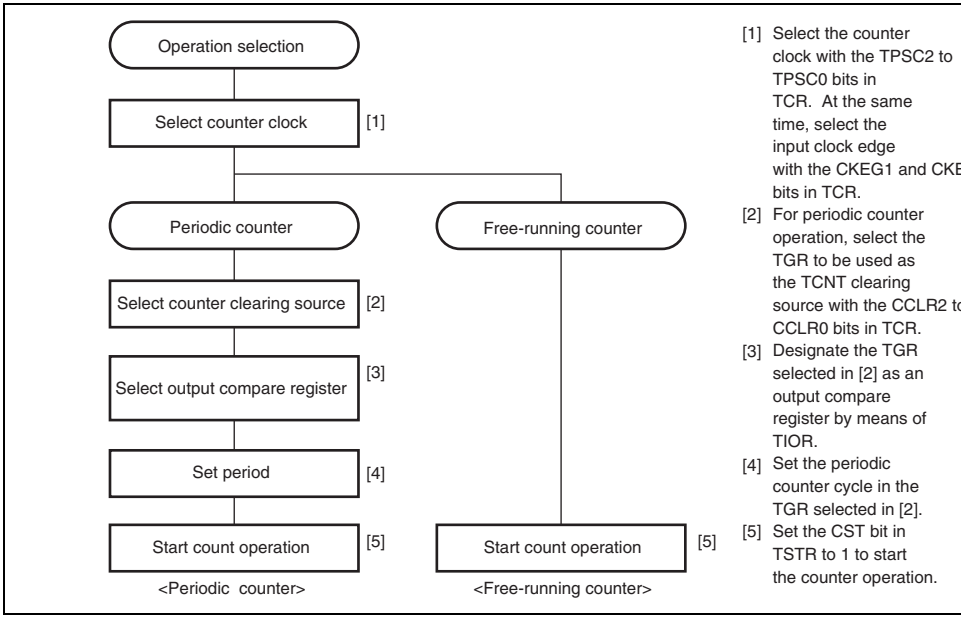


Figure 10.2 Example of Counter Operation Setting Procedure

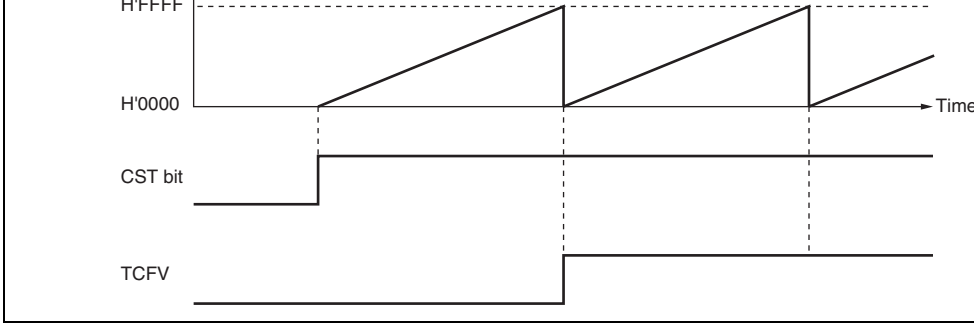


Figure 10.3 Free-Running Counter Operation

When compare match is selected as the TCNT clearing source, the TCNT counter for relevant channel performs periodic count operation. The TGR register for setting the p designated as an output compare register, and counter clearing by compare match is s by means of the CCLR0 to CCLR2 bits in TCR. After the settings have been made, T starts up-count operation as a periodic counter when the corresponding bit in TSTR is When the count value matches the value in TGR, the TGF bit in TSR is set to 1 and T cleared to H'0000.

If the value of the corresponding TGIE bit in TIER is 1 at this point, the TPU requests interrupt. After a compare match, TCNT starts counting up again from H'0000.

Figure 10.4 illustrates periodic counter operation.

Figure 10.4 Periodic Counter Operation

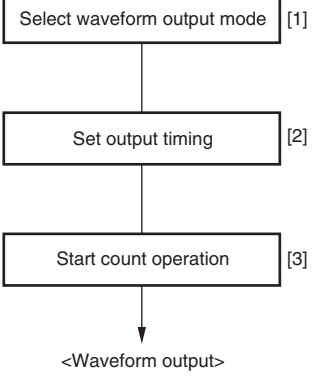


Figure 10.5 Example of Setting Procedure for Waveform Output by Compare M

2. Examples of waveform output operation

   Figure 10.6 shows an example of 0 output/1 output.

   In this example TCNT has been designated as a free-running counter, and settings hav made such that 1 is output by compare match A, and 0 is output by compare match B. the set level and the pin level coincide, the pin level does not change.

Figure 10.7 shows an example of toggle output.

In this example, TCNT has been designated as a periodic counter (with counter clear
compare match B), and settings have been made such that the output is toggled by bo
compare match A and compare match B.

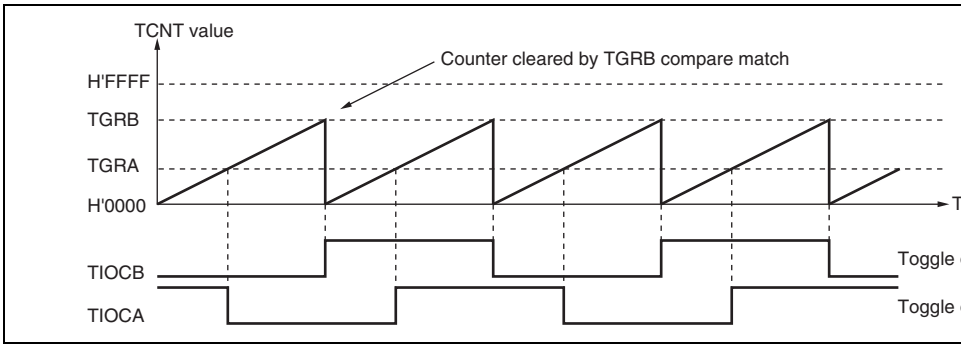


Figure 10.7 Example of Toggle Output Operation

Input Capture Function: The TCNT value can be transferred to TGR on detection of t
pin input edge.

Rising edge, falling edge, or both edges can be selected as the detected edge. For channe
and 4, it is also possible to specify another channel's counter input clock or compare mat
as the input capture source.

Note: When another channel's counter input clock is used as the input capture input fo
0 and 3, ϕ/1 should not be selected as the counter input clock used for input capt
Input capture will not be generated if ϕ/1 is selected.

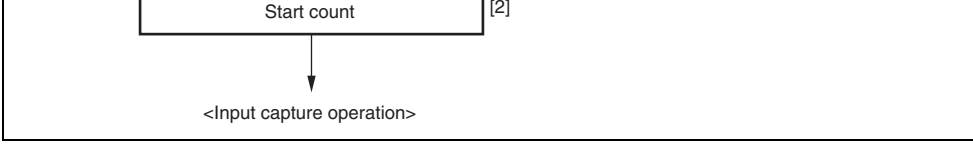


Figure 10.8 Example of Input Capture Operation Setting Procedure

2. Example of input capture operation

   Figure 10.9 shows an example of input capture operation.

   In this example both rising and falling edges have been selected as the TIOCA pin inp
   capture input edge, the falling edge has been selected as the TIOCB pin input capture
   edge, and counter clearing by TGRB input capture has been designated for TCNT.

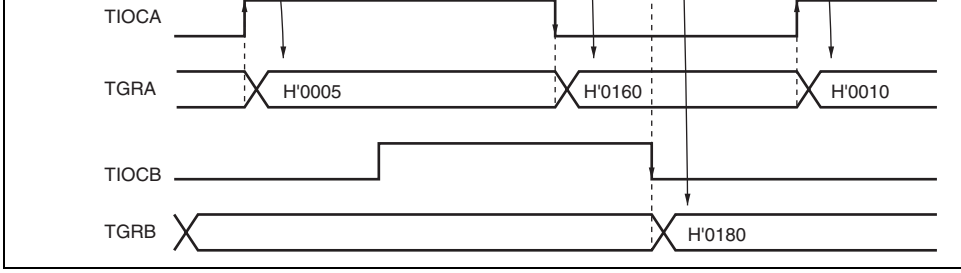


Figure 10.9 Example of Input Capture Operation

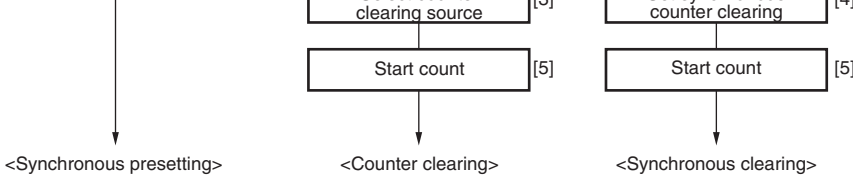
10.4.2 Synchronous Operation

In synchronous operation, the values in a number of TCNT counters can be rewritten simultaneously (synchronous presetting). Also, a number of TCNT counters can be clea simultaneously by making the appropriate setting in TCR (synchronous clearing).

Synchronous operation enables TGR to be incremented with respect to a single time bas

Channels 0 to 5 can all be designated for synchronous operation.

Example of Synchronous Operation Setting Procedure: Figure 10.10 shows an exam synchronous operation setting procedure.



[1] Set to 1 the SYNC bits in TSYR corresponding to the channels to be designated for synchronous operation.
[2] When the TCNT counter of any of the channels designated for synchronous operation is written to, the same value is simultaneously written to the other TCNT counters.
[3] Use the CCLR2 to CCLR0 bits in TCR to specify TCNT clearing by input capture/output compare, etc.
[4] Use the CCLR2 to CCLR0 bits in TCR to designate synchronous clearing for the counter clearing source.
[5] Set to 1 the CST bits in TSTR for the relevant channels, to start the count operation.

Figure 10.10 Example of Synchronous Operation Setting Procedure

Example of Synchronous Operation: Figure 10.11 shows an example of synchronous o

In this example, synchronous operation and PWM mode 1 have been designated for chan 2, TGRB_0 compare match has been set as the channel 0 counter clearing source, and synchronous clearing has been set for the channel 1 and 2 counter clearing source.

Three-phase PWM waveforms are output from pins TIOCA0, TIOCA1, and TIOCA2. At time, synchronous presetting, and synchronous clearing by TGRB_0 compare match, are performed for channel 0 to 2 TCNT counters, and the data set in TGRB_0 is used as the P cycle.

For details of PWM modes, see section 10.4.5, PWM Modes.

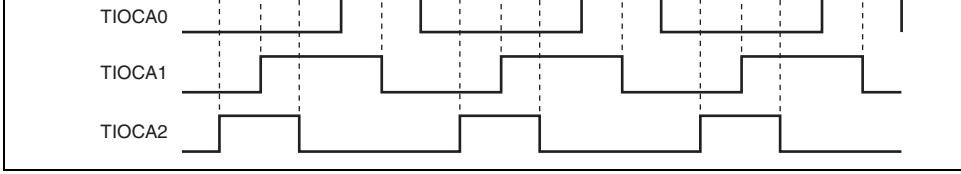


Figure 10.11 Example of Synchronous Operation

10.4.3 Buffer Operation

Buffer operation, provided for channels 0 and 3, enables TGRC and TGRD to be used a registers.

Buffer operation differs depending on whether TGR has been designated as an input cap register or as a compare match register.

Table 10.28 shows the register combinations used in buffer operation.

Table 10.28 Register Combinations in Buffer Operation

Channel	Timer General Register	Buffer Register
0	TGRA_0	TGRC_0
	TGRB_0	TGRD_0
3	TGRA_3	TGRC_3
	TGRB_3	TGRD_3

- When TGR is an output compare register

  When a compare match occurs, the value in the buffer register for the corresponding transferred to the timer general register.

When input capture occurs, the value in TCNT is transferred to TGR and the value pr
held in the timer general register is transferred to the buffer register.

This operation is illustrated in figure 10.13.

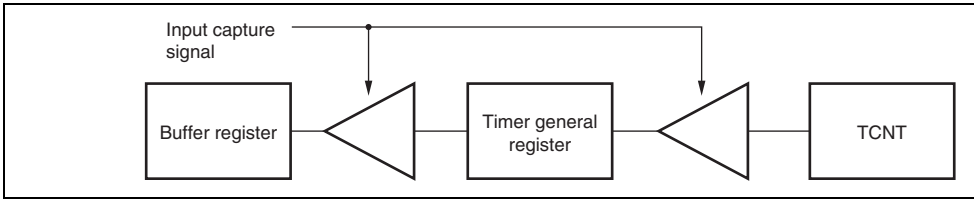


Figure 10.13 Input Capture Buffer Operation

Example of Buffer Operation Setting Procedure: Figure 10.14 shows an example of th
operation setting procedure.

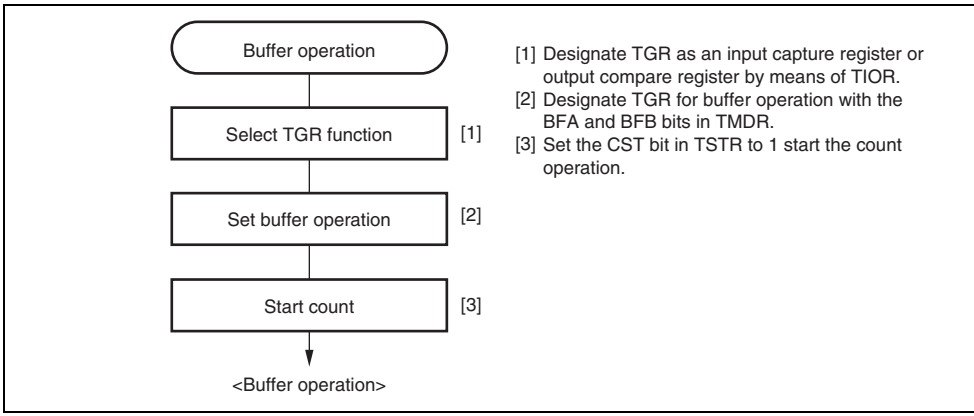


Figure 10.14 Example of Buffer Operation Setting Procedure

For details of PWM modes, see section 10.4.5, PWM Modes.

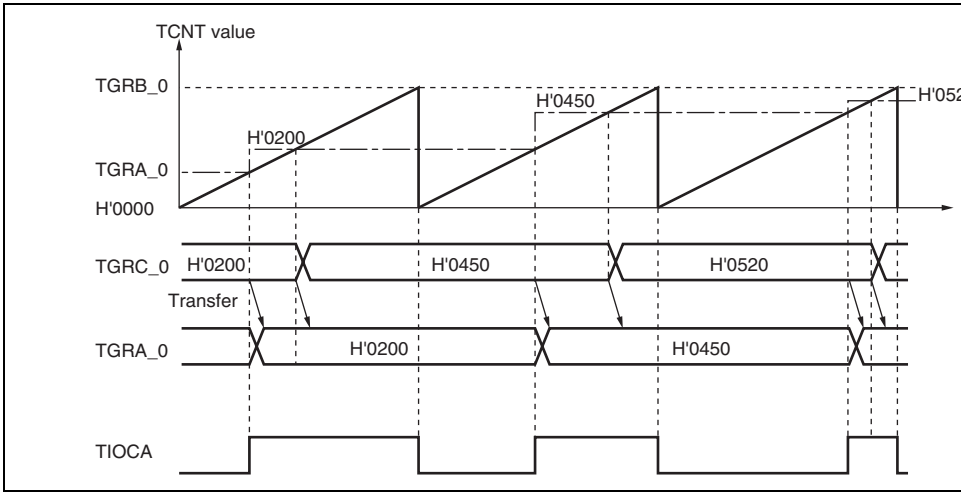


Figure 10.15 Example of Buffer Operation (1)

2. When TGR is an input capture register

   Figure 10.16 shows an operation example in which TGRA has been designated as an capture register, and buffer operation has been designated for TGRA and TGRC.

   Counter clearing by TGRA input capture has been set for TCNT, and both rising and edges have been selected as the TIOCA pin input capture input edge.

   As buffer operation has been set, when the TCNT value is stored in TGRA upon the occurrence of input capture A, the value previously stored in TGRA is simultaneously transferred to TGRC.

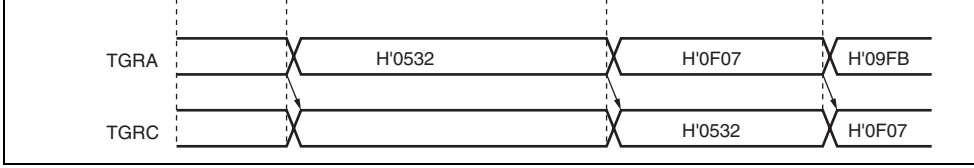


Figure 10.16 Example of Buffer Operation (2)

10.4.4 Cascaded Operation

In cascaded operation, two 16-bit counters for different channels are used together as a 32
counter.

This function works by counting the channel 1 (channel 4) counter clock upon overflow/u
of TCNT_2 (TCNT_5) as set in the TPSC0 to TPSC2 bits in TCR.

Underflow occurs only when the lower 16-bit TCNT is in phase-counting mode.

Table 10.29 shows the register combinations used in cascaded operation.

Note: When phase counting mode is set for channel 1 or 4, the counter clock setting is i
and the counters operates independently in phase counting mode.

Table 10.29 Cascaded Combinations

Combination	Upper 16 Bits	Lower 16 Bits
Channels 1 and 2	TCNT_1	TCNT_2
Channels 4 and 5	TCNT_4	TCNT_5

<Cascaded operation>

Figure 10.17 Cascaded Operation Setting Procedure

Examples of Cascaded Operation: Figure 10.18 illustrates the operation when TCNT_ overflow/underflow counting has been set for TCNT_1, when TGRA_1 and TGRA_2 h designated as input capture registers, and when TIOC pin rising edge has been selected.

When a rising edge is input to the TIOCA1 and TIOCA2 pins simultaneously, the upper the 32-bit data are transferred to TGRA_1, and the lower 16 bits to TGRA_2.

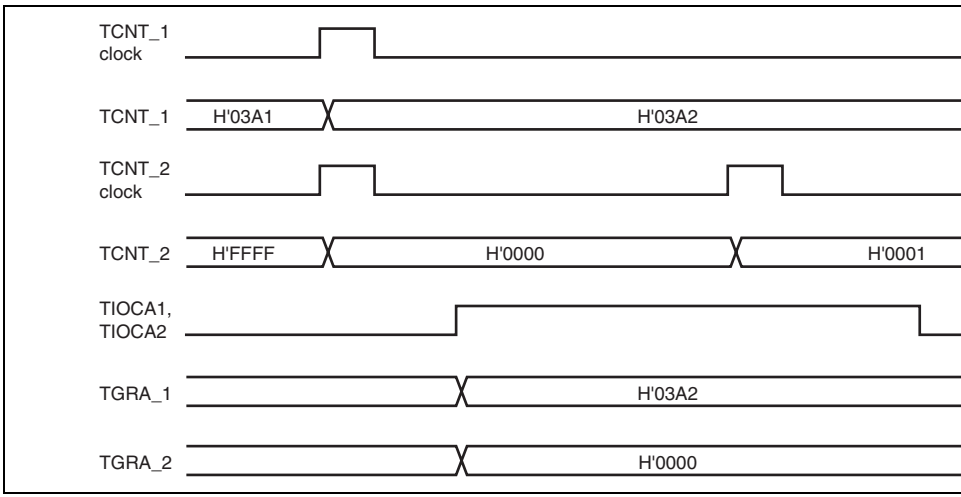


Figure 10.18 Example of Cascaded Operation (1)

Figure 10.19 illustrates the operation when TCNT_2 overflow/underflow counting has b TCNT_1 and phase counting mode has been designated for channel 2.

Figure 10.19 Example of Cascaded Operation (2)

10.4.5 PWM Modes

In PWM mode, PWM waveforms are output from the output pins. The output level can b
as 0, 1, or toggle output in response to a compare match of each TGR.

TGR registers settings can be used to output a PWM waveform in the range of 0% to 100

Designating TGR compare match as the counter clearing source enables the period to be s
register. All channels can be designated for PWM mode independently. Synchronous ope
also possible.

There are two PWM modes, as described below.

- PWM mode 1

  PWM output is generated from the TIOCA and TIOCC pins by pairing TGRA with T
  TGRC with TGRD. The output specified by the IOA0 to IOA3 bits and IOC0 to IOC3
  TIOR is output from the TIOCA and TIOCC pins at compare matches A and C, and th
  specified by the IOB0 to IOB3 bits and IOD0 to IOD3 bits in TIOR is output at comp
  matches B and D. The initial output value is the value set in TGRA or TGRC. If the s
  of paired TGRs are identical, the output value does not change when a compare match

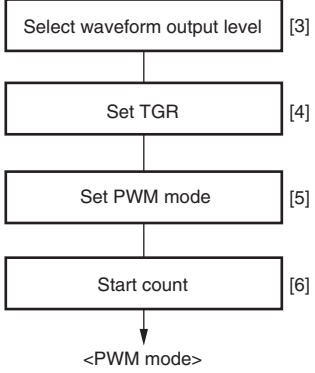
  In PWM mode 1, a maximum 8-phase PWM output is possible.

- PWM mode 2

  PWM output is generated using one TGR as the cycle register and the others as duty r
  The output specified in TIOR is performed by means of compare matches. Upon coun
  clearing by a synchronization register compare match, the output value of each pin is
  value set in TIOR. If the set values of the cycle and duty registers are identical, the ou
  value does not change when a compare match occurs.

	TGRC_0	TIOCC0	TIOCC0
	TGRD_0		TIOCD0
1	TGRA_1	TIOCA1	TIOCA1
	TGRB_1		TIOCB1
2	TGRA_2	TIOCA2	TIOCA2
	TGRB_2		TIOCB2
3	TGRA_3	TIOCA3	TIOCA3
	TGRB_3		TIOCB3
	TGRC_3	TIOCC3	TIOCC3
	TGRD_3		TIOCD3
4	TGRA_4	TIOCA4	TIOCA4
	TGRB_4		TIOCB4
5	TGRA_5	TIOCA5	TIOCA5
	TGRB_5		TIOCB5

Note: In PWM mode 2, PWM output is not possible for the TGR register in which the pe



[5] Select the PWM mode with the MD3 to MD0 bits in TMDR.
[6] Set the CST bit in TSTR to 1 start the count operation.

Figure 10.20 Example of PWM Mode Setting Procedure

Examples of PWM Mode Operation: Figure 10.21 shows an example of PWM mode 1 operation.

In this example, TGRA compare match is set as the TCNT clearing source, 0 is set for the initial output value and output value, and 1 is set as the TGRB output value.

In this case, the value set in TGRA is used as the period, and the values set in the TGRB are used as the duty levels.

Figure 10.22 shows an example of PWM mode 2 operation.

In this example, synchronous operation is designated for channels 0 and 1, TGRB_1 con
match is set as the TCNT clearing source, and 0 is set for the initial output value and 1 f
output value of the other TGR registers (TGRA_0 to TGRD_0, TGRA_1), outputting a
PWM waveform.

In this case, the value set in TGRB_1 is used as the cycle, and the values set in the other
used as the duty levels.

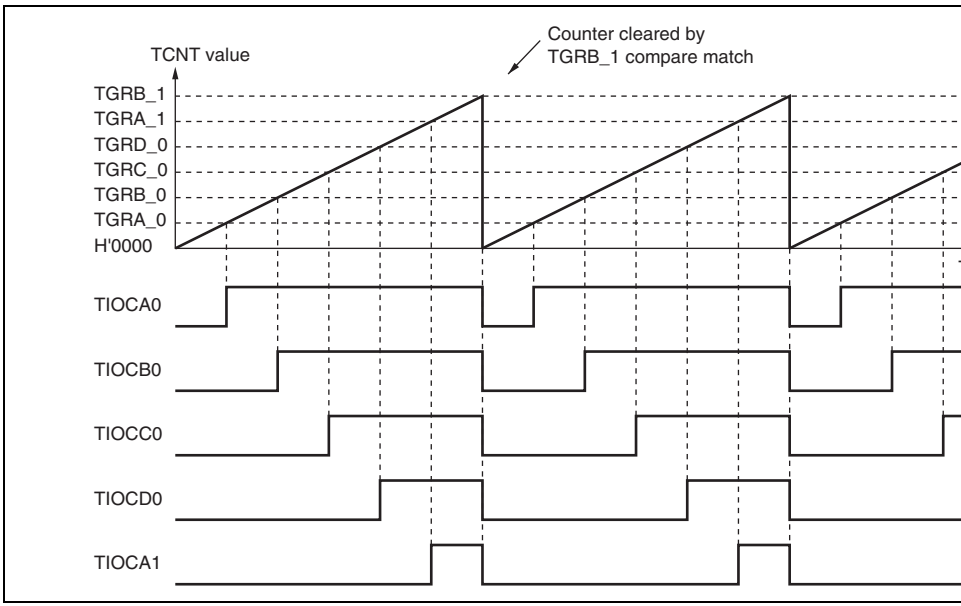


Figure 10.22 Example of PWM Mode Operation (2)

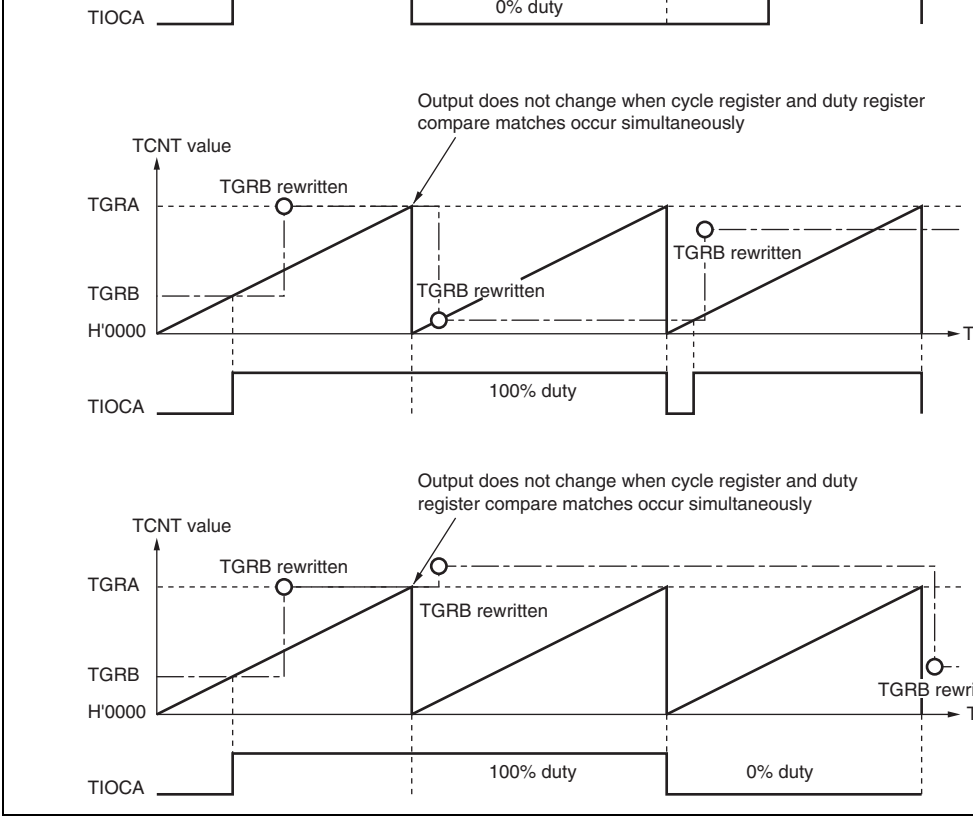


Figure 10.23 Example of PWM Mode Operation (3)

This can be used for two-phase encoder pulse input.

If overflow occurs when TCNT is counting up, the TCFV flag in TSR is set; if underflo
when TCNT is counting down, the TCFU flag is set.

The TCFD bit in TSR is the count direction flag. Reading the TCFD flag reveals whethe
counting up or down.

Table 10.31 shows the correspondence between external clock pins and channels.

Table 10.31 Phase Counting Mode Clock Input Pins

	External Clock Pins	
Channels	**A-Phase**	**B-Phase**
When channel 1 or 5 is set to phase counting mode	TCLKA	TCLKB
When channel 2 or 4 is set to phase counting mode	TCLKC	TCLKD

Example of Phase Counting Mode Setting Procedure: Figure 10.24 shows an exampl
phase counting mode setting procedure.

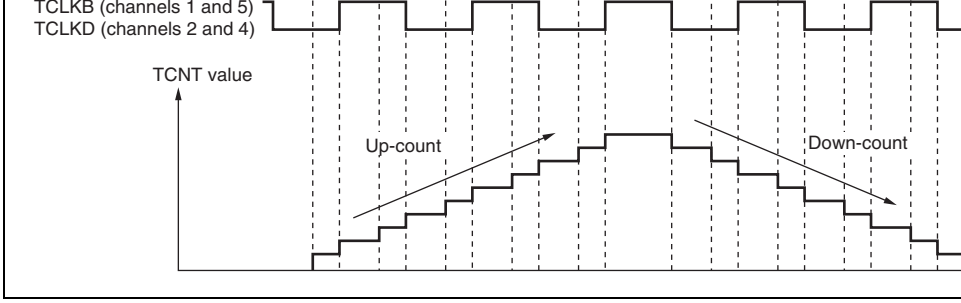


Figure 10.25 Example of Phase Counting Mode 1 Operation

Table 10.32 Up/Down-Count Conditions in Phase Counting Mode 1

TCLKA (Channels 1 and 5) TCLKC (Channels 2 and 4)	TCLKB (Channels 1 and 5) TCLKD (Channels 2 and 4)	Operation
High level	↑	Up-count
Low level	↓	Up-count
↑	Low level	Up-count
↓	High level	Up-count
High level	↓	Down-count
Low level	↑	Down-count
↑	High level	Down-count
↓	Low level	Down-count

Legend:
↑: Rising edge
↓: Falling edge

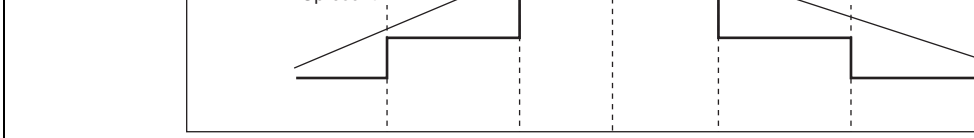


Figure 10.26 Example of Phase Counting Mode 2 Operation

Table 10.33 Up/Down-Count Conditions in Phase Counting Mode 2

TCLKA (Channels 1 and 5) TCLKC (Channels 2 and 4)	TCLKB (Channels 1 and 5) TCLKD (Channels 2 and 4)	Operation
High level	↑	Don't care
Low level	↓	Don't care
↑	Low level	Don't care
↓	High level	Up-count
High level	↓	Don't care
Low level	↑	Don't care
↑	High level	Don't care
↓	Low level	Down-count

Legend:
↑: Rising edge
↓: Falling edge

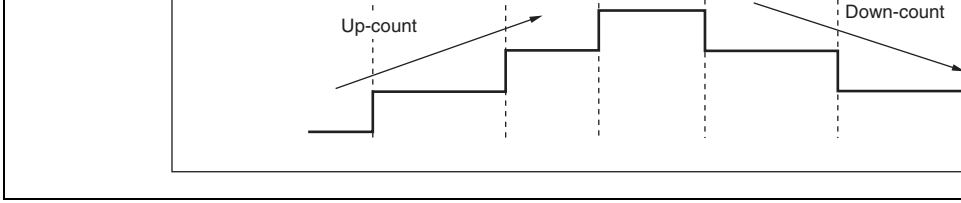


Figure 10.27 Example of Phase Counting Mode 3 Operation

Table 10.34 Up/Down-Count Conditions in Phase Counting Mode 3

TCLKA (Channels 1 and 5) TCLKC (Channels 2 and 4)	TCLKB (Channels 1 and 5) TCLKD (Channels 2 and 4)	Operation
High level	↑	Don't care
Low level	↓	Don't care
↑	Low level	Don't care
↓	High level	Up-count
High level	↓	Down-count
Low level	↑	Don't care
↑	High level	Don't care
↓	Low level	Don't care

Legend:
↑: Rising edge
↓: Falling edge

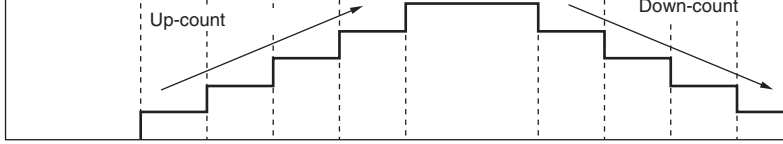


Figure 10.28 Example of Phase Counting Mode 4 Operation

Table 10.35 Up/Down-Count Conditions in Phase Counting Mode 4

TCLKA (Channels 1 and 5) TCLKC (Channels 2 and 4)	TCLKB (Channels 1 and 5) TCLKD (Channels 2 and 4)	Operation
High level	↑	Up-count
Low level	↓	Up-count
↑	Low level	Don't care
↓	High level	Don't care
High level	↓	Down-count
Low level	↑	Down-count
↑	High level	Don't care
↓	Low level	Don't care

Legend:
↑: Rising edge
↓: Falling edge

source, and the pulse widths of 2-phase encoder 4-multiplication pulses are detected.

TGRA_1 and TGRB_1 for channel 1 are designated for input capture, and channel 0 TG
TGRC_0 compare matches are selected as the input capture source and store the up/dow
values for the control periods.

This procedure enables the accurate detection of position and speed.

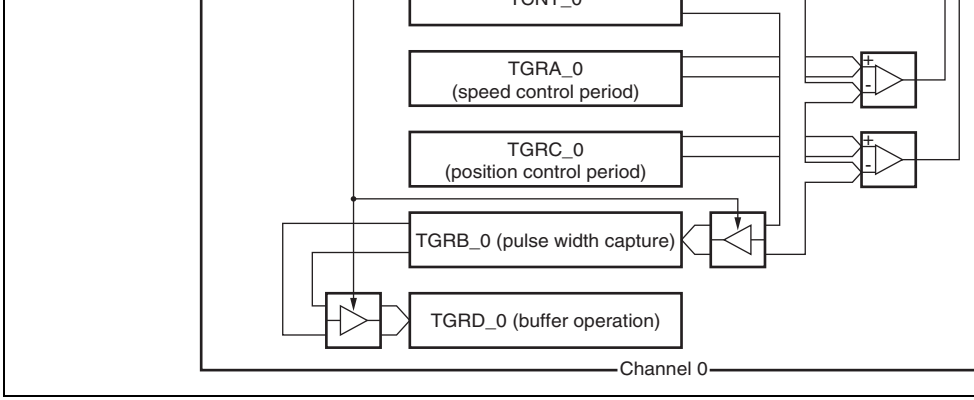


Figure 10.29 Phase Counting Mode Application Example

10.5 Interrupts

There are three kinds of TPU interrupt source; TGR input capture/compare match, TCNT overflow, and TCNT underflow. Each interrupt source has its own status flag and enable/ bit, allowing the generation of interrupt request signals to be enabled or disabled individu

When an interrupt request is generated, the corresponding status flag in TSR is set to 1. If corresponding enable/disable bit in TIER is set to 1 at this time, an interrupt is requested. interrupt request is cleared by clearing the status flag to 0.

Relative channel priorities can be changed by the interrupt controller, however the priorit within a channel is fixed. For details, see section 5, Interrupt Controller.

1	TGI1A	TGRA_1 input capture/compare match	TGFA_1	Pos
	TGI1B	TGRB_1 input capture/compare match	TGFB_1	Pos
	TCI1V	TCNT_1 overflow	TCFV_1	Not
	TCI1U	TCNT_1 underflow	TCFU_1	Not
2	TGI2A	TGRA_2 input capture/compare match	TGFA_2	Pos
	TGI2B	TGRB_2 input capture/compare match	TGFB_2	Pos
	TCI2V	TCNT_2 overflow	TCFV_2	Not
	TCI2U	TCNT_2 underflow	TCFU_2	Not
3	TGI3A	TGRA_3 input capture/compare match	TGFA_3	Pos
	TGI3B	TGRB_3 input capture/compare match	TGFB_3	Pos
	TGI3C	TGRC_3 input capture/compare match	TGFC_3	Pos
	TGI3D	TGRD_3 input capture/compare match	TGFD_3	Pos
	TCI3V	TCNT_3 overflow	TCFV_3	Not
4	TGI4A	TGRA_4 input capture/compare match	TGFA_4	Pos
	TGI4B	TGRB_4 input capture/compare match	TGFB_4	Pos
	TCI4V	TCNT_4 overflow	TCFV_4	Not
	TCI4U	TCNT_4 underflow	TCFU_4	Not
5	TGI5A	TGRA_5 input capture/compare match	TGFA_5	Pos
	TGI5B	TGRB_5 input capture/compare match	TGFB_5	Pos
	TCI5V	TCNT_5 overflow	TCFV_5	Not
	TCI5U	TCNT_5 underflow	TCFU_5	Not

Note: This table shows the initial state immediately after a reset. The relative channel p
can be changed by the interrupt controller.

Underflow Interrupt: An interrupt is requested if the TCIEU bit in TIER is set to 1 whe
TCFU flag in TSR is set to 1 by the occurrence of TCNT underflow on a channel. The int
request is cleared by clearing the TCFU flag to 0. The TPU has four underflow interrupts
each for channels 1, 2, 4, and 5.

10.6 DTC Activation

The DTC can be activated by the TGR input capture/compare match interrupt for a chann
details, see section 8, Data Transfer Controller (DTC).

A total of 16 TPU input capture/compare match interrupts can be used as DTC activation
four each for channels 0 and 3, and two each for channels 1, 2, 4, and 5.

10.7 A/D Converter Activation

The A/D converter can be activated by the TGRA input capture/compare match for a cha

If the TTGE bit in TIER is set to 1 when the TGFA flag in TSR is set to 1 by the occurren
TGRA input capture/compare match on a particular channel, a request to begin A/D conv
sent to the A/D converter. If the TPU conversion start trigger has been selected on the A/
converter side at this time, A/D conversion is begun.

In the TPU, a total of six TGRA input capture/compare match interrupts can be used as A
converter conversion start sources, one for each channel.

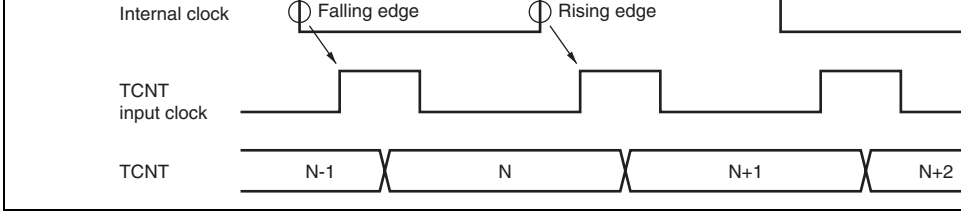


Figure 10.30 Count Timing in Internal Clock Operation

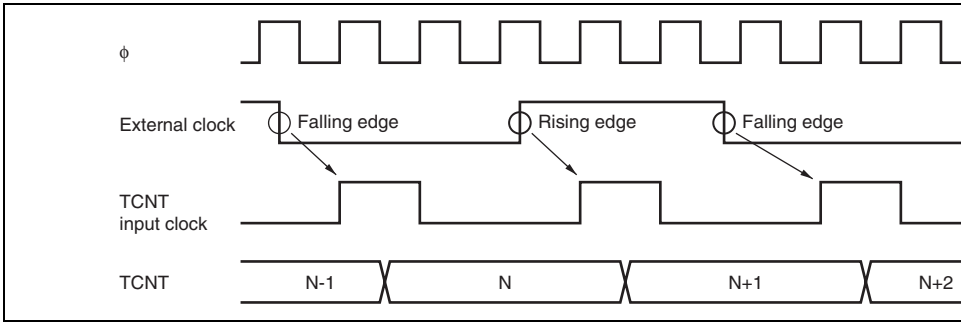


Figure 10.31 Count Timing in External Clock Operation

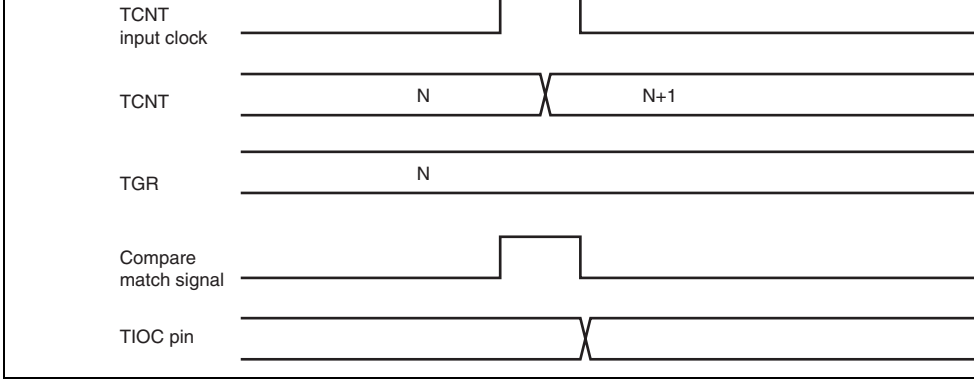


Figure 10.32 Output Compare Output Timing

Input Capture Signal Timing: Figure 10.33 shows input capture signal timing.

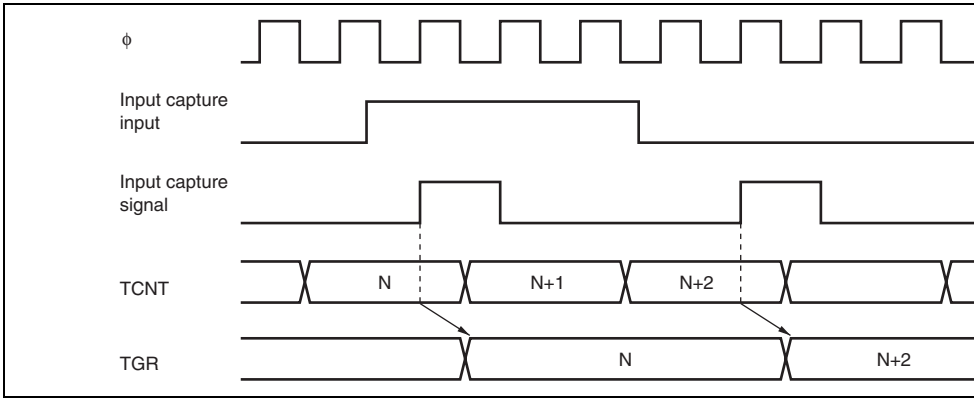


Figure 10.33 Input Capture Input Signal Timing

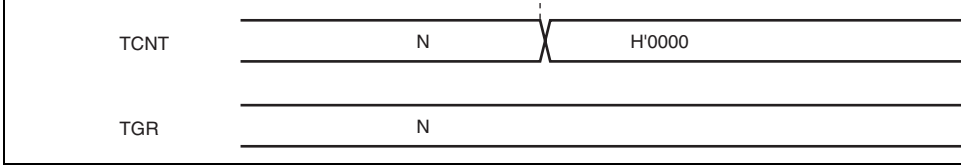


Figure 10.34 Counter Clear Timing (Compare Match)

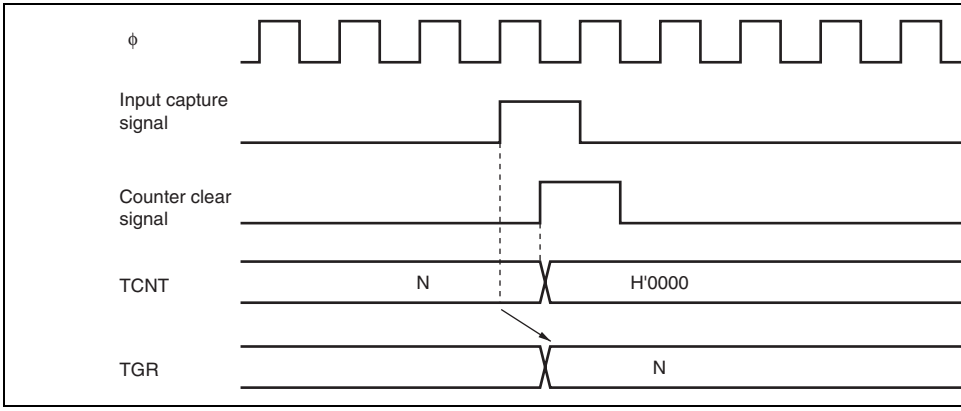


Figure 10.35 Counter Clear Timing (Input Capture)

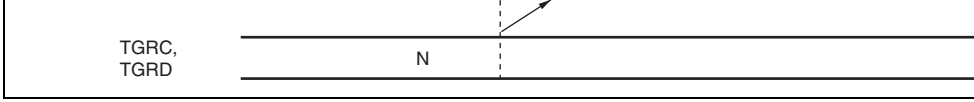


Figure 10.36 Buffer Operation Timing (Compare Match)

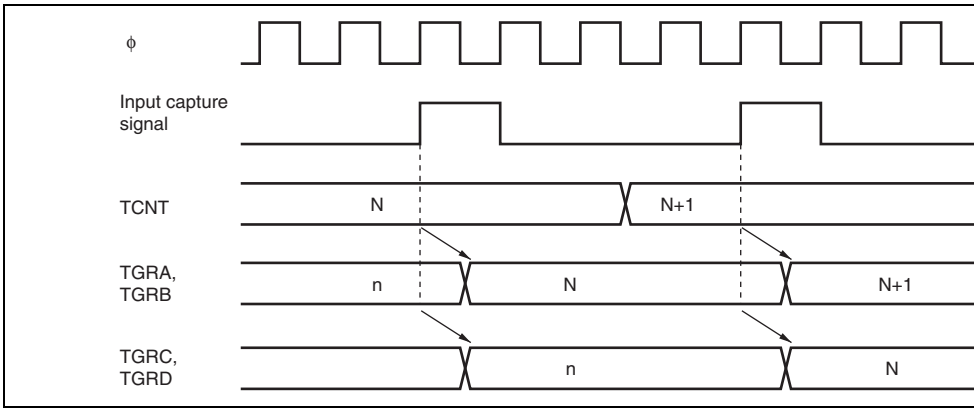


Figure 10.37 Buffer Operation Timing (Input Capture)

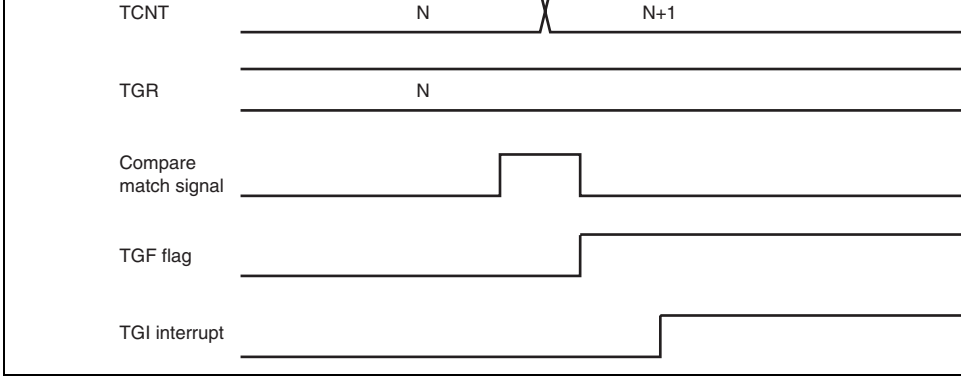


Figure 10.38 TGI Interrupt Timing (Compare Match)

TGF Flag Setting Timing in Case of Input Capture: Figure 10.39 shows the timing fo
of the TGF flag in TSR on input capture, and TGI interrupt request signal timing.

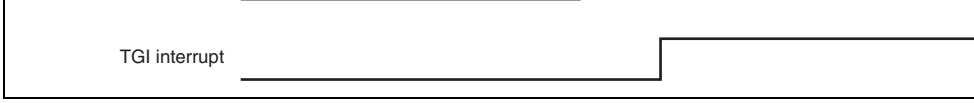


Figure 10.39 TGI Interrupt Timing (Input Capture)

TCFV Flag/TCFU Flag Setting Timing: Figure 10.40 shows the timing for setting of th flag in TSR on overflow, and TCIV interrupt request signal timing.

Figure 10.41 shows the timing for setting of the TCFU flag in TSR on underflow, and TC interrupt request signal timing.

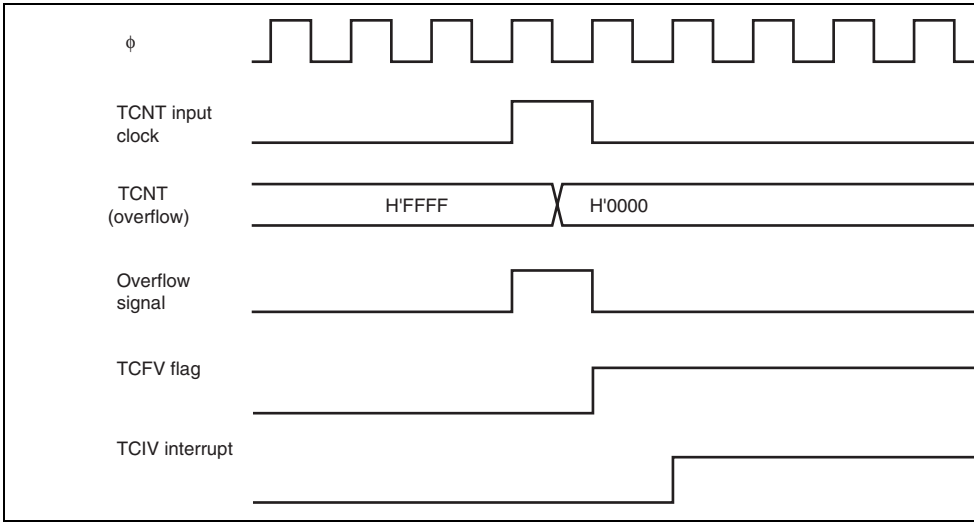


Figure 10.40 TCIV Interrupt Setting Timing

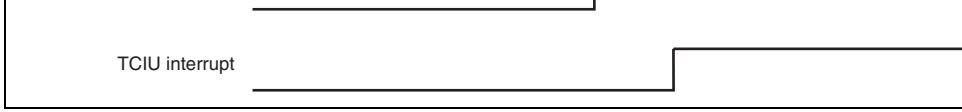


Figure 10.41 TCIU Interrupt Setting Timing

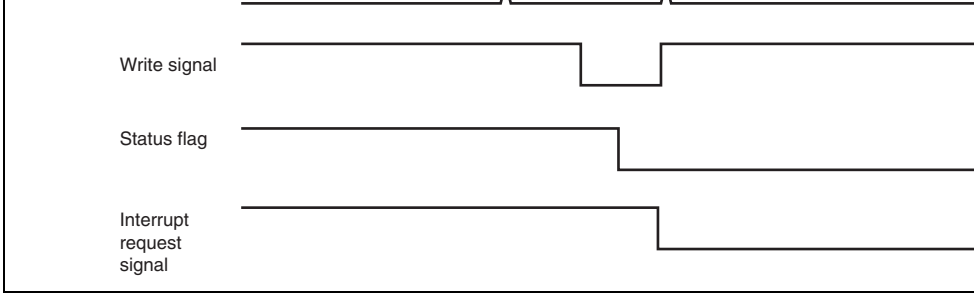


Figure 10.42 Timing for Status Flag Clearing by CPU

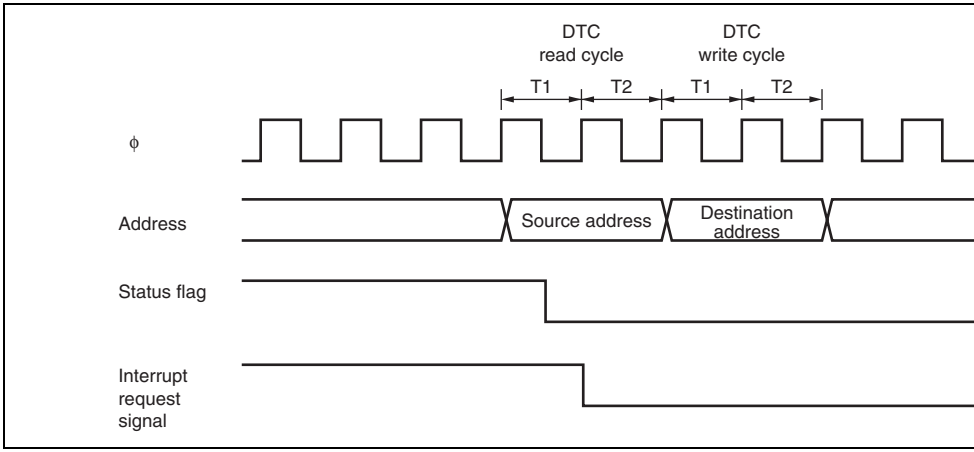


Figure 10.43 Timing for Status Flag Clearing by DTC Activation

The input clock pulse width must be at least 1.5 states in the case of single-edge detectio least 2.5 states in the case of both-edge detection. The TPU will not operate properly at r pulse widths.

In phase counting mode, the phase difference and overlap between the two input clocks least 1.5 states, and the pulse width must be at least 2.5 states. Figure 10.44 shows the in conditions in phase counting mode.

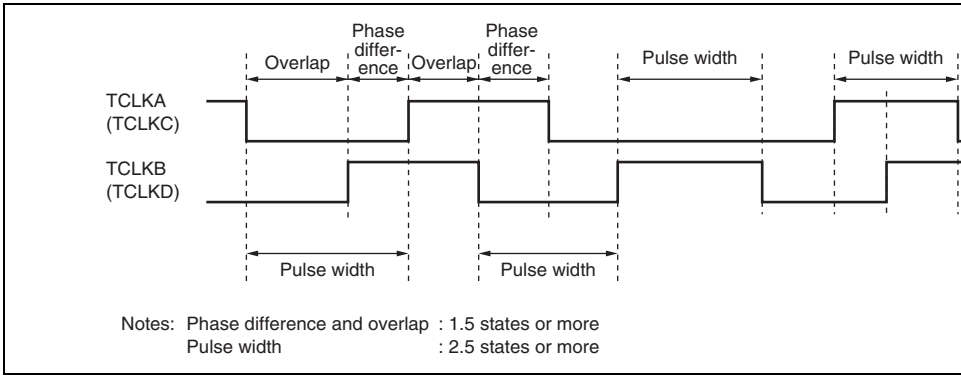


Figure 10.44 Phase Difference, Overlap, and Pulse Width in Phase Counting

N : TGR set value

10.9.4 Contention between TCNT Write and Clear Operations

If the counter clear signal is generated in the T2 state of a TCNT write cycle, TCNT clear precedence and the TCNT write is not performed.

Figure 10.45 shows the timing in this case.

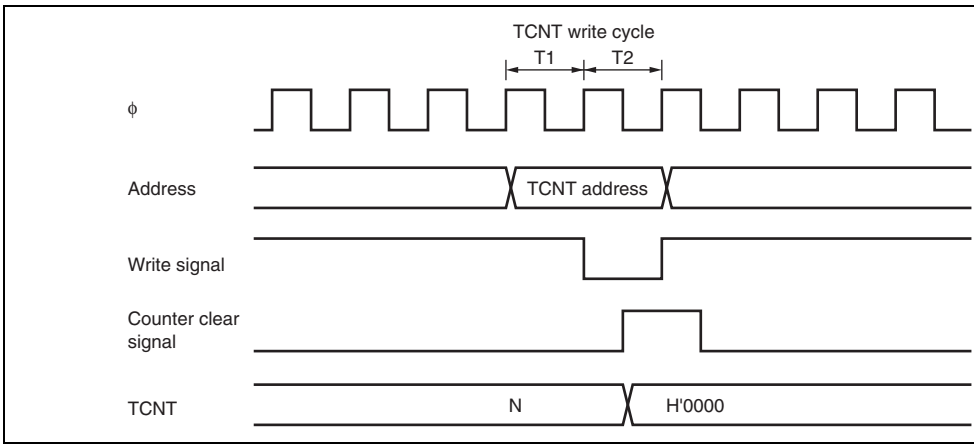


Figure 10.45 Contention between TCNT Write and Clear Operations

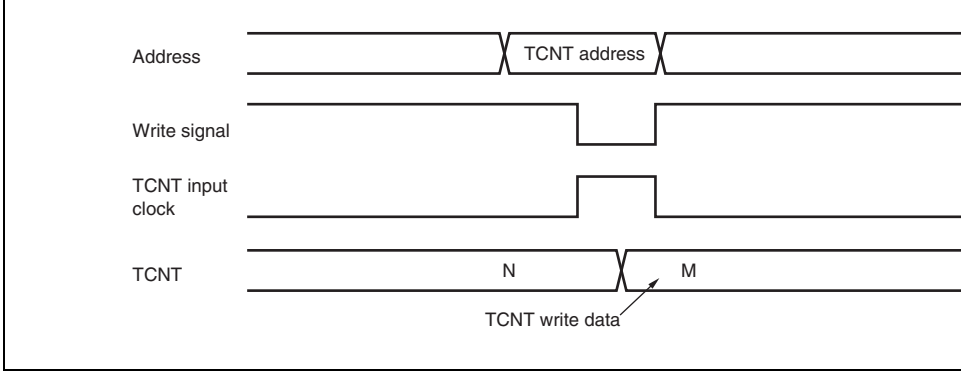


Figure 10.46 Contention between TCNT Write and Increment Operation

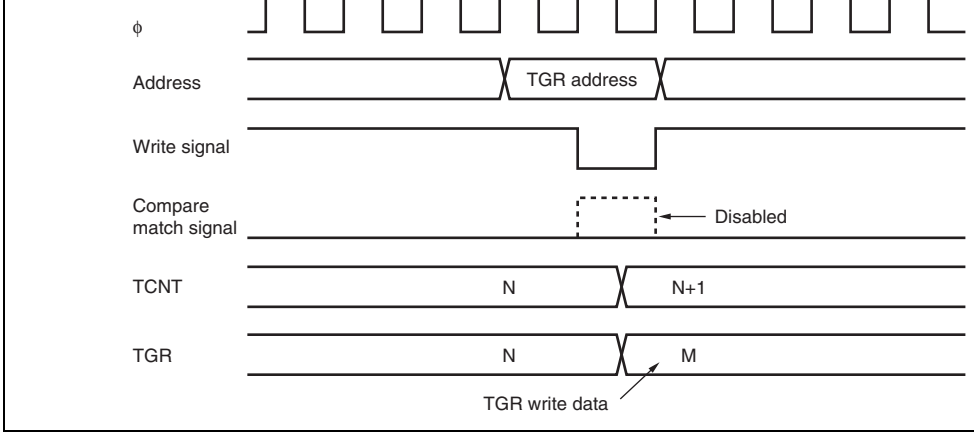


Figure 10.47 Contention between TGR Write and Compare Match

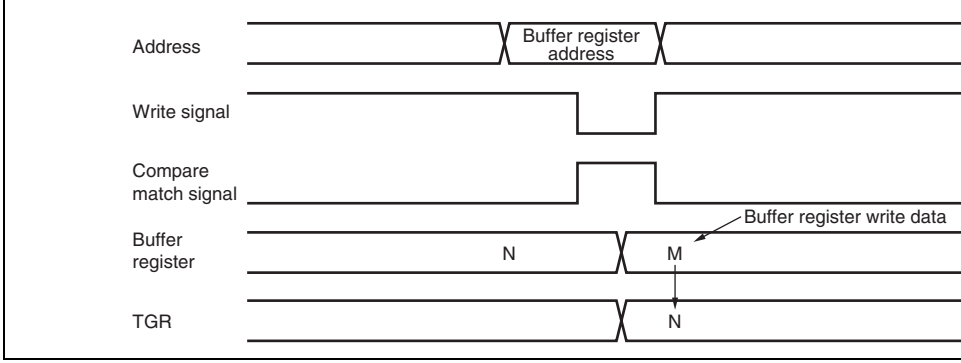


Figure 10.48 Contention between Buffer Register Write and Compare Ma

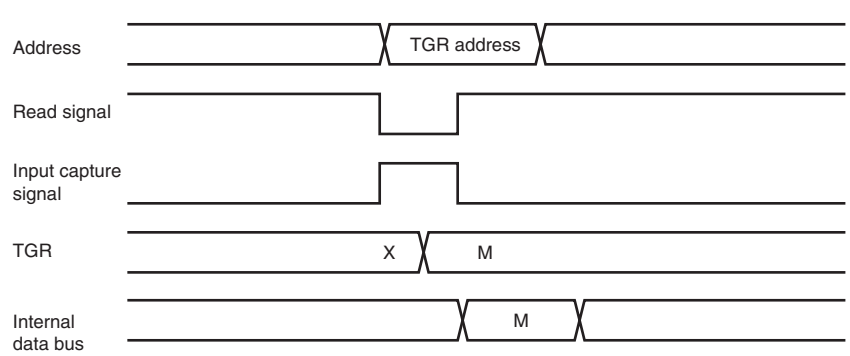


Figure 10.49 Contention between TGR Read and Input Capture

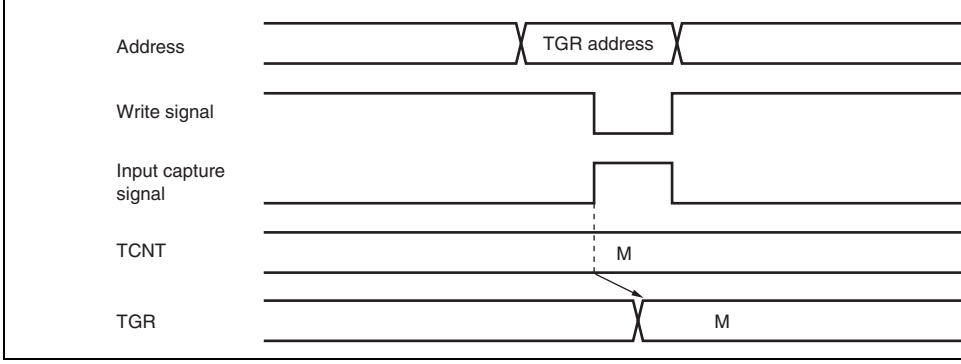


Figure 10.50 Contention between TGR Write and Input Capture

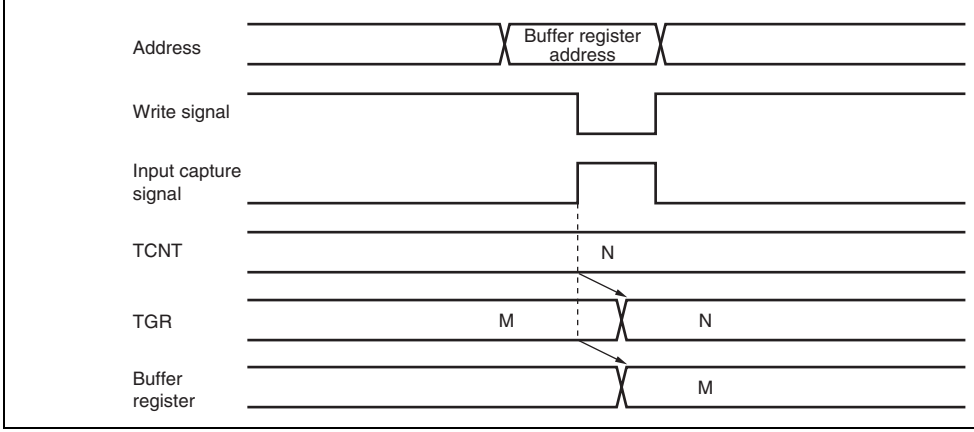


Figure 10.51 Contention between Buffer Register Write and Input Capture

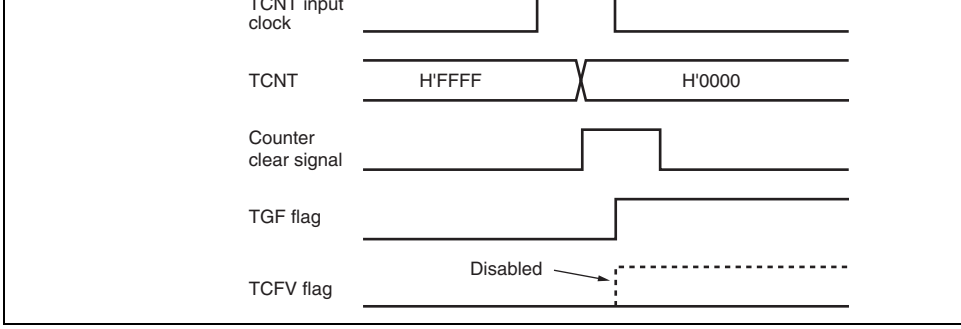


Figure 10.52 Contention between Overflow and Counter Clearing

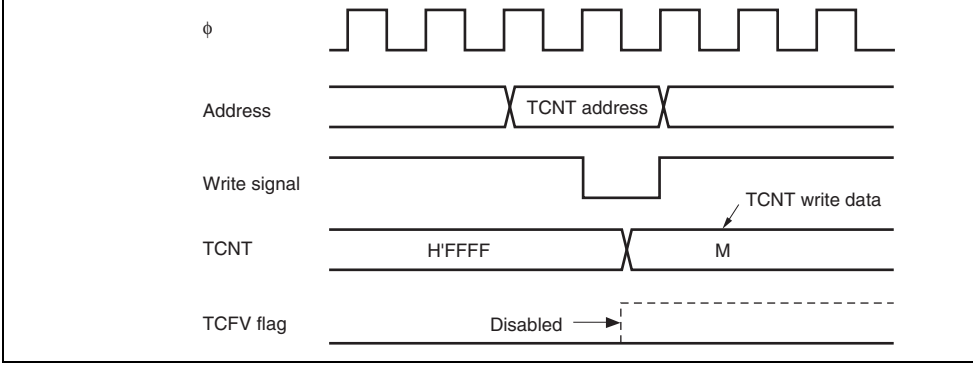


Figure 10.53 Contention between TCNT Write and Overflow

10.9.13 Multiplexing of I/O Pins

In this LSI, the TCLKA input pin is multiplexed with the TIOCC0 I/O pin, the TCLKB in with the TIOCD0 I/O pin, the TCLKC input pin with the TIOCB1 I/O pin, and the TCLK pin with the TIOCB2 I/O pin. When an external clock is input, compare match output sho be performed from a multiplexed pin.

10.9.14 Interrupts in Module Stop Mode

If module stop mode is entered when an interrupt has been requested, it will not be possib clear the CPU interrupt source or the DTC activation source. Interrupts should therefore b disabled before entering module stop mode.

Selected from three internal clocks (ϕ/8, ϕ/64, and ϕ/8192) and an external clock

- Selection of three ways to clear the counters

  The counters can be cleared on compare-match A or B, or by an external reset signal
- Timer output controlled by two compare-match signals

  The timer output signal in each channel is controlled by two independent compare-m signals, enabling the timer to be used for various applications, such as the generation output or PWM output with an arbitrary duty cycle
- Cascading of the two channels

  (Cascading of TMR_0, TMR_1)

  The module can operate as a 16-bit timer using TMR_0 as the upper half and TMR_ lower half (16-bit count mode)

  TMR_1 can be used to count TMR_0 compare-match occurrences (compare-match c mode)

  (Cascading of TMR_2, TMR_3)

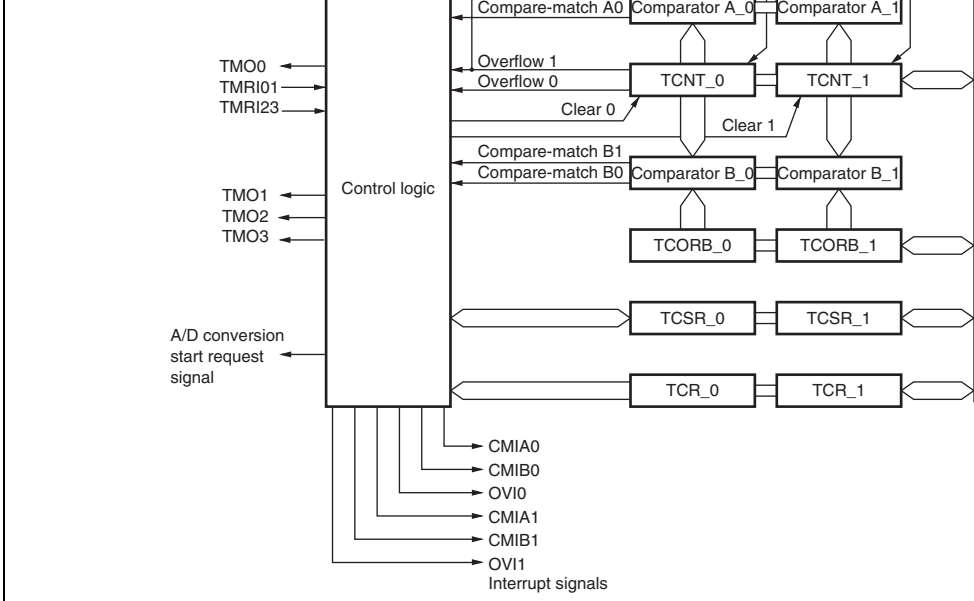
  The module can operate as a 16-bit timer using TMR_2 as the upper half and TMR_ lower half (16-bit count mode)

  TMR_3 can be used to count TMR_2 compare-match occurrences (compare-match c mode)
- Multiple interrupt sources for each channel

  Two compare-match interrupts and one overflow interrupt can be requested indepen
- Generation of A/D converter conversion start trigger

  Channel 0 compare-match signal can be used as the A/D converter conversion start t
- Module stop mode can be set

  As the initial setting, the 8-bit timer operation is halted. Register access is enabled by canceling the module stop mode.



Legend:

TCORA_0:	Time constant register A_0	TCORA_1:	Time constant register A_1
TCORB_0:	Time constant register B_0	TCORB_1:	Time constant register B_1
TCNT_0:	Timer counter_0	TCNT_1:	Timer counter_1
TCSR_0:	Timer control/status register_0	TCSR_1:	Timer control/status register_1
TCR_0:	Timer control register_0	TCR_1:	Timer control register_1

Figure 11.1 Block Diagram of 8-Bit Timer Module

	Timer reset input	TMRI01	Input	External reset input for the
2	Timer output	TMO2	Output	Output controlled by comp
3	Timer output	TMO3	Output	Output controlled by comp
Common to 2 and 3	Timer clock input	TMCI23	Input	External clock input for the
	Timer reset input	TMRI23	Input	External reset input for the

11.3 Register Descriptions

The 8-bit timer has the following registers. For details on the module stop register, see s
22.1.2, Module Stop Control Registers A to C (MSTPCRA to MSTPCRC).

Channel 0

- Timer counter_0 (TCNT_0)
- Time constant register A_0 (TCORA_0)
- Time constant register B_0 (TCORB_0)
- Timer control register_0 (TCR_0)
- Timer control/status register_0 (TCSR_0)

Channel 1

- Timer counter_1 (TCNT_1)
- Time constant register A_1 (TCORA_1)
- Time constant register B_1 (TCORB_1)
- Timer control register_1 (TCR_1)
- Timer control/status register_1 (TCSR_1)

- Time constant register B_3 (TCORB_3)
- Timer control register_3 (TCR_3)
- Timer control/status register_3 (TCSR_3)

11.3.1 Timer Counter (TCNT)

Each TCNT is an 8-bit up-counter. TCNT_0 and TCNT_1 (or TCNT_2 and TCNT_3) co
single 16-bit register, so they can be accessed together by word access.

This clock source is selected by the clock select bits, CKS2 to CKS0, in TCR. TCNT can
cleared by an external reset input signal or compare-match signals A and B. The CCLR1
CCLR0 bits in TCR select the method of TCNT clearing.

When TCNT overflows from H'FF to H'00, the overflow flag (OVF) in TCSR is set to 1.
initial value of TCNT is H'00.

11.3.2 Time Constant Register A (TCORA)

TCORA is an 8-bit readable/writable register. TCORA_0 and TCORA_1 (or TCORA_2
TCORA_3) comprise a single 16-bit register, so they can be accessed together by word a

TCORA is continually compared with the value in TCNT. When a match is detected, the
corresponding compare-match flag A (CMFA) in TCSR is set. Note, however, that comp
disabled during the T_2 state of a TCORA write cycle.

The timer output from the TMO pin can be freely controlled by the compare-match signa
the settings of the output select bits, OS1 and OS0, in TCSR.

The initial value of TCORA is H'FF.

The initial value of TCORB is H'FF.

11.3.4 Timer Control Register (TCR)

TCR selects the TCNT clock source and the time at which TCNT is cleared, and control requests.

Bit	Bit Name	Initial Value	R/W	Description
7	CMIEB	0	R/W	Compare-Match Interrupt Enable B Selects whether the CMFB interrupt request (CM enabled or disabled when the CMFB flag in TCSI 1. 0: CMFB interrupt request (CMIB) is disabled 1: CMFB interrupt request (CMIB) is enabled
6	CMIEA	0	R/W	Compare-Match Interrupt Enable A Selects whether the CMFA interrupt request (CM enabled or disabled when the CMFA flag in TCSI 1. 0: CMFA interrupt request (CMIA) is disabled 1: CMFA interrupt request (CMIA) is enabled
5	OVIE	0	R/W	Timer Overflow Interrupt Enable Selects whether the OVF interrupt request (OVI) or disabled when the OVF flag in TCSR is set to 0: OVF interrupt request (OVI) is disabled 1: OVF interrupt request (OVI) is enabled

0	CKS0	0	R/W	from the system clock (φ). When use of an externa selected, three types of count can be selected: at edge, the falling edge, and both rising and falling e 000: Clock input disabled 001: φ/8 internal clock source, counted on the falli 010: φ/64 internal clock source, counted on the fal 011: φ/8192 internal clock source, counted on the edge 100: For channel 0: Counted on TCNT1 overflow s For channel 1: Counted on TCNT0 compare- signal* For channel 2: Counted on TCNT3 overflow s For channel 3: Counted on TCNT2 compare- signal* 101: External clock source, counted at rising edge 110: External clock source, counted at falling edge 111: External clock source, counted at both rising falling edges

Note: * If the count input of channel 0 (channel 2) is the TCNT1 (TCNT3) overflow sign
that of channel 1 (channel 3) is the TCNT0 (TCNT2) compare-match signal, no
incrementing clock will be generated. Do not use this setting.

				[Clearing conditions] • Read CMFB when CMFB = 1, then write 0 in • When DTC is activated by CMIB interrupt, the bit in MRB of DTC is 0 with the transfer count than 0.
6	CMFA	0	R/(W)*	Compare-Match Flag A [Setting condition] When TCNT = TCORA [Clearing conditions] • Read CMFA when CMFA = 1, then write 0 in • When DTC is activated by CMIA interrupt, the bit in MRB of DTC is 0 with the transfer count than 0.
5	OVF	0	R/(W)*	Timer Overflow Flag [Setting condition] When TCNT overflows from H'FF to H'00 [Clearing condition] Read OVF when OVF = 1, then write 0 in OVF
4	ADTE	0	R/W	A/D Trigger Enable Enables or disables A/D converter start requests compare-match A. 0: A/D converter start requests by compare-matc disabled 1: A/D converter start requests by compare-matc enabled

				(toggle output)
1	OS1	0	R/W	Output Select 1 and 0
0	OS0	0	R/W	These bits specify how the timer output level changed by a compare-match A of TCORA a TCNT. 00: No change when compare-match A occu 01: 0 is output when compare-match A occur 10: 1 is output when compare-match A occur 11: Output is inverted when compare-match (toggle output)

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

				than 0.
6	CMFA	0	R/(W)*	Compare-Match Flag A [Setting condition] When TCNT = TCORA [Clearing conditions] • Read CMFA when CMFA = 1, then write 0 in • When DTC is activated by CMIA interrupt, the bit in MRB of DTC is 0 with the transfer count than 0.
5	OVF	0	R/(W)*	Timer Overflow Flag [Setting condition] When TCNT overflows from H'FF to H'00 [Clearing condition] Read OVF when OVF = 1, then write 0 in OVF
4	—	1	—	Reserved This bit is always read as 1 and cannot be modifi
3	OS3	0	R/W	Output Select 3 and 2
2	OS2	0	R/W	These bits specify how the timer output level is to changed by a compare-match B of TCORB and T 00: No change when compare-match B occurs 01: 0 is output when compare-match B occurs 10: 1 is output when compare-match B occurs 11: Output is inverted when compare-match B oc (toggle output)

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

- TCSR_2

Bit	Bit Name	Initial Value	R/W	Description
7	CMFB	0	R/(W)*	Compare-Match Flag B [Setting condition] When TCNT = TCORB [Clearing conditions] • Read CMFB when CMFB = 1, then write 0 in C • When DTC is activated by CMIB interrupt, the bit in MRB of DTC is 0 with the transfer counte than 0.
6	CMFA	0	R/(W)*	Compare-Match Flag A [Setting condition] When TCNT = TCORA [Clearing conditions] • Read CMFA when CMFA = 1, then write 0 in C • When DTC is activated by CMIA interrupt, the bit in MRB of DTC is 0 with the transfer counte than 0.

3	OS3	0	R/W	Output Select 3 and 2
2	OS2	0	R/W	These bits specify how the timer output level is to changed by a compare-match B of TCORB and T 00: No change when compare-match B occurs 01: 0 is output when compare-match B occurs 10: 1 is output when compare-match B occurs 11: Output is inverted when compare-match B oc (toggle output)
1	OS1	0	R/W	Output Select 1 and 0
0	OS0	0	R/W	These bits specify how the timer output level is to changed by a compare-match A of TCORA and T 00: No change when compare-match A occurs 01: 0 is output when compare-match A occurs 10: 1 is output when compare-match A occurs 11: Output is inverted when compare-match A oc (toggle output)

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

By the above settings, waveforms with the cycle of TCORA and the pulse width of TCOR
be output without software intervention.

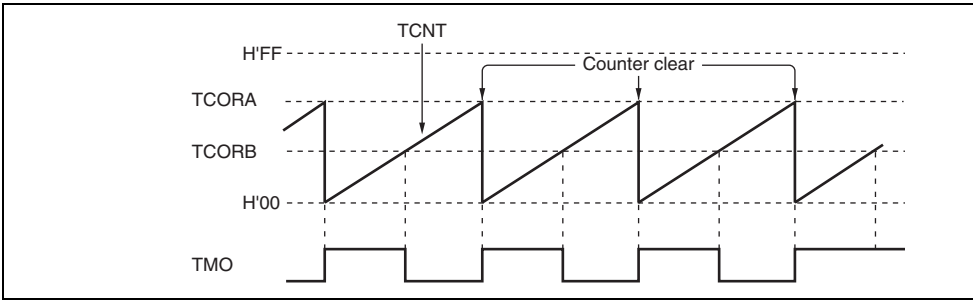


Figure 11.2 Example of Pulse Output

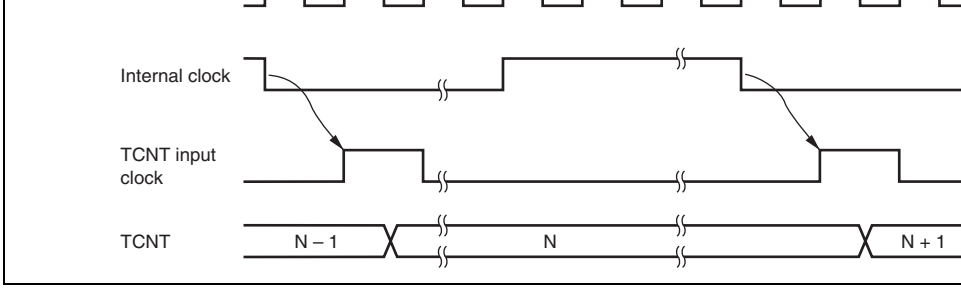


Figure 11.3 Count Timing for Internal Clock Input

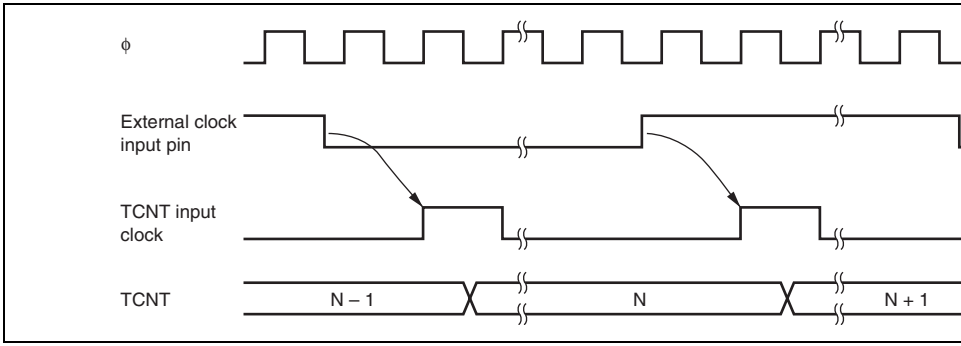


Figure 11.4 Count Timing for External Clock Input

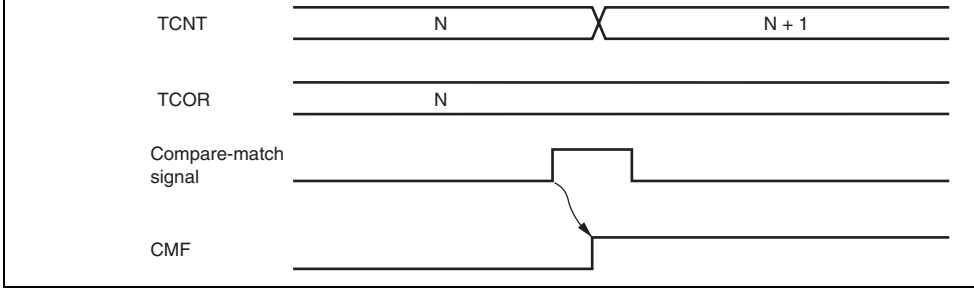


Figure 11.5 Timing of CMF Flag Setting

11.5.3 Timing of Timer Output When a Compare-Match Occurs

When a compare-match occurs, the timer output changes as specified by the output select OS3 to OS0, in TCSR. Figure 11.6 shows the timing when the output is set to toggle at co match A.

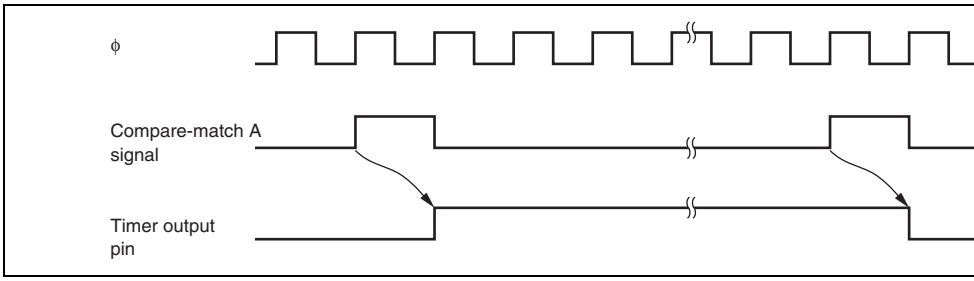


Figure 11.6 Timing of Timer Output

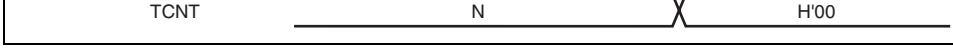


Figure 11.7 Timing of Compare-Match Clear

11.5.5 TCNT External Reset Timing

TCNT is cleared at the rising edge of an external reset input, depending on the settings o
CCLR1 and CCLR0 bits in TCR. The width of the clearing pulse must be at least 1.5 sta
11.8 shows the timing of this operation.

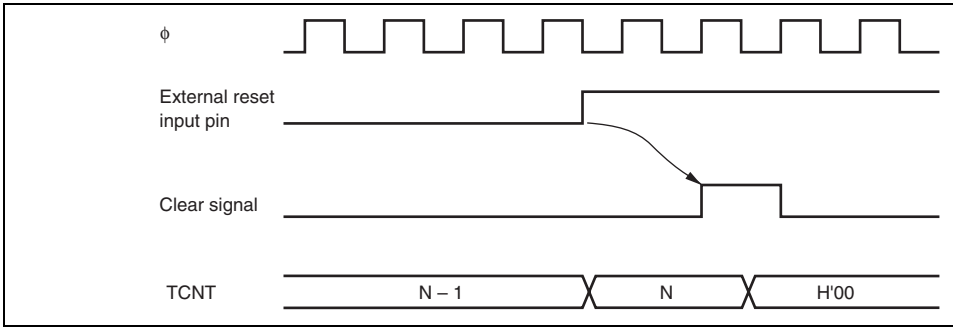


Figure 11.8 Timing of Clearing by External Reset Input

Figure 11.9 Timing of OVF Setting

When the CKS2 to CKS0 bits in TCR_0 are set to B'100, the timer functions as a single
timer with channel 0 occupying the upper 8 bits and channel 1 occupying the lower 8 bi

- Setting of compare-match flags
  - The CMF flag in TCSR_0 is set to 1 when a 16-bit compare-match occurs.
  - The CMF flag in TCSR_1 is set to 1 when a lower 8-bit compare-match occurs.
- Counter clear specification
  - If the CCLR1 and CCLR0 bits in TCR_0 have been set for counter clear at comp
    the 16-bit counter (TCNT_0 and TCNT_1 together) is cleared when a 16-bit com
    match occurs. The 16-bit counter (TCNT_0 and TCNT_1 together) is cleared eve
    counter clear by the TMRI01 pin has also been set.
  - The settings of the CCLR1 and CCLR0 bits in TCR_1 are ignored. The lower 8 b
    be cleared independently.
- Pin output
  - Control of output from the TMO0 pin by the OS3 to OS0 bits in TCSR_0 is in ac
    with the 16-bit compare-match conditions.
  - Control of output from the TMO1 pin by the OS3 to OS0 bits in TCSR_1 is in ac
    with the lower 8-bit compare-match conditions.

11.6.2 Compare-Match Count Mode

When the CKS2 to CKS0 bits in TCR_1 are B'100, TCNT_1 counts compare-match A f
0. Channels 0 and 1 are controlled independently. Conditions such as setting of the CMF
generation of interrupts, output from the TMO pin, and counter clearing are in accordan
settings for each channel.

Interrupt source	Description	Flag	DTC Activation*	Interrup Priority
CMIA0	TCORA_0 compare-match	CMFA	Possible	High
CMIB0	TCORB_0 compare-match	CMFB	Possible	
OVI0	TCNT_0 overflow	OVF	Not possible	
CMIA1	TCORA_1 compare-match	CMFA	Possible	
CMIB1	TCORB_1 compare-match	CMFB	Possible	
OVI1	TCNT_1 overflow	OVF	Not possible	
CMIA2	TCORA_2 compare-match	CMFA	Possible	
CMIB2	TCORB_2 compare-match	CMFB	Possible	
OVI2	TCNT_2 overflow	OVF	Not possible	
CMIA3	TCORA_3 compare-match	CMFA	Possible	
CMIB3	TCORB_3 compare-match	CMFB	Possible	
OVI3	TCNT_3 overflow	OVF	Not possible	Low

Note: This list shows the initial state directly after the reset. Relative channel priorities ca changed by the interrupt controller.

11.7.2 A/D Converter Activation

The A/D converter can be activated only by channel 0 compare match A.

If the ADTE bit in TCSR0 is set to 1 when the CMFA flag is set to 1 by the occurrence o 0 compare match A, a request to start A/D conversion is sent to the A/D converter. If the timer conversion start trigger has been selected on the A/D converter side at this time, A/ conversion is started.

If a timer counter clock pulse is generated during the T_2 state of a TCNT write cycle, the
takes priority, so that the counter is cleared and the write is not performed. Figure 11.10
this operation.

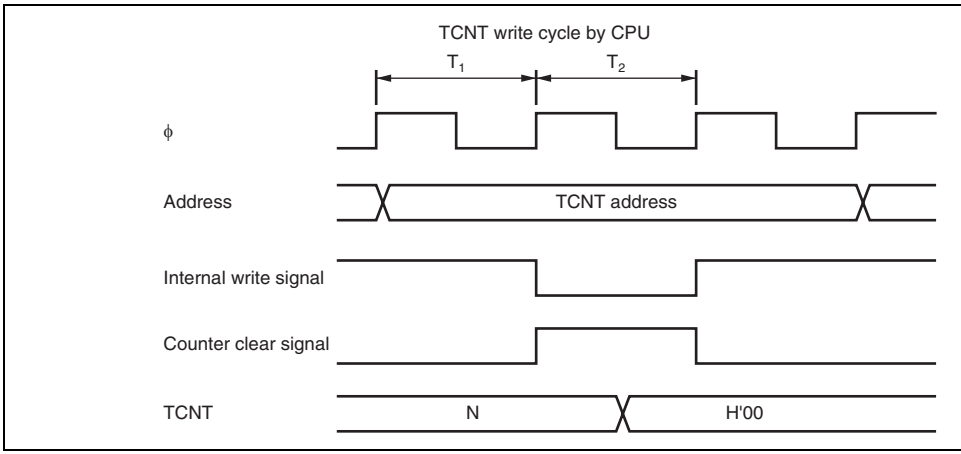


Figure 11.10 Contention between TCNT Write and Clear

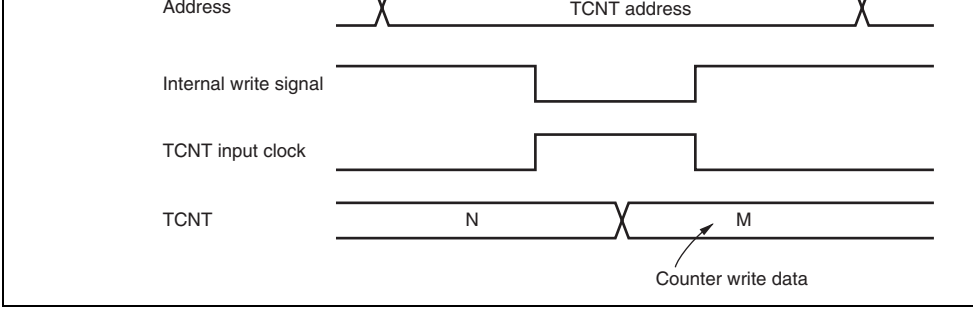


Figure 11.11 Contention between TCNT Write and Increment

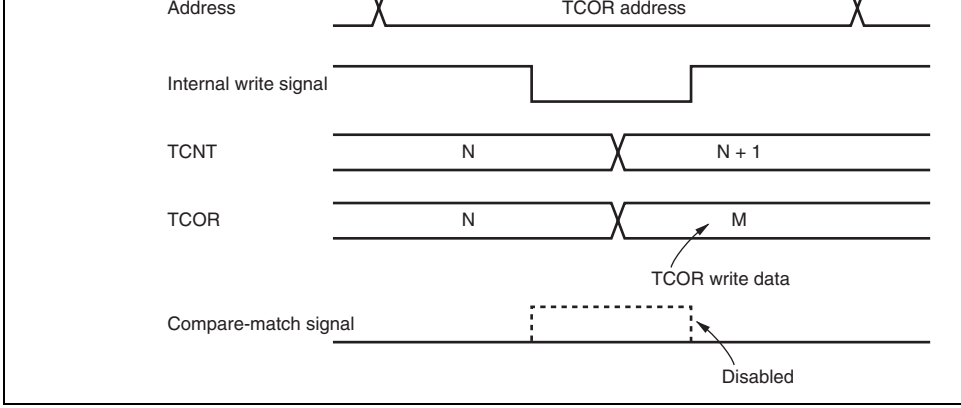


Figure 11.12 Contention between TCOR Write and Compare-Match

11.8.5 Contention between Compare-Matches A and B

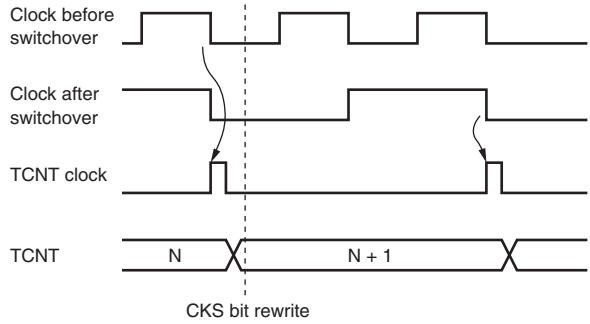
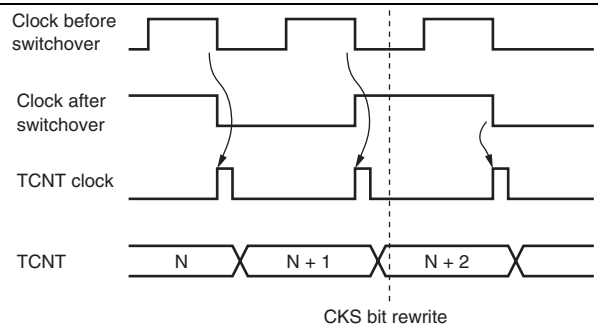
If compare-matches A and B occur at the same time, the 8-bit timer operates in accordan
the priorities for the output states set for compare-match A and compare-match B, as sho
table 11.3.

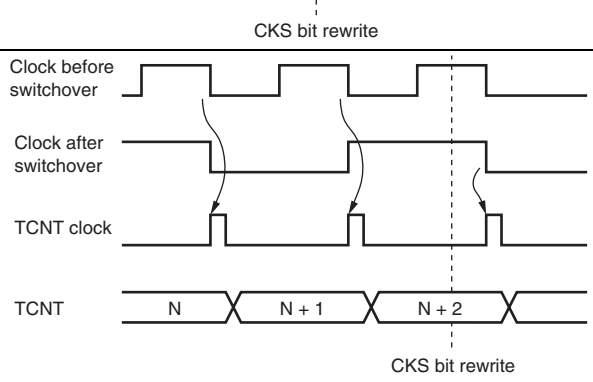
Table 11.3 Timer Output Priorities

Output Setting	Prio
Toggle output	High
1 output	
0 output	
No change	Low

Erroneous incrementation can also happen when switching between internal and external

Table 11.4 Switching of Internal Clock and TCNT Operation

No.	Timing of Switchover by Means of CKS1 and CKS0 Bits	TCNT Clock Operation
1	Switching from low to low*[1]	
2	Switching from low to high*[2]	

		
4	Switching from high to high	

Notes: 1. Includes switching from low to stop, and from stop to low.
2. Includes switching from stop to high.
3. Includes switching from high to stop.
4. Generated on the assumption that the switchover is a falling edge; TCNT is incremented.

operating. This mode should not be set.

12.1 Features

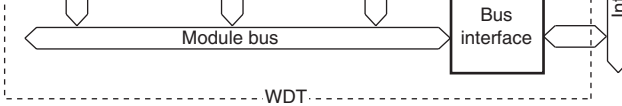
- Selectable from eight counter input clocks for WDT_0
  Selectable from 16 counter input clocks for WDT_1
- Switchable between watchdog timer mode and interval timer mode

Watchdog timer mode

- If the counter in WDT_0 overflows, it is possible to select whether this LSI is interna
  or not
- Power-on reset and manual reset are selectable for internal reset
- If the counter in WDT_1 overflows, it is possible to select whether this LSI is interna
  at a power-on timing or the internal NMI interrupt is generated

Interval timer mode

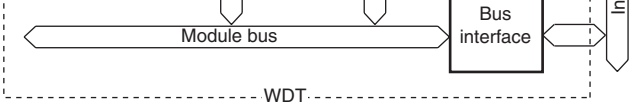
- If the counter overflows, the WDT generates an interval timer interrupt (WOVI)
- Selected clock can be output from BUZZ output pin (WDT_1)



Legend:
TCSR_0: Timer control/status register 0
TCNT_0: Timer counter 0
RSTCSR: Reset control/status register

Note: * The internal reset signal is generated by the register setting.
Power-on reset and manual reset are selectable for internal reset.

Figure 12.1 Block Diagram of WDT_0 (1)



Legend:
TCSR_1: Timer control/status register 1
TCNT_1: Timer counter 1

Note: * The internal reset signal is generated by the register setting.
Power-on reset is set for internal reset.

Figure 12.1 Block Diagram of WDT_1 (2)

12.2 Input/Output Pin

Table 12.1 shows the WDT pin.

Table 12.1 Pin Configuration

Name	Symbol	Input/Output	Function
Buzz output	BUZZ	Output	Clock output selected at WD

- Reset control/status register (RSTCSR)

12.3.1 Timer Counter (TCNT)

TCNT is an 8-bit readable/writable up-counter. TCNT is initialized to H'00 when the TM
TCSR is cleared to 0.

To initialize TCNT to H’00 during timer operation, write a value of H'00 directly to TCN
details, see 12.6.7, Initialization of TCNT by the TME Bit.

12.3.2 Timer Control/Status Register

TCSR functions include selecting the clock source to be input to TCNT and the timer mo

- TCSR_0

Bit	Bit Name	Initial Value	R/W	Description
7	OVF	0	R/(W)*[1]	Overflow Flag Indicates that TCNT has overflowed. Only a 0 can written to this bit, to clear the flag. [Setting condition] When TCNT overflows (changes from H'FF to H'0 However, when internal reset request generation i selected in watchdog timer mode, OVF is cleared automatically by the internal reset. [Clearing condition] Cleared by reading TCSR*[2] when OVF = 1, then w to OVF

				bit is cleared, TCNT stops counting and is initializ H'00.
4, 3	—	All 1	—	Reserved These bits are always read as 1 and cannot be m
2	CKS2	0	R/W	Clock Select 0 to 2
1	CKS1	0	R/W	These bits select the clock source to be input to T
0	CKS0	0	R/W	The overflow frequency*[3] for ϕ = 20 MHz is enclo parentheses. 000: Clock ϕ/2 (frequency: 25.6 μs) 001: Clock ϕ/64 (frequency: 819.2 μs) 010: Clock ϕ/128 (frequency: 1.6 ms) 011: Clock ϕ/512 (frequency: 6.6 ms) 100: Clock ϕ/2048 (frequency: 26.2 ms) 101: Clock ϕ/8192 (frequency: 104.9 ms) 110: Clock ϕ/32768 (frequency: 419.4 ms) 111: Clock ϕ/131072 (frequency: 1.68 s)

Notes: 1. Only 0 can be written for flag clearing.
2. When the OVF flag is polled with the interval timer interrupt disabled, read the while it is 1 at least twice.
3. The overflow period is the time from when TCNT starts counting up from H'00 overflow occurs.

				[Clearing condition] Cleared by reading TCSR*2 when OVF = 1, then to OVF
6	WT/$\overline{IT}$	0	R/W	Timer Mode Select Selects whether the WDT is used as a watchdog interval timer. 0: Interval timer mode (the interval timer interrupt request to the CPU) 1: Watchdog timer mode (a power-on reset or the interrupt request to the CPU)
5	TME	0	R/W	Timer Enable When this bit is set to 1, TCNT starts counting. W bit is cleared, TCNT stops counting and is initiali H'00.

				When SLEEP instruction is executed in high mode or medium-speed mode, transition to s mode, software standby mode, or watch mo made. Note: * When transition is made to watch mode sure that high-speed mode is set.
3	RST/$\overline{\text{NMI}}$	0	R/W	Reset or NMI (REST/$\overline{\text{NMI}}$) Selects either a power-on reset or the NMI inter request when TCNT overflows in watchdog tim 0: NMI interrupt is requested. 1: Power-on reset is requested.

011: Clock ϕ/512 (frequency: 6.6 ms)

100: Clock ϕ/2048 (frequency: 26.2 ms)

101: Clock ϕ/8192 (frequency: 104.9 ms)

110: Clock ϕ/32768 (frequency: 419.4 ms)

111: Clock ϕ/131072 (frequency: 1.68 s)

When PSS = 1:

000: Clock ϕ_{SUB}/2 (frequency: 15.6 ms)

001: Clock ϕ_{SUB}/4 (frequency: 31.3 ms)

010: Clock ϕ_{SUB}/8 (frequency: 62.5 ms)

011: Clock ϕ_{SUB}/16 (frequency: 125 ms)

100: Clock ϕ_{SUB}/32 (frequency: 250 ms)

101: Clock ϕ_{SUB}/64 (frequency: 500 ms)

110: Clock ϕ_{SUB}/128 (frequency: 1 s)

111: Clock ϕ_{SUB}/256 (frequency: 2 s)

Notes: 1. Only 0 can be written, for flag clearing.

2. When the OVF flag is polled with the interval timer interrupt disabled, read the while it is 1 at least twice.

3. The overflow period is the time from when TCNT starts counting up from H'00 overflow occurs.

				only 0 can be written, to clear the flag. [Setting condition] Set when TCNT overflows (changed from H'F in watchdog timer mode [Clearing condition] Cleared by reading RSTCSR when WOVF = then writing 0 to WOVF
6	RSTE	0	R/W	Reset Enable Specifies whether or not a reset signal is gen the chip if TCNT overflows during watchdog t operation. 0: Reset signal is not generated even if TCNT overflows. (Though this LSI is not reset, TC TCSR in WDT are reset.) 1: Reset signal is generated if TCNT overflow
5	RSTS	0	R/W	Reset Select Selects the type of internal reset, which is ger TCNT overflows during watchdog timer opera 0: Power-on reset 1: Manual reset
4 to 0	—	All 1	—	Reserved These bits are always read as 1 and cannot b modified.

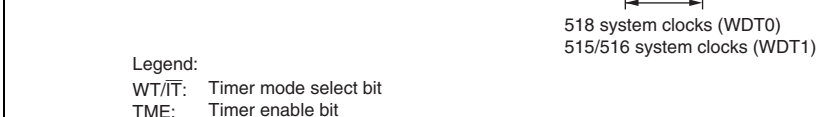
Note: * Only 0 can be written, to clear the flag.

and if TCNT overflows without being rewritten because of a system malfunction or other internal reset signal for this LSI is output for 518 system clocks.

When the RST/$\overline{\text{NMI}}$ bit in TCSR of WDT_1 is set to 1, and if TCNT overflows, the inter signal is output for 516 system clocks. When the RST/ $\overline{\text{NMI}}$ bit is cleared to 0, if TCNT overflows, an NMI interrupt request is generated (for 515 or 516 system clocks when the source is set to ϕ_{SUB} (PSS = 1)).

An internal reset request from the watchdog timer and a reset input from the $\overline{\text{RES}}$ pin are treated as having the same vector. If a WDT internal reset request and the $\overline{\text{RES}}$ pin reset o the same time, the $\overline{\text{RES}}$ pin reset has priority and the WOVF bit in RSTCSR is cleared to

An NMI interrupt request from the watchdog timer and an interrupt request from the NM both treated as having the same vector. So, avoid handling an NMI interrupt request from watchdog timer and an interrupt request from the NMI pin at the same time.



Legend:
WT/$\overline{IT}$: Timer mode select bit
TME: Timer enable bit
WOVF: Overflow flag

Note: * In the case of WDT_0, the internal reset signal is generated only when the RSTE bit is set to 1
In the case of WDT_1, either the internal reset or the NMI interrupt is generated.

Figure 12.2 Watchdog Timer Mode Operation

12.4.2 Interval Timer Mode

To use the WDT as an internal timer, set the WT/$\overline{IT}$ bit in TCSR to 0 and the TME bit to

When the WDT is used as an interval timer, an interval timer interrupt (WOVI) is gener
time the TCNT overflows. (The NMI interrupt request is not generated.) Therefore, an i
can be generated at specified times.

Legend:
WOVI: Interval timer interrupt request generation

Figure 12.3 Interval Timer Mode Operation

12.4.3 Timing of Setting Overflow Flag (OVF)

The OVF bit in TCSR is set to 1 if TCNT overflows during interval timer operation. At th
time, an interval timer interrupt (WOVI) is requested. This timing is shown in figure 12.4

In the case of WDT_1, when an NMI request is selected in watchdog timer mode, if TCN
overflows, the OVF bit in TCSR is set to 1 and an NMI interrupt is requested simultaneou

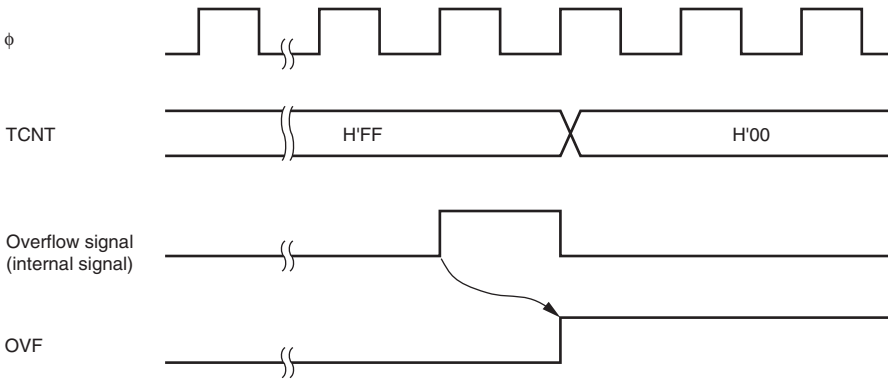


Figure 12.4 Timing of OVF Setting

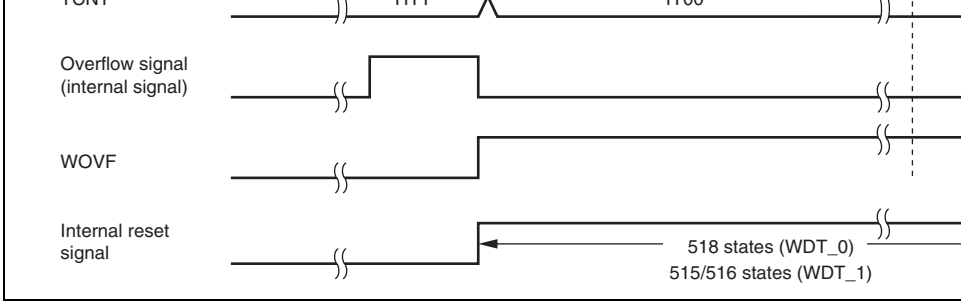


Figure 12.5 Timing of WOVF Setting

12.5 Interrupt Sources

During interval timer mode operation, an overflow generates an interval timer interrupt
The interval timer interrupt is requested whenever the OVF flag in TCSR is set to 1. OV
cleared to 0 in the interrupt handling routine.

If an NMI request has been chosen in watchdog timer mode, an NMI request is generate
TCNT overflow occurs.

Table 12.2 WDT Interrupt Source

Name	Interrupt Source	Interrupt Fla
WOVI	TCNT overflow (interval timer mode)	OVF
NMI	TCNT overflow (watchdog timer mode)	OVF

to TCNT, the upper byte must be H'5A and the lower byte must be data to be written to. W
writing to TCSR, the upper byte must be H'A5 and the lower byte must be data to be writ
Accordingly, the lower byte data is written to TCNT or TCSR.

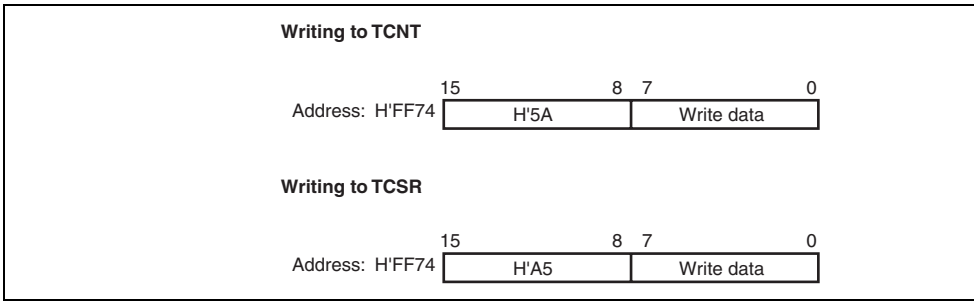


Figure 12.6 Writing to TCNT and TCSR (WDT_0)

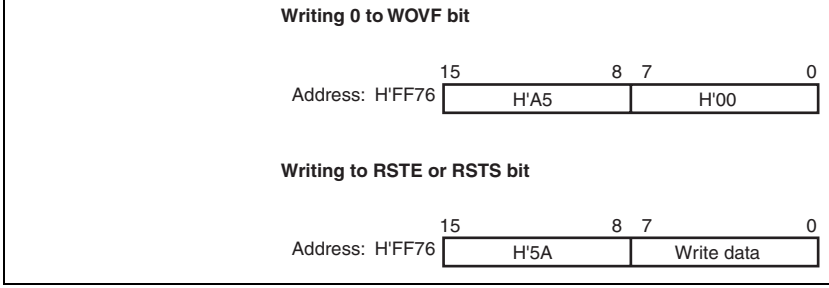


Figure 12.7 Writing to RSTCSR

Reading from TCNT, TCSR and RSTCSR (in the case of WDT_0): These registers a
the same way as other registers. The read addresses are allocated in H'FF74 for TCSR, H
TCNT, and H'FF77 for RSTCSR.

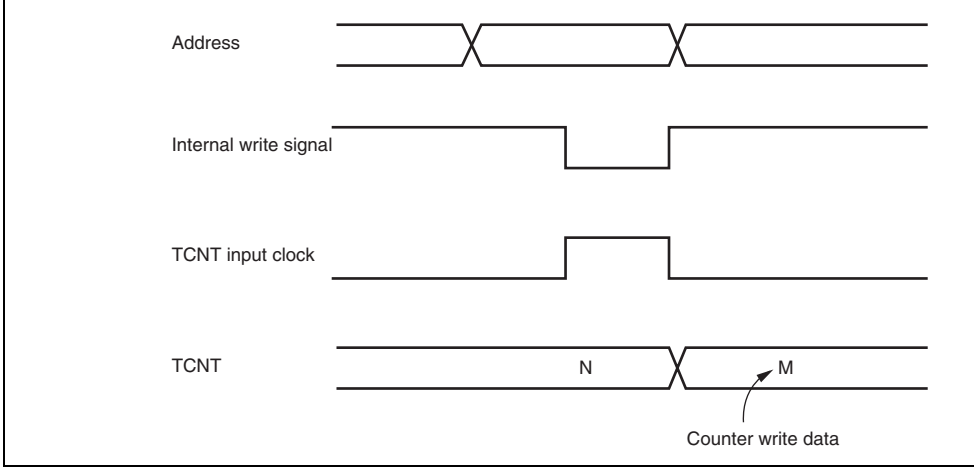


Figure 12.8 Contention between TCNT Write and Increment

12.6.3 Changing Value of PSS or CKS2 to CKS0

If the PSS or CKS0 to CKS2 bits in TCSR are modified while the WDT is operating, erro
occur in the incrementation. Software must be used to stop the watchdog timer (by clearin
TME bit to 0) before changing the value of the PSS or CKS0 to CKS2 bits.

12.6.4 Switching between Watchdog Timer Mode and Interval Timer Mode

If the mode is switched between watchdog timer mode and interval timer mode while the
operating, errors could occur. Software must be used to stop the watchdog timer (by clear
TME bit to 0) before switching the timer mode.

When the OVF flag setting conflicts with the OVF flag reading in interval timer mode, w
to the OVF flag may not clear the flag even though the OVF flag has been read while it
there is a possibility that the OVF flag setting and reading will conflict, such as when th
flag is polled with the interval timer interrupt disabled, read the OVF flag while it is 1 at
twice before writing 0 to the OVF flag to clear the flag.

12.6.7 Initialization of TCNT by the TME Bit

In high-speed or medium-speed mode, after the counter (TCNT) is initialized by clearin
TCSR to 0 while operation is in progress with ϕSUB (subclock) selected as the dividing
(PSS in TCSR set to 1) for the TCNT input clock, TCNT may not initialize properly wh
once again set to 1 to activate TCNT operation. To avoid this problem, TCNT by writing
of H'00 to it directly.

13.1 Features

- Choice of asynchronous or clocked synchronous serial communication mode
  (Because the same pin is used as the clock input/output pin for channel 1 and channe
  clocks cannot be output at the same time.)
- Full-duplex communication capability
  The transmitter and receiver are mutually independent, enabling transmission and re
  be executed simultaneously.
  The double-buffering configuration is adopted in both the transmitter and the receive
  enabling continuous transmission and continuous reception of serial data.
- On-chip baud rate generator allows any bit rate to be selected
  External clock can be selected as a transfer clock source (except in Smart Card interf
- Choice of LSB-first or MSB-first transfer (except in the case of asynchronous mode
- Four interrupt sources
  Transmit-end, transmit-data-empty, receive-data-full, and receive error — that can is
  requests.
  The transmit-data-empty and receive-data-full interrupts can be used to activate the d
  transfer controller (DTC).
- Module stop mode can be set

Asynchronous mode

- Data length: 7 or 8 bits
- Stop bit length: 1 or 2 bits
- Parity: Even, odd, or none
- Receive error detection: Parity, overrun, and framing errors

- Automatic data retransmission when an error signal is received in transmit mode
- Direct convention and inverse convention both supported

Figure 13.1 shows a block diagram of the SCI.

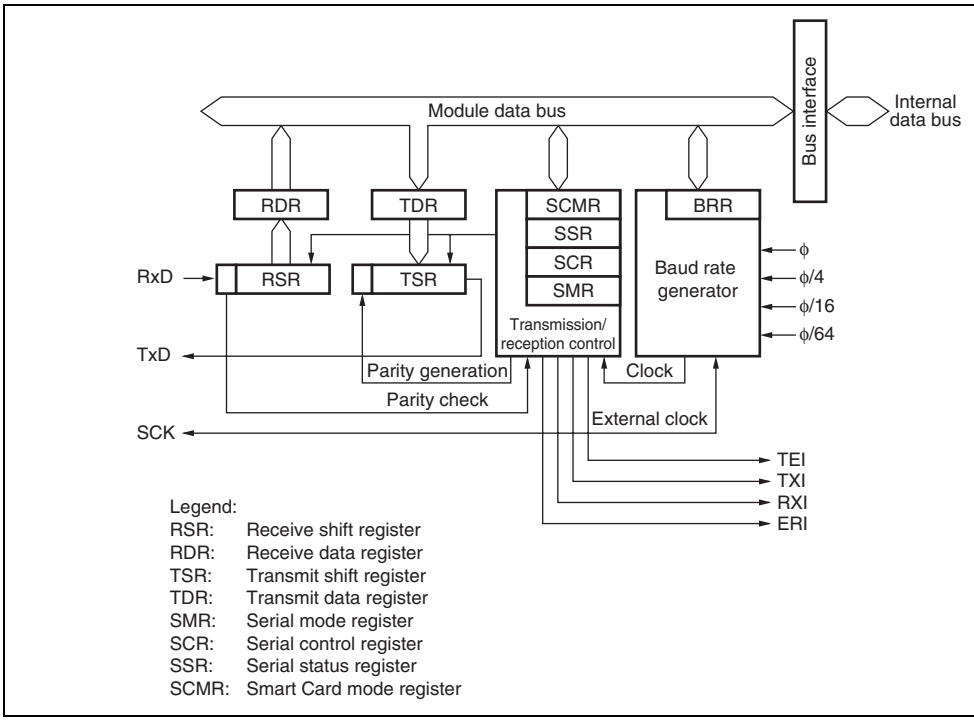


Figure 13.1 Block Diagram of SCI

1	SCK1*[2]	I/O	Clock input/output in channel 1
	RxD1	Input	Receive data input in channel 1
	TxD1	Output	Transmit data output in channel 1
2	SCK2	I/O	Clock input/output in channel 2
	RxD2	Input	Receive data input in channel 2
	TxD2	Output	Transmit data output in channel 2
3	SCK3	I/O	Clock input/output in channel 3
	RxD3	Input	Receive data input in channel 3
	TxD3	Output	Transmit data output in channel 3
4	SCK4*[2]	I/O	Clock input/output in channel 4
	RxD4	Input	Receive data input in channel 4
	TxD4	Output	Transmit data output in channel 4

Notes: 1. Pin names SCK, RxD, and TxD are used in this manual for all channels, omit channel designation.

2. Because SCK1 and SCK4 are allocated to the same pin, these clocks cannot at the same time.

- Transmit data register_0 (TDR_0)
- Transmit shift register_0 (TSR_0)
- Serial mode register_0 (SMR_0)
- Serial control register_0 (SCR_0)
- Serial status register_0 (SSR_0)
- Smart Card mode register_0 (SCMR_0)
- Bit rate register_0 (BRR_0)

Channel 1

- Receive shift register_1 (RSR_1)
- Receive data register_1 (RDR_1)
- Transmit data register_1 (TDR_1)
- Transmit shift register_1 (TSR_1)
- Serial mode register_1 (SMR_1)
- Serial control register_1 (SCR_1)
- Serial status register_1 (SSR_1)
- Smart Card mode register_1 (SCMR_1)
- Bit rate register_1 (BRR_1)

Channel 2

- Receive shift register_2 (RSR_2)
- Receive data register_2 (RDR_2)
- Transmit data register_2 (TDR_2)
- Transmit shift register_2 (TSR_2)
- Serial mode register_2 (SMR_2)
- Serial control register_2 (SCR_2)

- Serial control register_3 (SCR_3)
- Serial status register_3 (SSR_3)
- Smart Card mode register_3 (SCMR_3)
- Bit rate register_3 (BRR_3)

Channel 4

- Receive shift register_4 (RSR_4)
- Receive data register_4 (RDR_4)
- Transmit data register_4 (TDR_4)
- Transmit shift register_4 (TSR_4)
- Serial mode register_4 (SMR_4)
- Serial control register_4 (SCR_4)
- Serial status register_4 (SSR_4)
- Smart Card mode register_4 (SCMR_4)
- Bit rate register_4 (BRR_4)

13.3.1 Receive Shift Register (RSR)

RSR is a shift register that is used to receive serial data input to the RxD pin and conver parallel data. When one byte of data has been received, it is transferred to RDR automat

RSR cannot be directly accessed by the CPU.

13.3.2 Receive Data Register (RDR)

RDR is an 8-bit register that stores receive data. When the SCI has received one byte of data, it transfers the received serial data from RSR to RDR, where it is stored. After this

it transfers the transmit data written in TDR to TSR and starts transmission. As TDR and function as a double buffer in this way, continuous transmit operations are possible. Whe transmits one byte of serial data, if the next transmit data has already been written to TDR transfers the written data to TSR to continue transmission. Although TDR can be read or by the CPU at all times, to achieve reliable serial transmission, write transmit data to TDR once after confirming that the TDRE bit in SSR is set to 1.

TDR is initialized to H'FF by a reset, or in standby mode, watch mode, or module stop m

13.3.4 Transmit Shift Register (TSR)

TSR is a shift register that transmits serial data. To perform serial data transmission, the S transfers transmit data from TDR to TSR, and then sends the data to the TxD pin. TSR ca directly accessed by the CPU.

13.3.5 Serial Mode Register (SMR)

SMR is used to set the SCI's serial transfer format and select the baud rate generator cloc

Some bit functions of SMR differ between normal serial communication interface mode a Card interface mode.

- Normal Serial Communication Interface Mode (When SMIF in SCMR Is 0)

Bit	Bit Name	Initial Value	R/W	Description
7	C/$\overline{A}$	0	R/W	Communication Mode 0: Asynchronous mode 1: Clocked synchronous mode

				data before transmission, and the parity bit is c reception. For a multiprocessor format, parity b and checking are not performed regardless of t setting.
4	O/$\overline{E}$	0	R/W	Parity Mode (enabled only when the PE bit is 1 asynchronous mode) 0: Selects even parity. When even parity is set, parity bit addition is performed in transmission so that the total n bits in the transmit character plus the parity b In reception, a check is performed to see if th number of 1 bits in the receive character plus is even. 1: Selects odd parity. When odd parity is set, parity bit addition is p in transmission so that the total number of 1 transmit character plus the parity bit is odd. I reception, a check is performed to see if the number of 1 bits in the receive character plus parity bit is odd.
3	STOP	0	R/W	Stop Bit Length (enabled only in asynchronous Selects the stop bit length in transmission. 0: 1 stop bit 1: 2 stop bits In reception, only the first stop bit is checked. If second stop bit is 0, it is treated as the start bit next transmit character.

generator.

00: ϕ clock (n = 0)

01: ϕ/4 clock (n = 1)

10: ϕ/16 clock (n = 2)

11: ϕ/64 clock (n = 3)

For the relationship between the bit rate register
and the baud rate, see section 13.3.9, Bit Rate R
(BRR). n is the decimal representation of the val
bit rate register.

				• The TEND flag is generated 12.5 etu (11.5 block transfer mode) after the beginning of bit. • Clock output on/off control only 1: GSM mode operation in smart card interface • The TEND flag is generated 11.0 etu after beginning of the start bit. • In addition to clock output on/off control, hi fixed control is supported (set using SCR).
6	BLK	0	R/W	Setting this bit to 1 allows block transfer mode o For details, see section 13.7.3, Block Transfer I 0: Normal smart card interface mode operation value) • Error signal transmission, detection, and au data retransmission are performed. • The TXI interrupt is generated by the TEND • The TEND flag is set 12.5 etu (11.0 etu in t mode) after transmission starts. 1: Operation in block transfer mode • Error signal transmission, detection, and au data retransmission are not performed. • The TXI interrupt is generated by the TDRI • The TEND flag is set 11.5 etu (11.0 etu in t mode) after transmission starts.

				For details on setting this bit in Smart Card interf mode, see section 13.7.2, Data Format (Except f Transfer Mode).
3	BCP1	0	R/W	Basic Clock Pulse 1 and 0
2	BCP0	0	R/W	These bits specify the number of basic clock per 1-bit transfer interval on the Smart Card interface 00: 32 clock (S = 32) 01: 64 clock (S = 64) 10: 372 clock (S = 372) 11: 256 clock (S = 256) For details, see section 13.7.4, Receive Data Sa Timing and Reception Margin. S stands for the v in bit rate register.
1	CKS1	0	R/W	Clock Select 1 and 0
0	CKS0	0	R/W	These bits select the clock source for the baud r generator. 00: ϕ clock (n = 0) 01: ϕ/4 clock (n = 1) 10: ϕ/16 clock (n = 2) 11: ϕ/64 clock (n = 3) For the relationship between the bit rate register and the baud rate, see section 13.3.9, Bit Rate R (BRR). n is the decimal representation of the val BRR.

Note: etu: Elementary time unit (time for transfer of 1 bit)

7	TIE	0	R/W	Transmit Interrupt Enable When this bit is set to 1, the TXI interrupt reque enabled. TXI interrupt request cancellation can be perfor reading 1 from the TDRE flag in SSR, then clea 0, or clearing the TIE bit to 0.
6	RIE	0	R/W	Receive Interrupt Enable When this bit is set to 1, RXI and ERI interrupt are enabled. RXI and ERI interrupt request cancellation can performed by reading 1 from the RDRF, FER, P ORER flag in SSR, then clearing the flag to 0, c the RIE bit to 0.
5	TE	0	R/W	Transmit Enable When this bit is set to 1, transmission is enable In this state, serial transmission is started when data is written to TDR and the TDRE flag in SS cleared to 0. SMR setting must be performed to decide the t format before setting the TE bit to 1. When this cleared to 0, the transmission operation is disal the TDRE flag is fixed at 1.

3	MPIE	0	R/W	Multiprocessor Interrupt Enable (enabled only wh MP bit in SMR is 1 in asynchronous mode) When this bit is set to 1, receive data in which th multiprocessor bit is 0 is skipped, and setting of t RDRF, FER, and ORER status flags in SSR is p On receiving data in which the multiprocessor bi bit is automatically cleared and normal reception resumed. For details, see section 13.5, Multiproc Communication Function. When receive data including MPB = 0 is received receive data transfer from RSR to RDR, receive detection, and setting of the RERF, FER, and OR flags in SSR, are not performed. When receive data including MPB = 1 is received MPB bit in SSR is set to 1, the MPIE bit is cleare automatically, and generation of RXI and ERI int (when the TIE and RIE bits in SCR are set to 1) and ORER flag setting are enabled.
2	TEIE	0	R/W	Transmit End Interrupt Enable When this bit is set to 1, TEI interrupt request is TEI cancellation can be performed by reading 1 TDRE flag in SSR, then clearing it to 0 and clear TEND flag to 0, or clearing the TEIE bit to 0.

1X: External clock
Inputs a clock with a frequency 16 times the
from the SCK pin.

Clocked synchronous mode

0X: Internal clock (SCK pin functions as clock o

1X: External clock (SCK pin functions as clock

Legend:

X: Don't care

				are enabled. RXI and ERI interrupt request cancellation can b performed by reading 1 from the RDRF, FER, PE ORER flag in SSR, then clearing the flag to 0, or the RIE bit to 0.
5	TE	0	R/W	Transmit Enable When this bit is set to 1, transmission is enabled In this state, serial transmission is started when t data is written to TDR and the TDRE flag in SSR cleared to 0. SMR setting must be performed to decide the tra format before setting the TE bit to 1. When this b cleared to 0, the transmission operation is disabl the TDRE flag is fixed at 1.
4	RE	0	R/W	Receive Enable When this bit is set to 1, reception is enabled. Serial reception is started in this state when a sta detected in asynchronous mode or serial clock in detected in clocked synchronous mode. SMR setting must be performed to decide the re format before setting the RE bit to 1. Clearing the RE bit to 0 does not affect the RDR PER, and ORER flags, which retain their states.

				(when the TIE and RIE bits in SCR are set to 1) and ORER flag setting are enabled.
2	TEIE	0	R/W	Transmit End Interrupt Enable Write 0 to this bit in Smart Card interface mode TEI cancellation can be performed by reading 1 TDRE flag in SSR, then clearing it to 0 and clea TEND flag to 0, or clearing the TEIE bit to 0.
1	CKE1	0	R/W	Clock Enable 1 and 0
0	CKE0	0	R/W	Enable or disable clock output from the SCK pi clock output can be dynamically switched in GS For details, see section 13.7.8, Clock Output C When the GM bit in SMR is 0: 00: Output disabled (SCK pin can be used as a pin) 01: Clock output 1X: Reserved When the GM bit in SMR is 1: 00: Output fixed low 01: Clock output 10: Output fixed high 11: Clock output

Legend:

X: Don't care

7	TDRE	1	R/(W)*	Transmit Data Register Empty Displays whether TDR contains transmit data. [Setting conditions] • When the TE bit in SCR is 0 • When data is transferred from TDR to TSR a can be written to TDR [Clearing conditions] • When 0 is written to TDRE after reading TDF • When the DTC*[2] is activated by a TXI interru request and writes data to TDR
6	RDRF	0	R/(W)*[1]	Receive Data Register Full Indicates that the received data is stored in RDR [Setting condition] • When serial reception ends normally and rec is transferred from RSR to RDR [Clearing conditions] • When 0 is written to RDRF after reading RDF • When the DTC*[2] is activated by an RXI interr transfers data from RDR The RDRF flag is not affected and retains their p values when the RE bit in SCR is cleared to 0. If reception of the next data is completed while th flag is still set to 1, an overrun error will occur an receive data will be lost.

				the ORER flag is set to 1. In clocked synchronc serial transmission cannot be continued either. [Clearing condition] • When 0 is written to ORER after reading OR The ORER flag is not affected and retains its pr state when the RE bit in SCR is cleared to 0.
4	FER	0	R/(W)*1	Framing Error Indicates that a framing error occurred during re asynchronous mode, causing abnormal termina [Setting condition] • When the stop bit is 0 In 2 stop bit mode, only the first stop bit is chec value to 1; the second stop bit is not checked. I framing error occurs, the receive data is transfe RDR but the RDRF flag is not set. Also, subsec serial reception cannot be continued while the F set to 1. In clocked synchronous mode, serial transmission cannot be continued, either. [Clearing condition] • When 0 is written to FER after reading FER In 2-stop-bit mode, only the first stop bit is chec The FER flag is not affected and retains its prev when the RE bit in SCR is cleared to 0.

				serial reception cannot be continued while the F is set to 1. In clocked synchronous mode, serial transmission cannot be continued, either. [Clearing condition] • When 0 is written to PER after reading PER = The PER flag is not affected and retains its previo when the RE bit in SCR is cleared to 0.
2	TEND	1	R	Transmit End Indicates that transmission has been ended. [Setting conditions] • When the TE bit in SCR is 0 • When TDRE = 1 at transmission of the last b byte serial transmit character [Clearing conditions] • When 0 is written to TDRE after reading TDR • When the DTC*[2] is activated by a TXI interru request and transfers transmit data to TDR
1	MPB	0	R	Multiprocessor Bit MPB stores the multiprocessor bit in the receive When the RE bit in SCR is cleared to 0 its previo is retained.
0	MPBT	0	R/W	Multiprocessor Bit Transfer MPBT stores the multiprocessor bit to be added transmit data.

Notes: 1. Only a 0 can be written to this bit to clear the flag.
2. This bit is cleared by DTC only when DISEL is 0 with the transfer counter othe
3. To clear the flag by using the CPU, write 0 to the flag and then read it once ag

				• When 0 is written to TDRE after reading TD • When the DTC*2 is activated by a TXI interr request and writes data to TDR
6	RDRF	0	R/(W)*1	Receive Data Register Full Indicates that the received data is stored in RD [Setting condition] • When serial reception ends normally and re is transferred from RSR to RDR [Clearing conditions] • When 0 is written to RDRF after reading RD • When the DTC*2 is activated by an RXI inte transfers data from RDR The RDRF flag is not affected and retains their values when the RE bit in SCR is cleared to 0. If reception of the next data is completed while flag is still set to 1, an overrun error will occur a receive data will be lost.

				flag is set to 1. In clocked synchronous mode, se transmission cannot be continued, either. [Clearing condition] • When 0 is written to ORER after reading ORI The ORER flag is not affected and retains its pre state when the RE bit in SCR is cleared to 0.
4	ERS	0	R/(W)*1	Error Signal Status Indicates that the status of an error, signal 1 retu from the reception side at reception [Setting condition] • When the low level of the error signal is samp [Clearing condition] • When 0 is written to ERS after reading ERS = The ERS flag is not affected and retains its previ when the TE bit in SCR is cleared to 0.

set to 1. In clocked synchronous mode, serial transmission cannot be continued, either.

[Clearing condition]

- When 0 is written to PER after reading PER

The PER flag is not affected and retains its prev when the RE bit in SCR is cleared to 0.

				data. The timing of bit setting differs according to the r setting as follows: When GM = 0 and BLK = 0, 12.5 etu after transm starts When GM = 0 and BLK = 1, 11.5 etu after transm starts When GM = 1 and BLK = 0, 11.0 etu after transm starts When GM = 1 and BLK = 1, 11.0 etu after transm starts [Clearing conditions] • When 0 is written to TDRE after reading TDR • When the DTC*[2] is activated by a TXI interru transfers transmit data to TDR
1	MPB	0	R	Multiprocessor Bit This bit is not used in Smart Card interface mode
0	MPBT	0	R/W	Multiprocessor Bit Transfer Write 0 to this bit in Smart Card interface mode.

Notes: etu: Elementary time unit (time for transfer of 1 bit)

1. Only 0 can be written to this bit, to clear the flag.
2. This bit is cleared by DTC only when DISEL is 0 with the transfer counter othe
3. To clear the flag by using the CPU, write 0 to the flag and then read it once ag

				0: LSB-first in transfer 1: MSB-first in transfer The bit setting is valid only when the transfer da is 8 bits. For 7-bit data, LSB-first is fixed.
2	SINV	0	R/W	Smart Card Data Invert Specifies inversion of the data logic level. The S does not affect the logic level of the parity bit. T the parity bit, invert the O/$\overline{E}$ bit in SMR. 0: TDR contents are transmitted as they are. R data is stored as it is in RDR 1: TDR contents are inverted before being trans Receive data is stored in inverted form in RD
1	—	1	—	Reserved This bit is always read as 1, and cannot be mod
0	SMIF	0	R/W	Smart Card Interface Mode Select This bit is set to 1 to make the SCI operate in S interface mode. 0: Normal asynchronous mode or clocked sync mode 1: Smart Card interface mode

Asynchronous Mode	$B = \frac{\phi \times 10^6}{64 \times 2^{2n-1} \times (N + 1)}$	Error (%) = { $\frac{\phi \times 10^6}{B \times 64 \times 2^{2n-1} \times (N + 1)}$ -1
Clocked Synchronous Mode	$B = \frac{\phi \times 10^6}{8 \times 2^{2n-1} \times (N + 1)}$	—
Smart Card Interface Mode	$B = \frac{\phi \times 10^6}{S \times 2^{2n+1} \times (N + 1)}$	Error (%) = { $\frac{\phi \times 10^6}{B \times S \times 2^{2n+1} \times (N + 1)}$ -1

Legend:

B: Bit rate (bit/s)

N: BRR setting for baud rate generator ($0 \leq N \leq 255$)

ϕ: Operating frequency (MHz)

n and S: Determined by the SMR settings shown in the following tables.

SMR Setting		Clock	
CKS1	**CKS0**	**Source**	**n**
0	0	ϕ	0
0	1	ϕ/4	1
1	0	ϕ/16	2
1	1	ϕ/64	3

SMR Setting		
BCP1	**BCP0**	**S**
0	0	32
0	1	64
1	0	372
1	1	256

Table 13.3 shows sample N settings in BRR in normal asynchronous mode. Table 13.4 sh
maximum bit rate for each frequency in normal asynchronous mode. Table 13.6 shows sa
settings in BRR in clocked synchronous mode. Table 13.8 shows sample N settings in BR
Smart Card interface mode. In Smart Card interface mode, S (the number of basic clock p
a 1-bit transfer interval) can be selected. For details, see section 13.7.4, Receive Data Sam
Timing and Reception Margin. Tables 13.5 and 13.7 show the maximum bit rates with ex
clock input.

1200	0	207	0.16	0	255	0.00	1	64	0.16	1	77	0.16	1	79
2400	0	103	0.16	0	127	0.00	0	129	0.16	0	155	0.16	0	15
4800	0	51	0.16	0	63	0.00	0	64	0.16	0	77	0.16	0	79
9600	0	25	0.16	0	31	0.00	0	32	–1.36	0	38	0.16	0	39
19200	0	12	0.16	0	15	0.00	0	15	1.73	0	19	–2.34	0	19
31250	0	7	0.00	0	9	–1.70	0	9	0.00	0	11	0.00	0	11
38400	—	—	—	0	7	0.00	0	7	1.73	0	9	–2.34	0	9

	Operating Frequency ϕ (MHz)										
	14			14.7456			16			17.2	
Bit Rate (bps)	n	N	Error (%)	n	N	Error (%)	n	N	Error (%)	n	N
110	2	248	–0.17	3	64	0.70	3	70	0.03	3	75
150	2	181	0.16	2	191	0.00	2	207	0.16	2	223
300	2	90	0.16	2	95	0.00	2	103	0.16	2	111
600	1	181	0.16	1	191	0.00	1	207	0.16	1	223
1200	1	90	0.16	1	95	0.00	1	103	0.16	1	111
2400	0	181	0.16	0	191	0.00	0	207	0.16	0	223
4800	0	90	0.16	0	95	0.00	0	103	0.16	0	111
9600	0	45	–0.93	0	47	0.00	0	51	0.16	0	55
19200	0	22	–0.93	0	23	0.00	0	25	0.16	0	27
31250	0	13	0.00	0	14	–1.70	0	15	0.00	0	16
38400	—	—	—	0	11	0.00	0	12	0.16	0	13

2400	0	233	0.16	0	255	0.00	1	64	0.16	1	80
4800	0	116	0.16	0	127	0.00	0	129	0.16	0	162
9600	0	58	–0.69	0	63	0.00	0	64	0.16	0	80
19200	0	28	1.02	0	31	0.00	0	32	–1.36	0	40
31250	0	17	0.00	0	19	–1.70	0	19	0.00	0	24
38400	0	14	–2.34	0	15	0.00	0	15	1.73	0	19

Table 13.4 Maximum Bit Rate for Each Frequency (Asynchronous Mode)

ϕ (MHz)	Maximum Bit Rate (bit/s)	n	N
8	250000	0	0
9.8304	307200	0	0
10	312500	0	0
12	375000	0	0
12.288	384000	0	0
14	437500	0	0
14.7456	460800	0	0
16	500000	0	0
17.2032	537600	0	0
18	562500	0	0
19.6608	614400	0	0
20	625000	0	0
25	781250	0	0

14.7456	3.6864	230400
16	4.0000	250000
17.2032	4.3008	268800
18	4.5000	281250
19.6608	4.9152	307200
20	5.0000	312500
25	6.2500	390625

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

Legend:

Blank: Cannot be set.

—: Can be set, but there will be a degree of error.

*: Continuous transfer is not possible.

25	4.1667	4166666.7

Table 13.8 Examples of Bit Rate for Various BRR Settings (Smart Card Interface (When n = 0 and S = 372)

Bit Rate (bps)	Operating Frequency ϕ (MHz)							
	10.00		10.7136		13.00		14.2	
	N	Error (%)	N	Error (%)	N	Error (%)	N	E
9600	1	30.00	1	25.00	1	8.99	1	0

Bit Rate (bps)	Operating Frequency ϕ (MHz)							
	16.00		18.00		20.00		25	
	N	Error (%)	N	Error (%)	N	Error (%)	N	E
9600	1	12.01	2	15.99	2	6.66	3	1

20.00	26882	0	0
25.00	33602	0	0

synchronization at the falling edge of the start bit in reception. The SCI samples the data
8th pulse of a clock with a frequency of 16 times the length of one bit, so that the transfe
latched at the center of each bit.

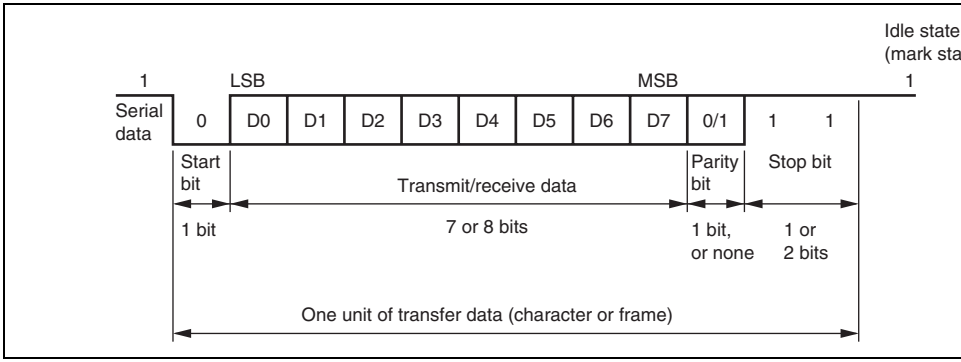
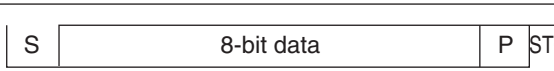
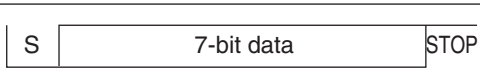
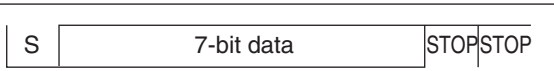
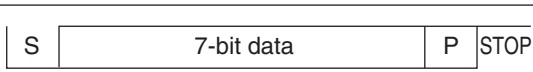
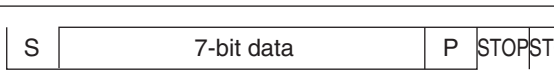
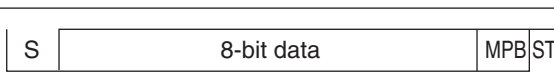
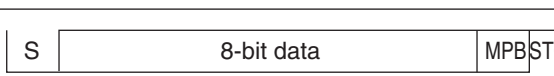
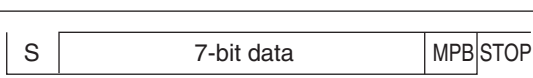
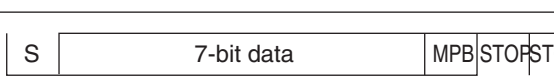


Figure 13.2 Data Format in Asynchronous Communication (Example with 8-Bit Data, Parity, Two Stop Bits)

13.4.1 Data Transfer Format

Table 13.10 shows the data transfer formats that can be used in asynchronous mode. An
transfer formats can be selected according to the SMR setting. For details on the multipr
bit, see section 13.5, Multiprocessor Communication Function.

0	1	0	1	
1	0	0	0	
1	0	0	1	
1	1	0	0	
1	1	0	1	
0	—	1	0	
0	—	1	1	
1	—	1	0	
1	—	1	1	

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

Where M: Reception margin (%)
N: Ratio of bit rate to clock (N = 16)
D: Clock duty (D = 0 to 1.0)
L: Frame length (L = 9 to 12)
F: Absolute value of clock rate deviation

Assuming values of F (absolute value of clock rate deviation) = 0, D (clock duty) = 0.5, (ratio of bit rate to clock) = 16 in formula (1), the reception margin can be given by the f

$$M = \{0.5 - 1/(2 \times 16)\} \times 100\ [\%] = 46.875\%$$

However, this is only the computed value, and a margin of 20% to 30% should be allow system design.

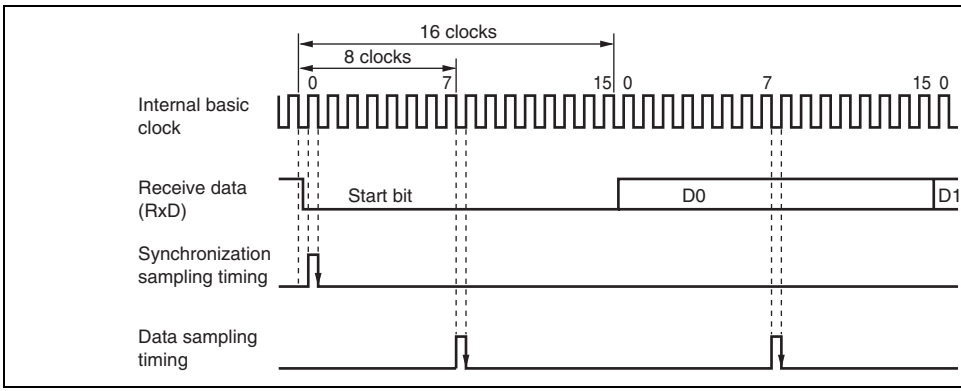


Figure 13.3 Receive Data Sampling Timing in Asynchronous Mode

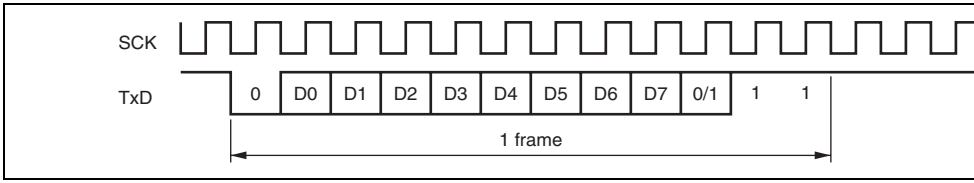
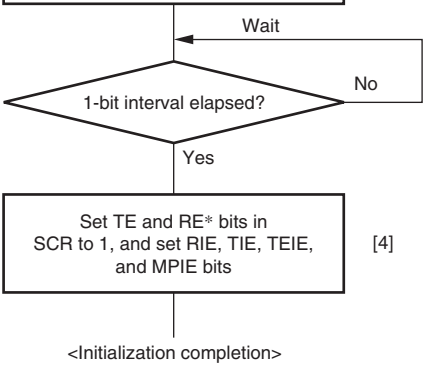


Figure 13.4 Relationship between Output Clock and Transfer Data Phase (Asynchronous Mode)

13.4.4 SCI Initialization (Asynchronous Mode)

Before transmitting and receiving data, you should first clear the TE and RE bits in SCR initialize the SCI as described in figure 13.5. When the operating mode, or transfer forma changed for example, the TE and RE bits must be cleared to 0 before making the change following procedure. When the TE bit is cleared to 0, the TDRE flag is set to 1. Note that the RE bit to 0 does not initialize the contents of the RDRF, PER, FER, and ORER flags, contents of RDR. When the external clock is used in asynchronous mode, the clock must supplied even during initialization.



[4] Wait at least one bit interval, then set the TE bit or RE bit in SCR to 1. Also set the RIE, TIE, TEIE, and MPIE bits.

Setting the TE and RE bits enables the TxD and RxD pins to be used.

Note: * Set the RE bit while the RxD pin is driven 1. When the RE bit is set to 1 while the RxD pin is driven 0, it may be received as the start bit.

Figure 13.5 Sample SCI Initialization Flowchart

3. Data is sent from the TxD pin in the following order: start bit, transmit data, parity bit multiprocessor bit (may be omitted depending on the format), and stop bit.
4. The SCI checks the TDRE flag at the timing for sending the stop bit.
5. If the TDRE flag is 0, the data is transferred from TDR to TSR, the stop bit is sent, an serial transmission of the next frame is started.
6. If the TDRE flag is 1, the TEND flag in SSR is set to 1, the stop bit is sent, and then t state" is entered, in which 1 is output. If the TEIE bit in SCR is set to 1 at this time, a interrupt request is generated.

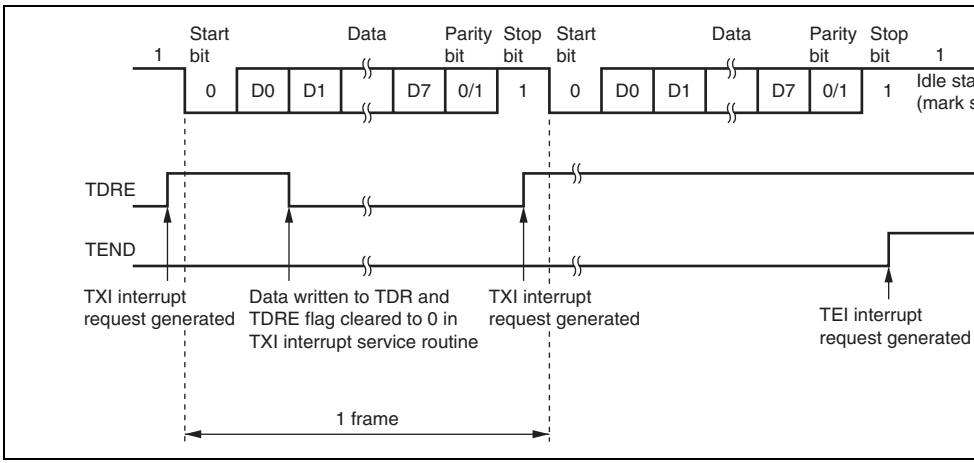
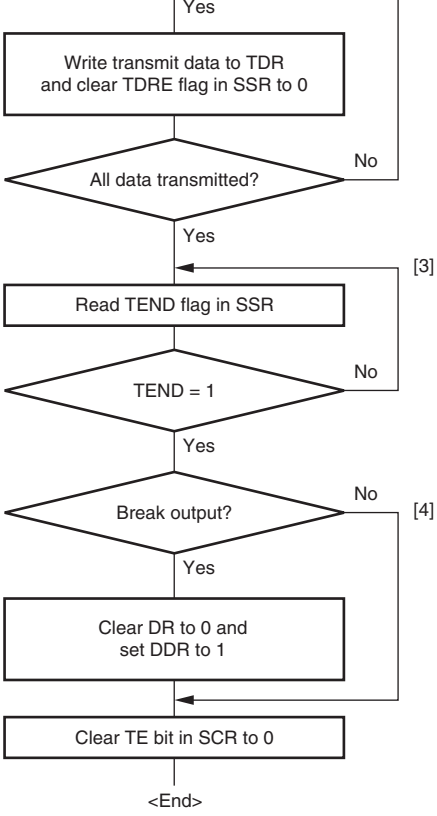


Figure 13.6 Example of Operation in Transmission in Asynchronous Mode (Example with 8-Bit Data, Parity, One Stop Bit)



transmit data to TDR and clear the TDRE flag to 0.

[3] Serial transmission continuation procedure:
To continue serial transmission, read 1 from the TDRE flag to confirm that writing is possible, then write data to TDR, and then clear the TDRE flag to 0. Checking and clearing of the TDRE flag is automatic when the DTC* is activated by a transmit data empty interrupt (TXI) request, and data is written to TDR.

[4] Break output at the end of serial transmission:
To output a break in serial transmission, clear DR for the port corresponding to the TxD pin to 0, set DDR to 1, then clear the TE bit in SCR to 0.

Note: * The case, where the TDRE flag check and
automatically executed by DTC,
occurs only when the DISEL bit in DTC is
with the transfer counter other than 0.
Therefore, when the DISEL bit is 1, or
both the DISEL bit and the transfer counte
give the CPU an instruction to clear the TD

Figure 13.7 Sample Serial Transmission Flowchart

3. If a parity error is detected, the PER bit in SSR is set to 1 and receive data is transferr RDR. If the RIE bit in SCR is set to 1 at this time, an ERI interrupt request is generate
4. If a framing error is detected (when the stop bit is 0), the FER bit in SSR is set to 1 an data is transferred to RDR. If the RIE bit in SCR is set to 1 at this time, an ERI interru request is generated.
5. If reception is completed successfully, the RDRF bit in SSR is set to 1, and receive da transferred to RDR. If the RIE bit in SCR is set to 1 at this time, an RXI interrupt requ generated. Continuous reception is possible because the RXI interrupt routine reads th data transferred to RDR before reception of the next receive data has been completed.

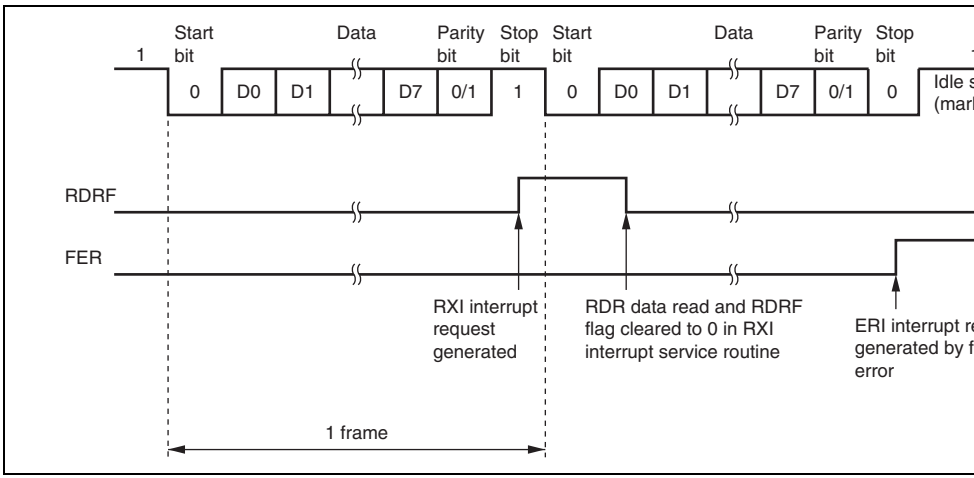
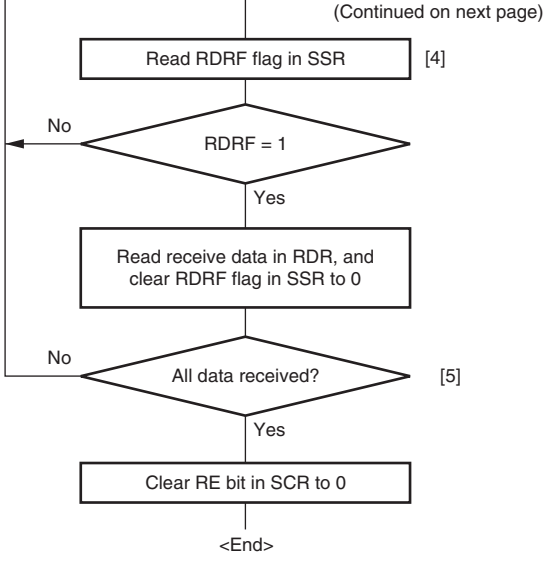


Figure 13.8 Example of SCI Operation in Reception (Example with 8-Bit Data, Parity, One Stop Bit)

0	0	1	0	Transferred to RDR	Framing error
0	0	0	1	Transferred to RDR	Parity error
1	1	1	0	Lost	Overrun error + frami
1	1	0	1	Lost	Overrun error + parity
0	0	1	1	Transferred to RDR	Framing error + parity
1	1	1	1	Lost	Overrun error + frami parity error

Note: * The RDRF flag retains the state it had before data reception.



value of the input port corresponding to the RxD pin.

[4] SCI status check and receive data read: Read SSR and check that RDRF = 1, then read the receive data in RDR and clear the RDRF flag to 0. Transition of the RDRF flag from 0 to 1 can also be identified by an RXI interrupt.

[5] Serial reception continuation procedure: To continue serial reception, before the stop bit for the current frame is received, read the RDRF flag, read RDR, and clear the RDRF flag to 0. The RDRF flag is cleared automatically when DTC* is activated by an RXI interrupt and the RDR value is read.

Note: * The case, where the RDRF flag is automatically cleared by DTC, occurs only when the DISEL bit in DTC with the transfer counter other than 0. Therefore, when the DISEL bit is 1, or both the DISEL bit and the transfer cou give the CPU an instruction to clear the

Figure 13.9 Sample Serial Reception Data Flowchart (1)

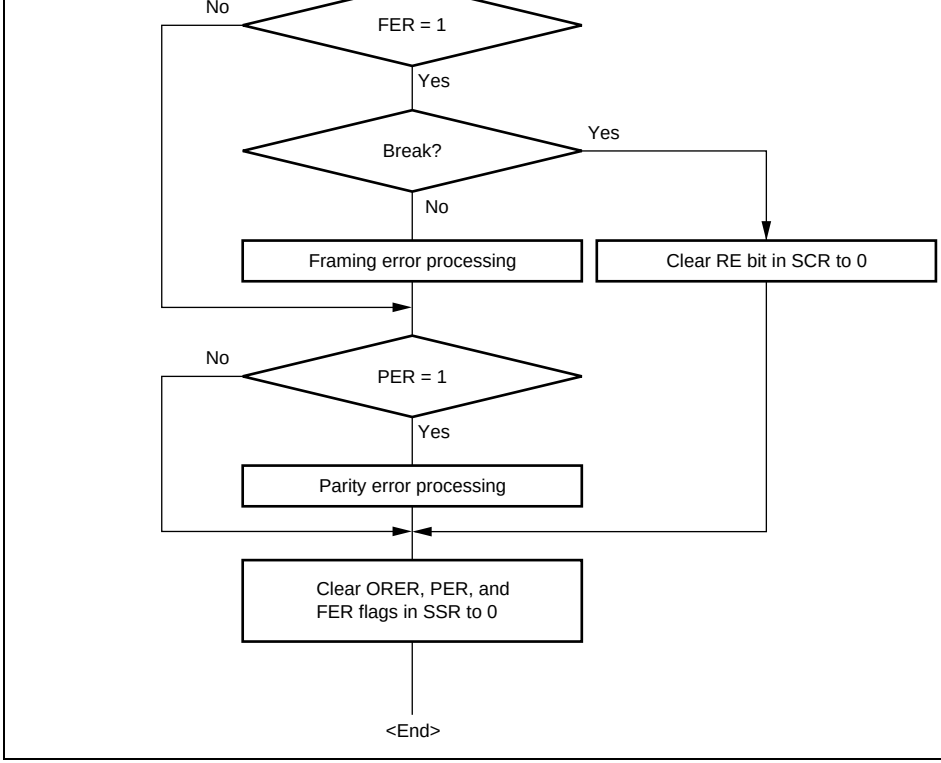


Figure 13.9 Sample Serial Reception Data Flowchart (2)

cycle is a data transmission cycle. Figure 15.16 shows an example of inter-processor
communication using the multiprocessor format. The transmitting station first sends the I
of the receiving station with which it wants to perform serial communication as data with
multiprocessor bit added. It then sends transmit data as data with a 0 multiprocessor bit a
When data with a 1 multiprocessor bit is received, the receiving station compares that dat
own ID. The station whose ID matches then receives the data sent next. Stations whose II
match continue to skip data until data with a 1 multiprocessor bit is again received.

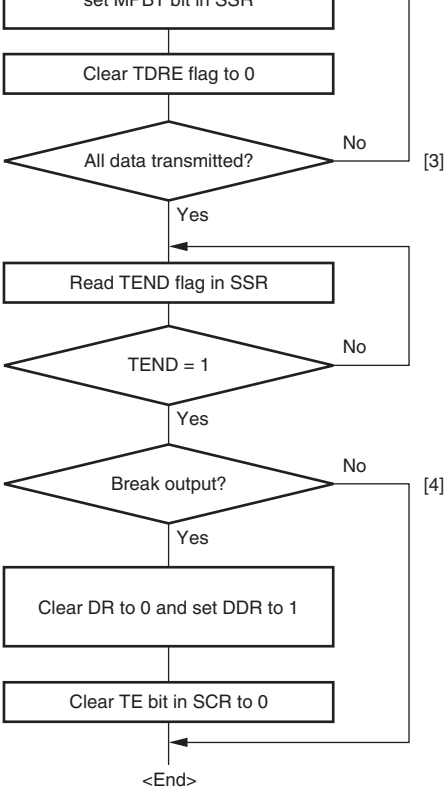
The SCI uses the MPIE bit in SCR to implement this function. When the MPIE bit is set
transfer of receive data from RSR to RDR, error flag detection, and setting the SSR status
RDRF, FER, and ORER to 1, are inhibited until data with a 1 multiprocessor bit is receiv
reception of a receive character with a 1 multiprocessor bit, the MPB bit in SSR is set to
MPIE bit is automatically cleared, thus normal reception is resumed. If the RIE bit in SCI
1 at this time, an RXI interrupt is generated.

When the multiprocessor format is selected, the parity bit setting is rendered invalid. All
settings are the same as those in normal asynchronous mode. The clock used for multipro
communication is the same as that in normal asynchronous mode.

specification | receiving station specified by ID

Legend:
MPB: Multiprocessor bit

Figure 13.10 Example of Communication Using Multiprocessor Format (Transmission of Data H'AA to Receiving Station A)



[3] Serial transmission continuation procedure:
To continue serial transmission, be sure to read 1 from the TDRE flag to confirm that writing is possible, then write data to TDR, and then clear the TDRE flag to 0. Checking and clearing of the TDRE flag is automatic when the DTC* is activated by a transmit data empty interrupt (TXI) request, and data is written to TDR.

[4] Break output at the end of serial transmission:
To output a break in serial transmission, set the port DR to 0, set DDR to 1, then clear the TE bit in SCR to 0.

Note: * The case, where the TDRE flag is automatically cleared by DTC, occurs only when the DISEL bit in DTC is with the transfer counter other than 0. Therefore, when the DISEL bit is 1, or both the DISEL bit and the transfer count give the CPU an instruction to clear the T

Figure 13.11 Sample Multiprocessor Serial Transmission Flowchart

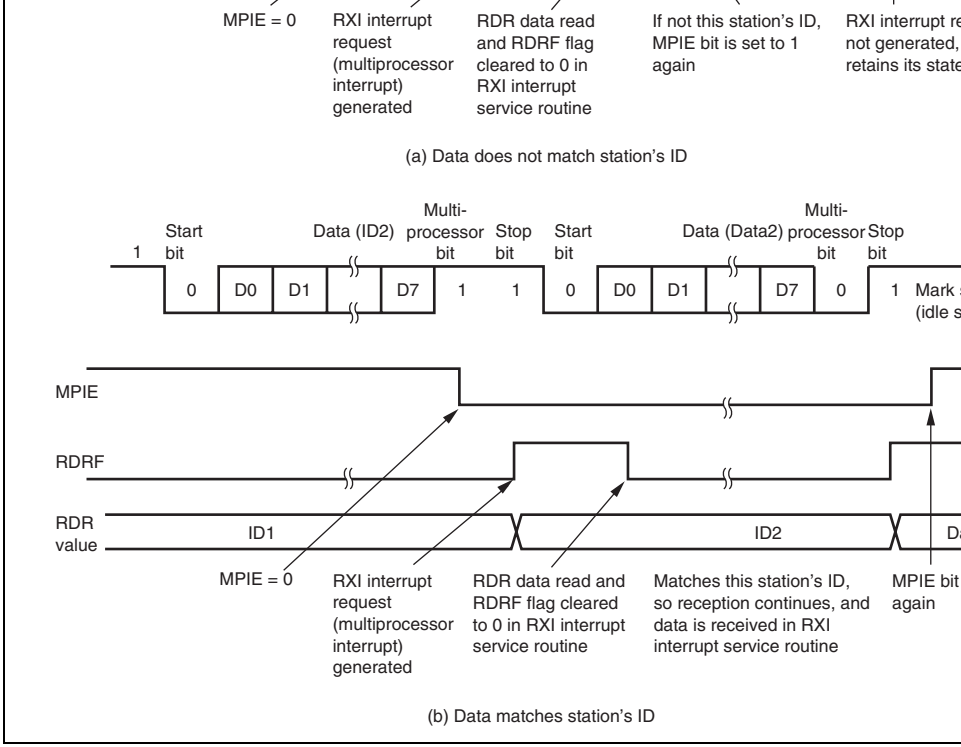
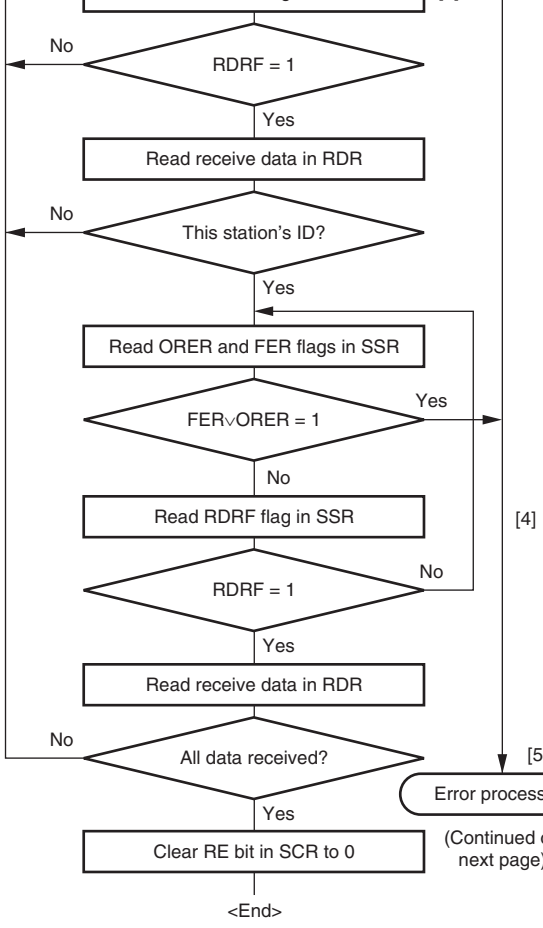


Figure 13.12 Example of SCI Operation in Reception (Example with 8-Bit Data, Multiprocessor Bit, One Stop Bit)



If the data is this station's ID, clear t
RDRF flag to 0.

[4] SCI status check and data reception
Read SSR and check that the RDRI
flag is set to 1, then read the data in
RDR.

[5] Receive error processing and break
detection:
If a receive error occurs, read the O
and FER flags in SSR to identify the
error. After performing the appropri
error processing, ensure that the OR
and FER flags are all cleared to 0.
Reception cannot be resumed if eith
of these flags is set to 1.
In the case of a framing error, a brea
can be detected by reading the RxD
value.

Figure 13.13 Sample Multiprocessor Serial Reception Flowchart (1)

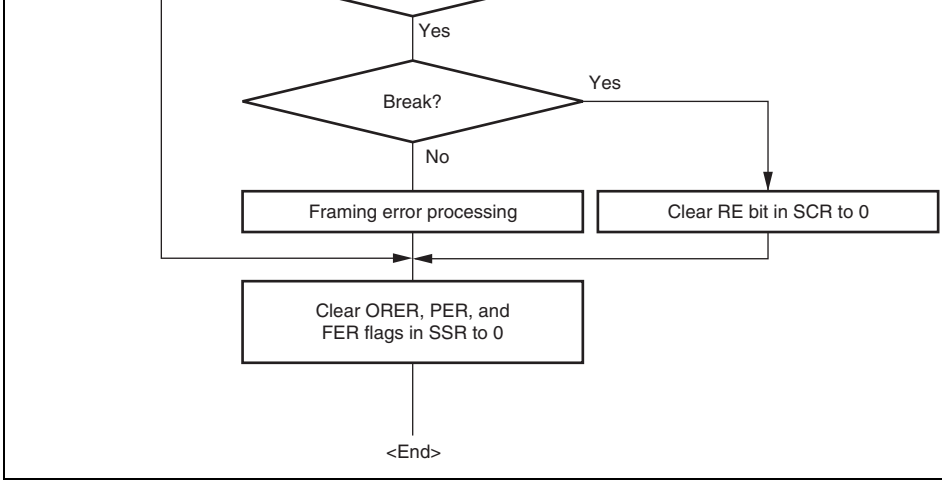


Figure 13.13 Sample Multiprocessor Serial Reception Flowchart (2)

structure, so data can be read or written during transmission or reception, enabling contin
transfer.

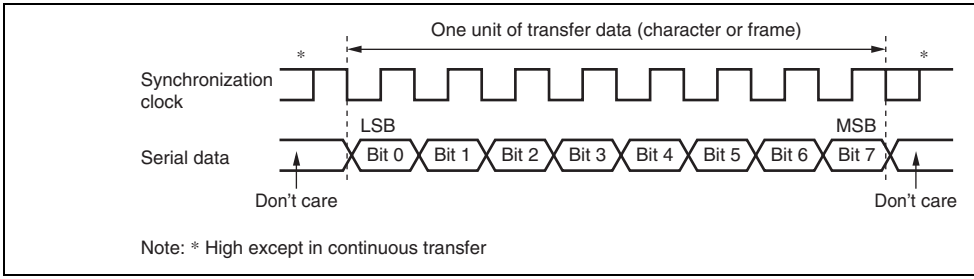
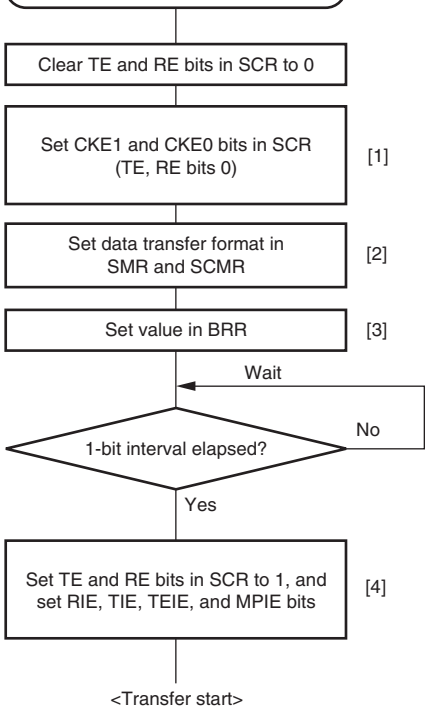


Figure 13.14 Data Format in Clocked Synchronous Communication (For LSB-

13.6.1 Clock

Either an internal clock generated by the on-chip baud rate generator or an external synchronization clock input at the SCK pin can be selected, according to the setting of CI CKE1 bits in SCR. When the SCI is operated on an internal clock, the serial clock is outp the SCK pin. Eight serial clock pulses are output in the transfer of one character, and whe transfer is performed the clock is fixed high.



to clear bits RIE, TIE, TEIE, and MPIE, TE and RE, to 0.

[2] Set the data transfer format in SMR and SCMR.

[3] Write a value corresponding to the bit rate to BRR. Not necessary if an external clock is used.

[4] Wait at least one bit interval, then set the TE bit or RE bit in SCR to 1. Also set the RIE, TIE TEIE, and MPIE bits.
Setting the TE and RE bits enables the TxD and RxD pins to be used.

Note: In simultaneous transmit and receive operations, the TE and RE bits should both be cleared to 0 or set to 1 simultaneously.

Figure 13.15 Sample SCI Initialization Flowchart

next transmit data to TDR before transmission of the current transmit data has been co

3. 8-bit data is sent from the TxD pin synchronized with the output clock when output cl mode has been specified, and synchronized with the input clock when use of an extern has been specified.
4. The SCI checks the TDRE flag at the timing for sending the MSB (bit 7).
5. If the TDRE flag is cleared to 0, data is transferred from TDR to TSR, and serial trans of the next frame is started.
6. If the TDRE flag is set to 1, the TEND flag in SSR is set to 1, and the TDRE flag mai output state of the last bit. If the TEIE bit in SCR is set to 1 at this time, a TEI interrup is generated. The SCK pin is fixed high.

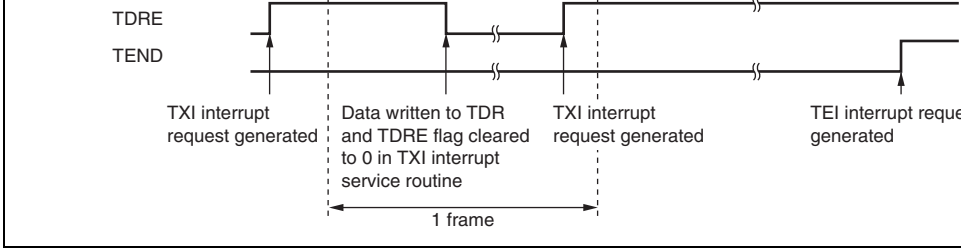
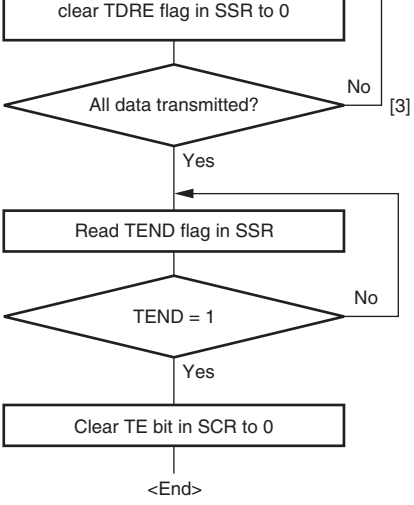


Figure 13.16 Sample SCI Transmission Operation in Clocked Synchronous M



write data to TDR, and then clear the TDRE flag to 0.
Checking and clearing of the TDRE flag is automatic when the DTC* is activated by a transmit data empty interrupt (TXI) request and data is written to TDR.

Note: * The case, where the TDRE flag is automatically cleared by DTC, occurs only when the DISEL bit in DTC is 0 with the transfer counter other than 0. Therefore, when the DISEL bit is 1, or both the DISEL bit and the transfer counter give the CPU an instruction to clear the TDI

Figure 13.17 Sample Serial Transmission Flowchart

RDRF flag remains to be set to 1.

3. If reception is completed successfully, the RDRF bit in SSR is set to 1, and receive d transferred to RDR. If the RIE bit in SCR is set to 1 at this time, an RXI interrupt req generated. Continuous reception is possible because the RXI interrupt routine reads t data transferred to RDR before reception of the next receive data has finished.

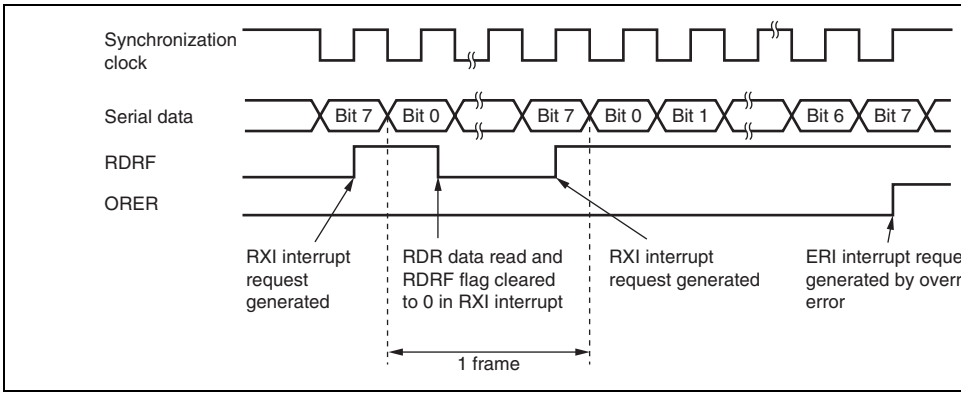
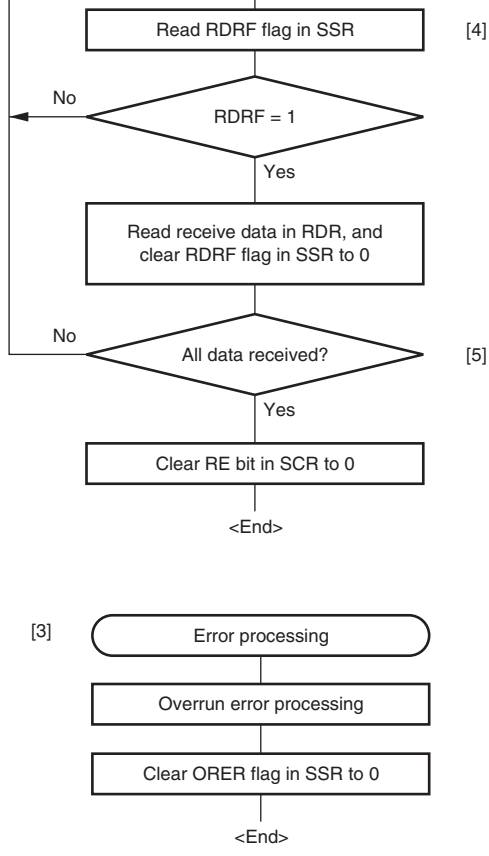


Figure 13.18 Example of SCI Operation in Reception

Reception cannot be resumed while a receive error flag is set to 1. Accordingly, clear th FER, PER, and RDRF bits to 0 before resuming reception. Figure 13.19 shows a sample chart for serial data reception.

An overrun error occurs or synchronous clocks are output until the RE bit is cleared to 0 internal clock is selected and only receive operation is possible. When a transmission an reception will be carried out in a unit of one frame, be sure to carry out a dummy transm with only one frame by the simultaneous transmit and receive operations at the same tim



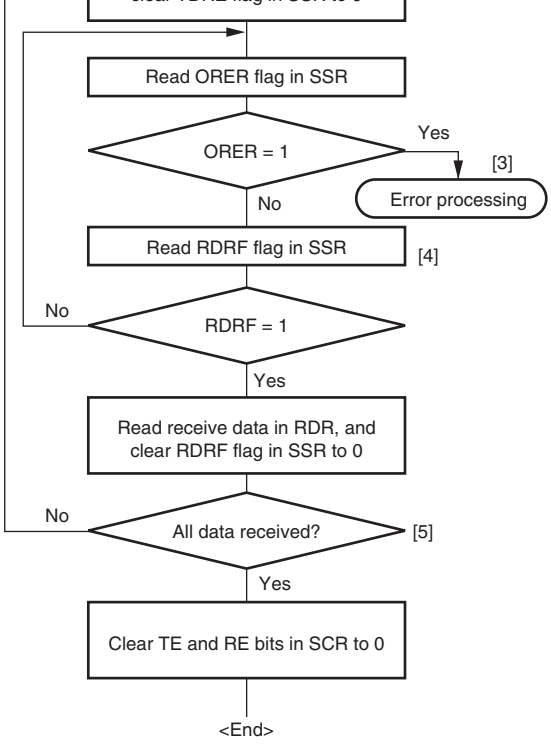
[4] Receive error processing and break detection:
If a receive error occurs, the RDRF flag is set to 1, then read the receive data in RDR and clear the RDRF flag to 0.
Transition of the RDRF flag from 0 to 1 can also be identified by an RXI interrupt.

[5] Serial reception continuation procedure:
To continue serial reception, before the final bit of the current frame is received, reading the RDRF flag, reading RDR, and clearing the RDRF flag to 0 should be finished. The RDRF flag is cleared automatically when the DTC* is activated by a receive data full interrupt (RXI) request and the RDR value is read.

Note: * The case, where the RDRF flag check clearing are automatically executed by occurs only when the DISEL bit in DTC with the transfer counter other than 0. Therefore, when the DISEL bit is 1, or both the DISEL bit and the transfer cou give the CPU an instruction to clear the

Figure 13.19 Sample Serial Reception Flowchart

and PER) are cleared to 0, simultaneously set TE and RE to 1 with a single instruction.



ORER flag in SSR, and after performing the appropriate error processing, clear the ORER flag to Transmission/reception cannot be resumed if the ORER flag is set to

[4] SCI status check and receive data read:
Read SSR and check that the RDR flag is set to 1, then read the receiv data in RDR and clear the RDRF fl to 0. Transition of the RDRF flag f 0 to 1 can also be identified by an interrupt.

[5] Serial transmission/reception continuation procedure:
To continue serial transmission/ reception, before the final bit of the current frame is received, finish reading the RDRF flag, reading RD and clearing the RDRF flag to 0. Also, before the final bit of the curr frame is transmitted, read 1 from th TDRE flag to confirm that writing is possible. Then write data to TDR a clear the TDRE flag to 0.
Checking and clearing of the TDRE flag is automatic when the DTC* is activated by a transmit data empty interrupt (TXI) request and data is written to TDR. Also, the RDRF fla is cleared automatically when the DTC* is activated by a receive data full interrupt (RXI) request and the RDR value is read.

Notes: When switching from transmit or receive operation to simultaneous transmit and receive operations, first clear the TE bit and RE bit to 0, then set both these bits to 1 by one instruction simultaneously.

* The case, where the TDRE flag or RDRF flag is automatically cleared by DTC, occurs only when the DISEL bit in the corresponding DTC is 0 with the transfer counter other than 0. Therefore, when the DISEL bit in the corresponding DTC is 1, or both the DISEL bit and the transfer co give the CPU an instruction to clear the corresponding flags.

Figure 13.20 Sample Flowchart of Simultaneous Serial Transmit and Receive Op

pin and RxD pin should be connected to the LSI pin. The data transmission line should b
up to the power supply (other than channel 2: P2Vcc, channel 2: P1Vcc) with a resistor.
card is not connected, and the TE and RE bits are both set to 1, closed transmission/rece
possible, enabling self-diagnosis to be carried out. When the clock generated on the SCI
supplied to an IC card, the SCK pin output is input to the CLK pin of the IC card. When
internal clock is used in an IC card, this connection is not necessary. This LSI port outpu
as the reset signal. Adding to these connections, connection of pins with power supply a
is necessary.

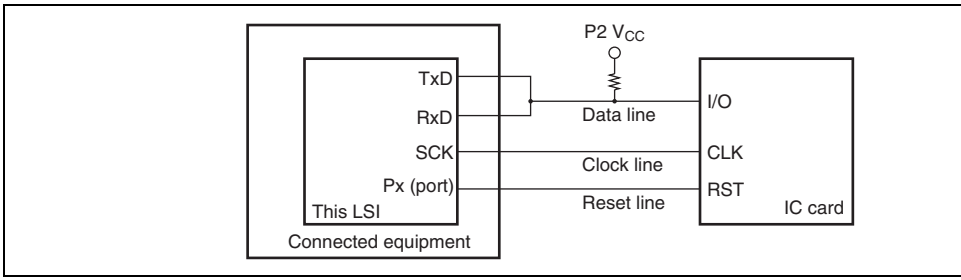


**Figure 13.21 (1) Schematic Diagram of Smart Card Interface Pin Connecti
(Channels 0, 1, 3, and 4)**

13.7.2 Data Format (Except for Block Transfer Mode)

Figure 13.22 shows the transfer data format in Smart Card interface mode.

- One frame consists of 8-bit data plus a parity bit in asynchronous mode.
- In transmission, a guard time of at least 2 etu (Elementary time unit: the time for trans one bit) is left between the end of the parity bit and the start of the next frame.
- If a parity error is detected during reception, a low error signal level is output for one period, 10.5 etu after the start bit.
- If an error signal is sampled during transmission, the same data is retransmitted autom after a delay of 2 etu or longer.

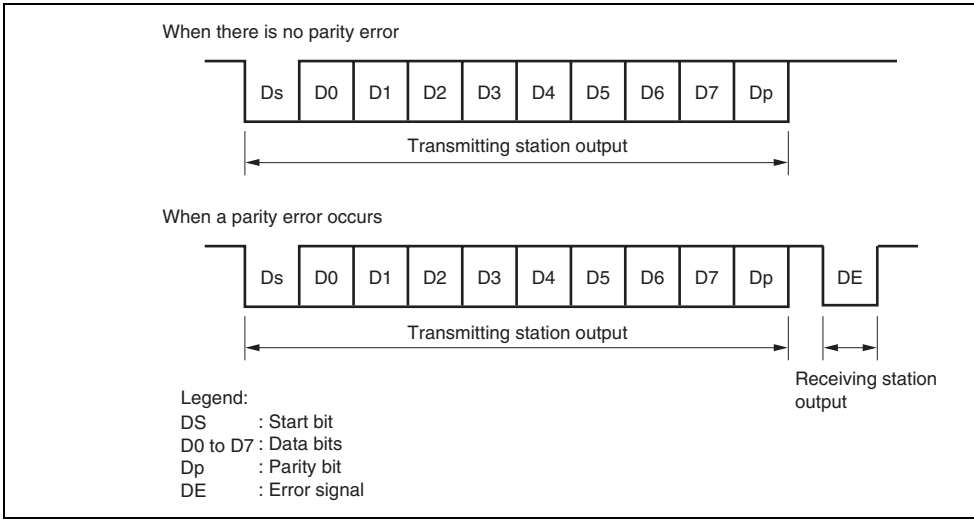


Figure 13.22 Normal Smart Card Interface Data Format

The start character data above is H'3F. For the direct convention type, clear the SDIR and bits in SCMR to 0. According to Smart Card regulations, clear the O/$\overline{E}$ bit in SMR to 0 t even parity mode.

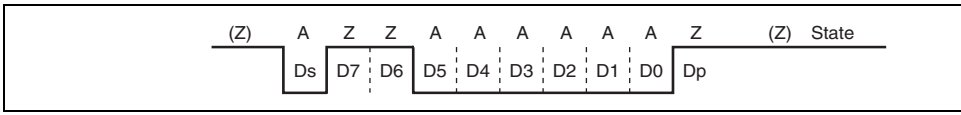


Figure 13.24 Inverse Convention (SDIR = SINV = O/$\overline{E}$ = 1)

With the inverse convention type, the logic 1 level corresponds to state A and the logic 0 state Z, and transfer is performed in MSB-first order. The start character data for the abo H'3F. For the inverse convention type, set the SDIR and SINV bits in SCMR to 1. Acco Smart Card regulations, even parity mode is the logic 0 level of the parity bit, and corres state Z. In this LSI, the SINV bit inverts only data bits D0 to D7. Therefore, set the O/$\overline{E}$ SMR to 1 to invert the parity bit for both transmission and reception.

13.7.3 Block Transfer Mode

Operation in block transfer mode is the same as that in the normal Smart Card interface except for the following points.

- In reception, though the parity check is performed, no error signal is output even if a detected. However, the PER bit in SSR is set to 1 and must be cleared before receivi parity bit of the next frame.
- In transmission, a guard time of at least 1 etu is left between the end of the parity bit start of the next frame.
- In transmission, because retransmission is not performed, the TEND flag is set to 1, after transmission start.

falling edge of the start bit using the basic clock, and performs internal synchronization. R
in figure 13.25, by sampling receive data at the rising-edge of the 16th, 32nd, 186th, or 12
pulse of the basic clock, data can be latched at the middle of the bit. The reception margin
by the following formula.

$$M = \left| (0.5 - \frac{1}{2N}) - (L - 0.5) F - \frac{|D - 0.5|}{N} (1 + F) \right| \times 100\%$$

Where M: Reception margin (%)
N: Ratio of bit rate to clock (N = 32, 64, 372, and 256)
D: Clock duty (D = 0 to 1.0)
L: Frame length (L = 10)
F: Absolute value of clock frequency deviation

Assuming values of F = 0, D = 0.5 and N = 372 in the above formula, the reception marg
formula is as follows.

$$M = (0.5 - 1/2 \times 372) \times 100\%$$
$$= 49.866\%$$

**Figure 13.25 Receive Data Sampling Timing in Smart Card Interface Mo
(Using Clock of 372 Times the Transfer Rate)**

13.7.5 Initialization

Before transmitting and receiving data, initialize the SCI as described below. Initializati
necessary when switching from transmit mode to receive mode, or vice versa.

1. Clear the TE and RE bits in SCR to 0.
2. Clear the error flags ERS, PER, and ORER in SSR to 0.
3. Set the GM, BLK, O/$\overline{E}$, BCP0, BCP1, CKS0, and CKS1 bits in SMR. Set the PE bit
4. Set the SMIF, SDIR, and SINV bits in SCMR.
   When the SMIF bit is set to 1, the TxD and RxD pins are both switched from ports t
   and are placed in the high-impedance state.
5. Set the value corresponding to the bit rate in BRR.
6. Set the CKE0 and CKE1 bits in SCR. Clear the TIE, RIE, TE, RE, MPIE, and TEIE
   If the CKE0 bit is set to 1, the clock is output from the SCK pin.
7. Wait at least one bit interval, then set the TIE, RIE, TE, and RE bits in SCR. Do not
   bit and RE bit at the same time, except for self-diagnosis.

To switch from receive mode to transmit mode, after checking that the SCI has finished
initialize the SCI, and set RE to 0 and TE to 1. Whether SCI has finished reception or n
checked with the RDRF, PER, or ORER flags. To switch from transmit mode to receive
after checking that the SCI has finished transmission, initialize the SCI, and set TE to 0
1. Whether SCI has finished transmission or not can be checked with the TEND flag.

2. The TEND bit in SSR is not set for a frame in which an error signal indicating an abn is received. Data is retransferred from TDR to TSR, and retransmitted automatically.
3. If an error signal is not sent back from the receiving end, the ERS bit in SSR is not se Transmission of one frame, including a retransfer, is judged to have been completed, TEND bit in SSR is set to 1. If the TIE bit in SCR is enabled at this time, a TXI interr request is generated. Writing transmit data to TDR transfers the next transmit data.

Figure 13.28 shows a flowchart for transmission. A sequence of transmit operations can b performed automatically by specifying the DTC to be activated with a TXI interrupt sour transmit operation, the TDRE flag is set to 1 at the same time as the TEND flag in SSR is a TXI interrupt will be generated if the TIE bit in SCR has been set to 1. If the TXI reque designated beforehand as a DTC activation source, the DTC will be activated by the TXI and transfer of the transmit data will be carried out. At this moment, when the DISEL bit is 0 and the transfer counter is other than 0, the TDRE and TEND flags are automatically to 0 when data is transferred by the DTC. When the DISEL bit in the corresponding DTC both the DISEL bit and the transfer counter are 0, flags are not cleared although the trans written to TDR by DTC. Consequently, give the CPU an instruction of flag clear processi addition, in the event of an error, the SCI retransmits the same data automatically. During period, the TEND flag remains cleared to 0 and the DTC is not activated. Therefore, the S DTC will automatically transmit the specified number of bytes in the event of an error, in retransmission. However, the ERS flag is not cleared automatically when an error occurs, the RIE bit should be set to 1 beforehand so that an ERI request will be generated in the e an error, and the ERS flag will be cleared.

When performing transfer using the DTC, it is essential to set and enable the DTC before out SCI setting. For details on the DTC setting procedures, see section 8, Data Transfer C (DTC).

The timing for setting the TEND flag depends on the value of the GM bit in SMR. The T
set timing is shown in figure 13.27.

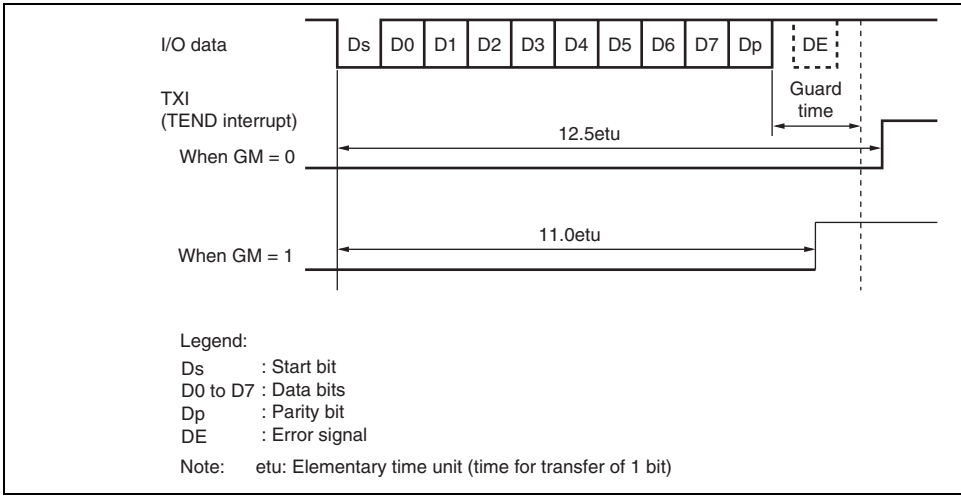


Figure 13.27 TEND Flag Generation Timing in Transmission Operation

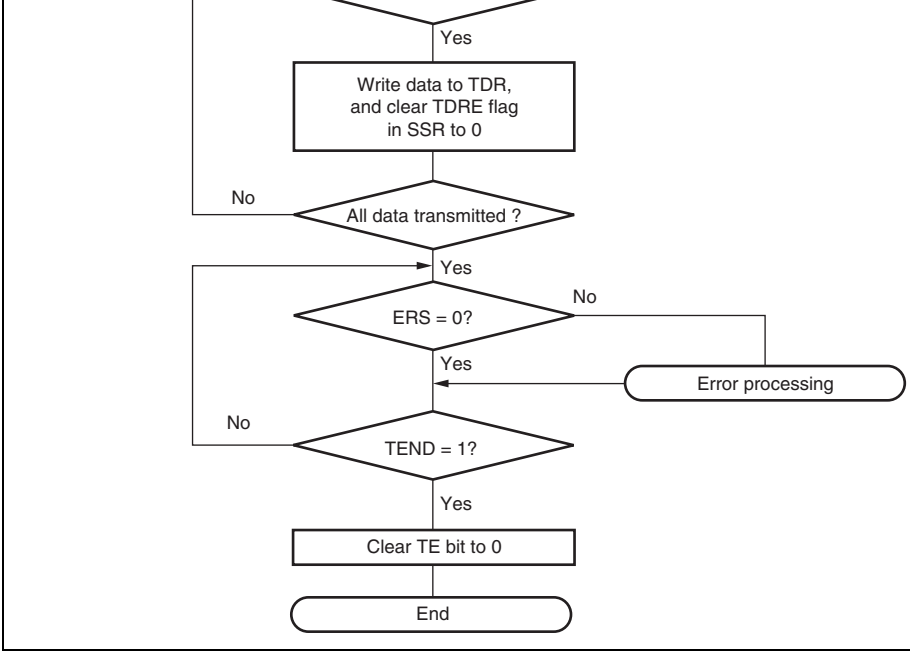


Figure 13.28 Example of Transmission Processing Flow

5. If no error is found when the received parity bit is checked, the PER bit in SSR is no the receive operation is judged to have been completed normally, and the RDRF flag automatically set to 1. If the RIE bit in SCR is enabled at this time, an RXI interrupt generated.

Figure 13.30 shows a flowchart for reception. A sequence of receive operations can be p automatically by specifying the DTC to be activated with an RXI interrupt source. In a r operation, an RXI interrupt request is generated when the RDRF flag in SSR is set to 1. request is designated beforehand as a DTC activation source, the DTC will be activated RXI request, and the receive data will be transferred. The RDRF flag is cleared to 0 auto when the DISEL bit in DTC is 0 and the transfer counter is other than 0. When the DISE DTC is 1, or both the DISEL bit and the transfer counter are 0, flags are not cleared alth receive data is transferred by DTC. Consequently, give the CPU an instruction of flag cl processing. If an error occurs in receive mode and the ORER or PER flag is set to 1, a tr error interrupt (ERI) request will be generated. Hence, so the error flag must be cleared t event of an error, the DTC is not activated and receive data is skipped. Therefore, receiv transferred for only the specified number of bytes in the event of an error. Even when a error occurs in receive mode and the PER flag is set to 1, the data that has been received transferred to RDR and can be read from there.

Note: For details on receive operations in block transfer mode, see section 13.4, Opera Asynchronous Mode.

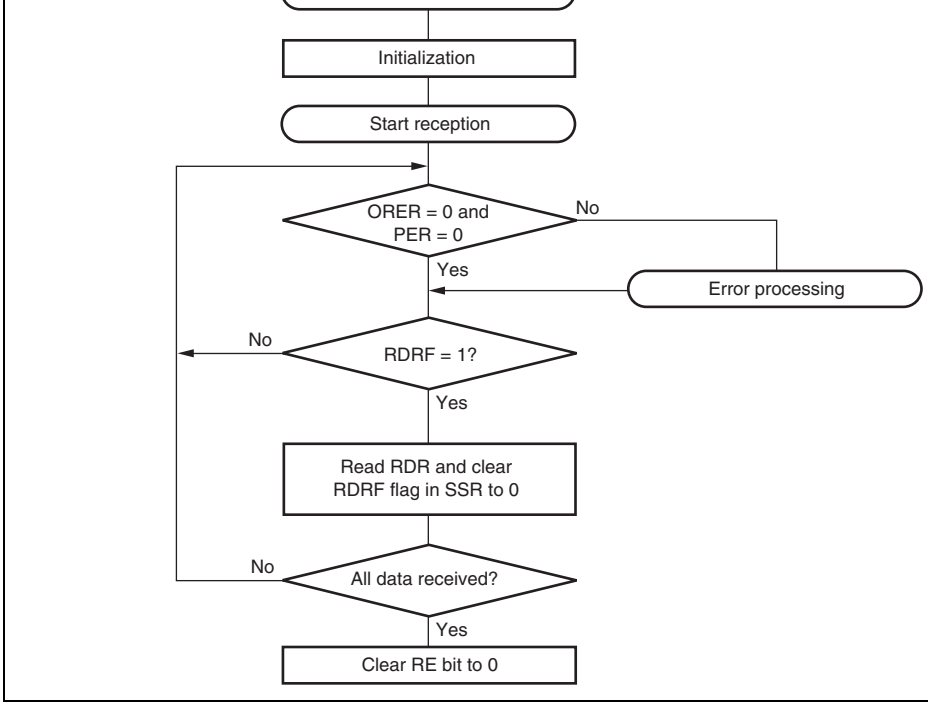


Figure 13.30 Example of Reception Processing Flow

13.7.8 Clock Output Control

When the GM bit in SMR is set to 1, the clock output level can be fixed with bits CKE0 a
CKE1 in SCR. At this time, the minimum clock pulse width can be made the specified wi
Figure 13.31 shows the timing for fixing the clock output level. In this example, GM is se
CKE1 is cleared to 0, and the CKE0 bit is controlled.

When turning on the power or switching between Smart Card interface mode and softw
standby mode, the following procedures should be followed in order to maintain the clo

Powering On: To secure clock duty from power-on, the following switching procedure
followed.

1. The initial state is port input and high impedance. Use a pull-up resistor or pull-d
   resistor to fix the potential.
2. Fix the SCK pin to the specified output level with the CKE1 bit in SCR.
3. Set SMR and SCMR, and switch to smart card mode operation.
4. Set the CKE0 bit in SCR to 1 to start clock output.

When Changing from Smart Card Interface Mode to Software Standby Mode:

1. Set the data register (DR) and data direction register (DDR) corresponding to the
   to the value for the fixed output state in software standby mode.
2. Write 0 to the TE bit and RE bit in the serial control register (SCR) to halt transm
   operation. At the same time, set the CKE1 bit to the value for the fixed output sta
   software standby mode.
3. Write 0 to the CKE0 bit in SCR to halt the clock.
4. Wait for one serial clock period.
   During this interval, clock output is fixed at the specified level, with the duty pre
5. Make the transition to the software standby state.

When Returning to Smart Card Interface Mode from Software Standby Mode:

1. Exit the software standby state.
2. Write 1 to the CKE0 bit in SCR and output the clock. Signal generation is started
   normal duty.

13.8.1 Interrupts in Normal Serial Communication Interface Mode

Table 13.12 shows the interrupt sources in normal serial communication interface mode. different interrupt vector is assigned to each interrupt source, and individual interrupt sou be enabled or disabled using the enable bits in SCR.

When the TDRE flag in SSR is set to 1, a TXI interrupt request is generated. When the T in SSR is set to 1, a TEI interrupt request is generated. A TXI interrupt can activate the D perform data transfer. The TDRE flag is cleared to 0 automatically when data is transferr DTC*.

When the RDRF flag in SSR is set to 1, an RXI interrupt request is generated. When the PER, or FER flag in SSR is set to 1, an ERI interrupt request is generated. An RXI interru request can activate the DTC to transfer data. The RDRF flag is cleared to 0 automaticall data is transferred by the DTC*.

A TEI interrupt is requested when the TEND flag is set to 1 and the TEIE bit is set to 1. I interrupt and a TXI interrupt are requested simultaneously, the TXI interrupt has priority acceptance. However, if the TDRE and TEND flags are cleared simultaneously by the TX interrupt routine, the SCI cannot branch to the TEI interrupt routine later.

Note: * Flags are cleared only when the DISEL bit in DTC is 0 with the transfer coute than 0.

	TEI1	Transmission End	TEND	Not possible	
2	ERI2	Receive Error	ORER, FER, PER	Not possible	
	RXI2	Receive Data Full	RDRF	Possible	
	TXI2	Transmit Data Empty	TDRE	Possible	
	TEI2	Transmission End	TEND	Not possible	
3	ERI3	Receive Error	ORER, FER, PER	Not possible	
	RXI3	Receive Data Full	RDRF	Possible	
	TXI3	Transmit Data Empty	TDRE	Possible	
	TEI3	Transmission End	TEND	Not possible	
4	ERI4	Receive Error	ORER, FER, PER	Not possible	
	RXI4	Receive Data Full	RDRF	Possible	
	TXI4	Transmit Data Empty	TDRE	Possible	
	TEI4	Transmission End	TEND	Not possible	L

Note: * Indicates the initial state immediately after a reset. Priorities in channels can b changed by the interrupt controller.

Channel	Name	Interrupt Source	Interrupt Flag		Pr
0	ERI0	Receive Error, detection	ORER, PER, ERS	Not possible	Hi
	RXI0	Receive Data Full	RDRF	Possible	
	TXI0	Transmit Data Empty	TEND	Possible	
1	ERI1	Receive Error, detection	ORER, PER, ERS	Not possible	
	RXI1	Receive Data Full	RDRF	Possible	
	TXI1	Transmit Data Empty	TEND	Possible	
2	ERI2	Receive Error, detection	ORER, PER, ERS	Not possible	
	RXI2	Receive Data Full	RDRF	Possible	
	TXI2	Transmit Data Empty	TEND	Possible	
3	ERI3	Receive Error, detection	ORER, PER, ERS	Not possible	
	RXI3	Receive Data Full	RDRF	Possible	
	TXI3	Transmit Data Empty	TEND	Possible	
4	ERI4	Receive Error, detection	ORER, PER, ERS	Not possible	
	RXI4	Receive Data Full	RDRF	Possible	
	TXI4	Transmit Data Empty	TEND	Possible	Lo

Note: * Indicates the initial state immediately after a reset. Priorities in channels can be changed by the interrupt controller.

When framing error (FER) detection is performed, a break can be detected by reading th value directly. In a break, the input from the RxD pin becomes all 0s, setting the FER fla possibly the PER flag. Note that as the SCI continues the receive operation after receivi even if the FER flag is cleared to 0, it will be set to 1 again.

13.9.3 Mark State and Break Detection (Asynchronous Mode Only)

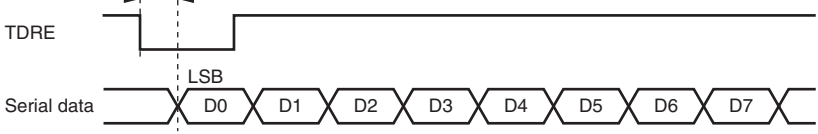
When TE is 0, the TxD pin is used as an I/O port whose direction (input or output) and l determined by DDR. This can be used to set the TxD pin to mark state (high level) or se during serial data transmission. To maintain the communication line at mark state until T 1, set both DDR and DR to 1. As TE is cleared to 0 at this point, the TxD pin becomes a and 1 is output from the TxD pin. To send a break during serial transmission, first set PI and DR to 0, and then clear TE to 0. When TE is cleared to 0, the transmitter is initialize regardless of the current transmission state, the TxD pin becomes an I/O port, and 0 is o the TxD pin.

13.9.4 Receive Error Flags and Transmit Operations (Clocked Synchronous Mo

Transmission cannot be started when a receive error flag (ORER, PER, or FER) is set to the TDRE flag is cleared to 0. Be sure to clear the receive error flags to 0 before starting transmission. Note also that receive error flags cannot be cleared to 0 even if the RE bit to 0.

13.9.5 Restrictions on Use of DTC

- When an external clock source is used as the serial clock, the transmit clock should r input until at least 5 ϕ clock cycles after TDR is updated by the DTC. Incorrect opera occur if the transmit clock is input within 4 ϕ clocks after TDR is updated. (Figure 1



Note: When operating on an external clock, set t >4 clocks.

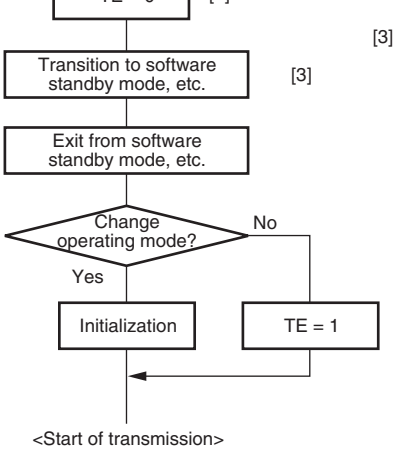
Figure 13.33 Example of Clocked Synchronous Transmission by DTC

13.9.6 Operation in Case of Mode Transition

- Transmission

  Operation should be stopped (by clearing TE, TIE, and TEIE bits to 0) before making module stop mode, software standby mode, or watch mode transition. TSR, TDR, and reset. The output pin states in module stop mode, software standby mode, or watch m depend on the port settings, and become high-level output after the relevant mode is c a transition is made during transmission, the data being transmitted will be undefined. transmitting without changing the transmit mode after the relevant mode is cleared, transmission can be started by setting TE to 1 again, and performing the following seq SSR read -> TDR write -> TDRE clearance. To transmit with a different transmit mo clearing the relevant mode, the procedure must be started again from initialization. Fi 13.34 shows a sample flowchart for mode transition during transmission. Port pin stat shown in figures 13.35 and 13.36.

  Operation should also be stopped (by clearing TE, TIE, and TEIE bits to 0) before ma transition from transmission by DTC transfer to module stop mode, software standby watch mode. To perform transmission with the DTC after the relevant mode is cleare the TE and TIE bits to 1 will set the TXI flag and start DTC transmission.



[3] Includes module stop mode and watch mode.

Figure 13.34 Sample Flowchart for Mode Transition during Transmissio

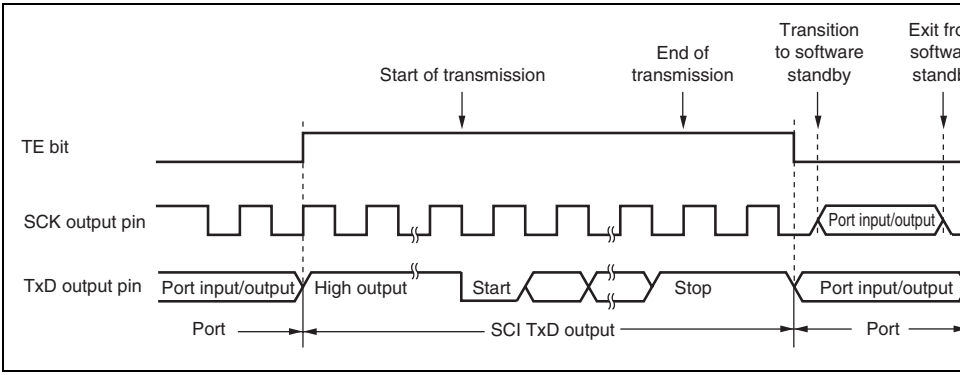


Figure 13.35 Asynchronous Transmission Using Internal Clock

Note: * Initialized by software standby.

Figure 13.36 Clocked Synchronous Transmission Using Internal Clock

- Reception

  Receive operation should be stopped (by clearing RE to 0) before making a module st
  software standby mode, or watch mode transition. RSR, RDR, and SSR are reset. If a
  transition is made without stopping operation, the data being received will be invalid.

  To continue receiving without changing the reception mode after the relevant mode is
  set RE to 1 before starting reception. To receive with a different receive mode, the pro
  must be started again from initialization.

  Figure 13.37 shows a sample flowchart for mode transition during reception.

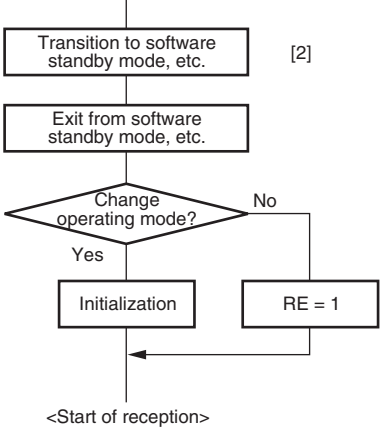


Figure 13.37 Sample Flowchart for Mode Transition during Reception

13.9.7 Notes when Switching from SCK Pin to Port Pin

- Problem in Operation: When DDR and DR are set to 1, SCI clock output is used in c synchronous mode, and the SCK pin is changed to the port pin while transmission is port output is enabled after low-level output occurs for one half-cycle.

  When switching the SCK pin to the port pin by making the following settings while DR = 1, C/$\overline{A}$ = 1, CKE1 = 0, CKE0 = 0, and TE = 1, low-level output occurs for one cycle.

1. End of serial data transmission
2. TE bit = 0
3. C/$\overline{A}$ bit = 0 ... switchover to port output
4. Occurrence of low-level output (see figure 13.38)

CKE0

Figure 13.38 Operation when Switching from SCK Pin to Port Pin

- Usage Note: To prevent low-level output occurred when switching the SCK pin to po
  follow the procedure described below.

  As this sample procedure temporarily places the SCK pin in the input state, the SCK/
  should be pulled up beforehand with an external circuit.

  With DDR = 1, DR = 1, C/$\overline{A}$ = 1, CKE1 = 0, CKE0 = 0, and TE = 1, make the follow
  settings in the order shown.

1. End of serial data transmission
2. TE bit = 0
3. CKE1 bit = 1
4. C/$\overline{A}$ bit = 0 ... switchover to port output
5. CKE1 bit = 0

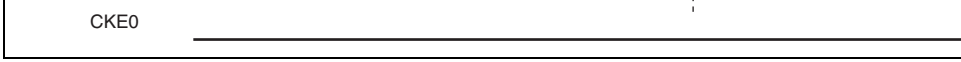


Figure 13.39 Operation when Switching from SCK Pin to Port Pin (Example of Preventing Low-Level Output)

Figure 14.2 shows an example of I/O pin connections to external circuits.

14.1 Features

- Selection of I^2C format or clocked synchronous serial format
- Continuous transmission/reception

  Since the shift register, transmit data register, and receive data register are independe each other, the continuous transmission/reception can be performed.
- Module stop mode can be set.

I^2C bus format

- Start and stop conditions generated automatically in master mode
- Selection of acknowledge output levels when receiving
- Automatic loading of acknowledge bit when transmitting
- Bit synchronization/wait function

  In master mode, the state of SCL is monitored per bit, and the timing is synchronized automatically.

  If transmission/reception is not yet possible, set the SCL to low until preparations ar completed.
- Six interrupt sources

  Transmit data empty (including slave-address match), transmit end, receive data full slave-address match), arbitration lost, NACK detection, and stop condition detection

voltages in excess of the power supply (P2Vcc) voltage for this LSI. The maximum volta
not exceed 0.3 V + this LSI's power supply voltage (P2Vcc). Since the I/O pins for chann
driven only by NMOS transistors, so in terms of appearance they carry out the same oper
an NMOS open drain. However, the voltage which can be applied to the I/O pins depends
voltage of the power supply (P2Vcc) of this LSI.

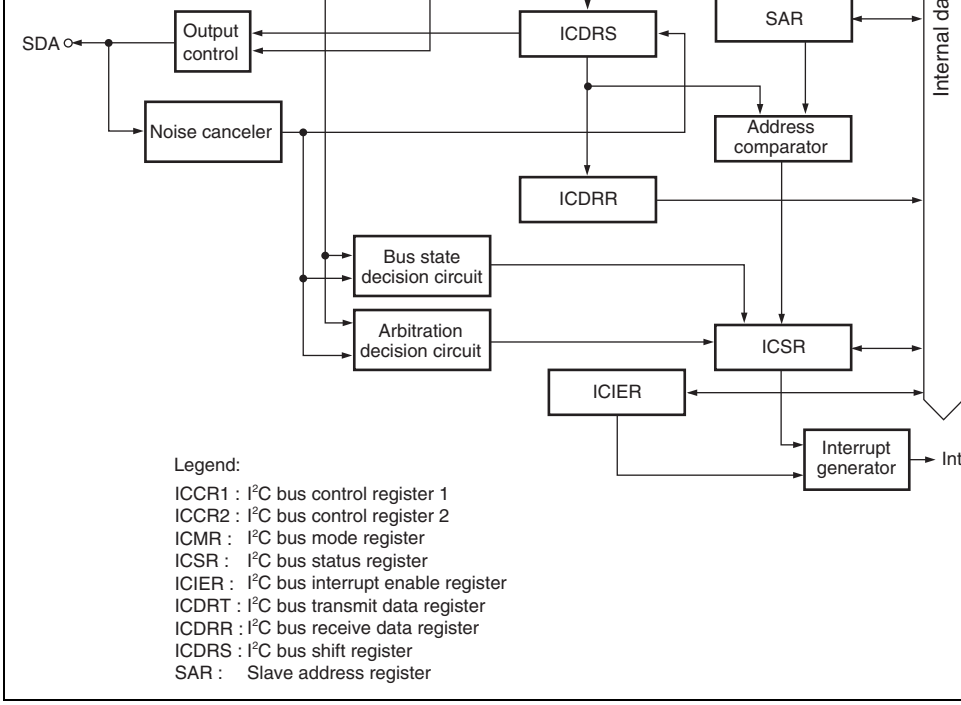


Figure 14.1 Block Diagram of I²C Bus Interface 2

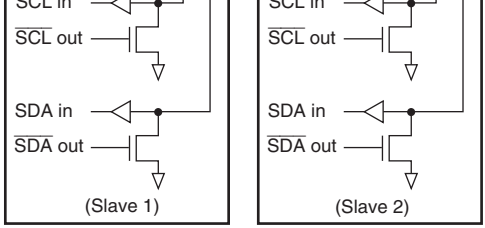


Figure 14.2 External Circuit Connections of I/O Pins

14.2 Input/Output Pins

Table 14.1 summarizes the input/output pins used by the I²C bus interface 2.

Table 14.1 I²C Bus Interface Pins

Channel	Abbreviation	I/O	Description
0	SCL0	I/O	Serial clock input/output for channel 0
	SDA0	I/O	Serial data input/output for channel 0
1	SCL1	I/O	Serial clock input/output for channel 1
	SDA1	I/O	Serial data input/output for channel 1

Note: * In the text, the channel subscript is omitted, and only SCL and SDA are used.

- I²C bus status register_0 (ICSR_0)
- I²C bus slave address register_0 (SAR_0)
- I²C bus transmit data register_0 (ICDRT_0)
- I²C bus receive data register_0 (ICDRR_0)
- I²C bus shift register_0 (ICDRS_0)

Channel 1

- I²C bus control register 1_1 (ICCR1_1)
- I²C bus control register 2_1 (ICCR2_1)
- I²C bus mode register_1 (ICMR_1)
- I²C bus interrupt enable register_1 (ICIER_1)
- I²C bus status register_1 (ICSR_1)
- I²C bus slave address register_1 (SAR_1)
- I²C bus transmit data register_1 (ICDRT_1)
- I²C bus receive data register_1 (ICDRR_1)
- I²C bus shift register_1 (ICDRS_1)

14.3.1 I²C Bus Control Register 1 (ICCR1)

ICCR1 enables or disables the I²C bus interface 2, controls transmission or reception, an
master or slave mode, transmission or reception, and transfer clock frequency in master

				0: Enables next reception 1: Disables next reception
5	MST	0	R/W	Master/Slave Select
4	TRS	0	R/W	Transmit/Receive Select In master mode with the I^2C bus format, when arbi lost, MST and TRS are both reset by hardware, ca transition to slave receive mode. Modification of th bit should be made between transfer frames. After data receive has been started in slave receiv when the first seven bits of the receive data agree slave address that is set to SAR and the eighth bit TRS is automatically set to 1. If an overrun error o master mode with the clock synchronous serial for MST is cleared to 0 and slave receive mode is ent Operating modes are described below according t and TRS combination. When clocked synchronous format is selected and MST is 1, clock is output. 00: Slave receive mode 01: Slave transmit mode 10: Master receive mode 11: Master transmit mode
3	CKS3	0	R/W	Transfer Clock Select 3 to 0
2	CKS2	0	R/W	Set these bits according to the necessary transfer
1	CKS1	0	R/W	master mode (see table 14.2). During slave mode,
0	CKS0	0	R/W	bits are specified to ensure enough time for data s transmit mode. When CKS3 is 0, the time is 10 tcy CKS3 is 1, 20 tcyc.

		1	0	φ/112	71.4 kHz	89.3 kHz	143 kHz	179 kHz	2
			1	φ/128	62.5 kHz	78.1 kHz	125 kHz	156 kHz	
1	0	0	0	φ/56	143 kHz	179 kHz	286 kHz	357 kHz	4
			1	φ/80	100 kHz	125 kHz	200 kHz	250 kHz	3
		1	0	φ/96	83.3 kHz	104 kHz	167 kHz	208 kHz	2
			1	φ/128	62.5 kHz	78.1 kHz	125 kHz	156 kHz	
	1	0	0	φ/336	23.8 kHz	29.8 kHz	47.6 kHz	59.5 kHz	1
			1	φ/200	40.0 kHz	50.0 kHz	80.0 kHz	100 kHz	
		1	0	φ/224	35.7 kHz	44.6 kHz	71.4 kHz	89.3 kHz	1
			1	φ/256	31.3 kHz	39.1 kHz	62.5 kHz	78.1 kHz	9

				has no meaning. With the I²C bus format, this bit is when the SDA level changes from high to low und condition of SCL = high, assuming that the start co has been issued. This bit is cleared to 0 when the level changes from low to high under the condition = high, assuming that the stop condition has been Write 1 to BBSY and 0 to SCP to issue a start con Follow this procedure when also re-transmitting a condition. Write 0 in BBSY and 0 in SCP to issue a condition. To issue start/stop conditions, use the M instruction.
6	SCP	1	R/W	Start/Stop Issue Condition Disable The SCP bit controls the issue of start/stop conditi master mode. To issue a start condition, write 1 in BBSY and 0 in retransmit start condition is issued in the same wa issue a stop condition, write 0 in BBSY and 0 in SC bit is always read as 1. If 1 is written, the data is n
5	SDAO	1	R/W	SDA Output Value Control This bit is used with SDAOP when modifying outpu SDA. This bit should not be manipulated during tra 0: When reading, SDA pin outputs low. When writing, SDA pin is changed to output low 1: When reading, SDA pin outputs high. When writing, SDA pin is changed to output Hi-Z (outputs high by external pull-up resistance).

				This bit is always read as 1, and cannot be modif
1	IICRST	0	R/W	IIC Control Part Reset This bit resets the control part except for I²C2 reg this bit is set to 1 when hang-up occurs because communication failure during I²C2 operation, I²C2 part can be reset without setting ports and initializ registers.
0	—	1	—	Reserved This bit is always read as 1, and cannot be modif

14.3.3 I²C Bus Mode Register (ICMR)

ICMR selects whether the MSB or LSB is transferred first, performs master mode wait c and selects the transfer bit count.

Bit	Bit Name	Initial Value	R/W	Description
7	MLS	0	R/W	MSB-First/LSB-First Select 0: MSB-first 1: LSB-first Set this bit to 0 when the I²C bus format is used.

5, 4	—	All 1	—	Reserved These bits are always read as 1, and cannot be m
3	BCWP	1	R/W	BC Write Protect This bit controls the BC2 to BC0 modifications. Wh modifying BC2 to BC0, this bit should be cleared t use the MOV instruction. In clock synchronous ser mode, BC should not be modified. 0: When writing, values of BC2 to BC0 are set. 1: When reading, 1 is always read. When writing, settings of BC2 to BC0 are inval

serial format, these bits should not be modified.

I^2C Bus Format	Clock Synchronous Serial F
000: 9 bits	000: 8 bits
001: 2 bits	001: 1 bits
010: 3 bits	010: 2 bits
011: 4 bits	011: 3 bits
100: 5 bits	100: 4 bits
101: 6 bits	101: 5 bits
110: 7 bits	110: 6 bits
111: 8 bits	111: 7 bits

				1: Transmit data empty interrupt request (TXI) is e
6	TEIE	0	R/W	Transmit End Interrupt Enable This bit enables or disables the transmit end interr at the rising of the ninth clock while the TDRE bit i is 1. TEI can be canceled by clearing the TEND bi TEIE bit to 0. 0: Transmit end interrupt request (TEI) is disabled 1: Transmit end interrupt request (TEI) is enabled.
5	RIE	0	R/W	Receive Interrupt Enable This bit enables or disables the receive data full in request (RXI) and the overrun error interrupt reque with the clocked synchronous format, when a rece is transferred from ICDRS to ICDRR and the RDR ICSR is set to 1. RXI can be canceled by clearing RDRF or RIE bit to 0. 0: Receive data full interrupt request (RXI) and ov error interrupt request (ERI) with the clocked synchronous format are disabled. 1: Receive data full interrupt request (RXI) and ov error interrupt request (ERI) with the clocked synchronous format are enabled.

3	STIE	0	R/W	Stop Condition Detection Interrupt Enable 0: Stop condition detection interrupt request (STP disabled. 1: Stop condition detection interrupt request (STP enabled.
2	ACKE	0	R/W	Acknowledge Bit Judgement Select 0: The value of the receive acknowledge bit is ign continuous transfer is performed. 1: If the receive acknowledge bit is 1, continuous halted.
1	ACKBR	0	R	Receive Acknowledge In transmit mode, this bit stores the acknowledge are returned by the receive device. This bit canno modified. 0: Receive acknowledge = 0 1: Receive acknowledge = 1
0	ACKBT	0	R/W	Transmit Acknowledge In receive mode, this bit specifies the bit to be se acknowledge timing. 0: 0 is sent at the acknowledge timing. 1: 1 is sent at the acknowledge timing.

				• When TRS is set • When a start condition (including re-transfer) h issued • When transmit mode is entered from receive m slave mode [Clearing conditions] • When 0 is written in TDRE after reading TDRE • When data is written to ICDRT with an instruct
6	TEND	0	R/W	Transmit End [Setting conditions] • When the ninth clock of SCL rises with the I^2C format while the TDRE flag is 1 • When the final bit of transmit frame is sent with clock synchronous serial format [Clearing conditions] • When 0 is written in TEND after reading TEND • When data is written to ICDRT with an instruct
5	RDRF	0	R/W	Receive Data Register Full [Setting condition] • When a receive data is transferred from ICDR ICDRR [Clearing conditions] • When 0 is written in RDRF after reading RDRF • When ICDRR is read with an instruction

				• In master mode, when a stop condition is dete frame transfer • In slave mode, when a stop condition is detec the general call address or the first byte slave next to detection of start condition, accords w address set in SAR [Clearing condition] • When 0 is written in STOP after reading STO
2	AL/OVE	0	R/W	Arbitration Lost Flag/Overrun Error Flag This flag indicates that arbitration was lost in mas with the I^2C bus format and that the final bit has b received while RDRF = 1 with the clocked synchr format. When two or more master devices attempt to sei at nearly the same time, if the I^2C bus interface d data differing from the data it sent, it sets AL to 1 indicate that the bus has been taken by another r [Setting conditions] • If the internal SDA and SDA pin disagree at th SCL in master transmit mode • When the SDA pin outputs high in master mo start condition is detected • When the final bit is received with the clocked synchronous format while RDRF = 1 [Clearing condition] • When 0 is written in AL/OVE after reading AL

				receive mode. [Clearing condition] • When 0 is written in AAS after reading AAS=1
0	ADZ	0	R/W	General Call Address Recognition Flag This bit is valid in I^2C bus format slave receive mo [Setting condition] • When the general call address is detected in s receive mode [Clearing condition] • When 0 is written in ADZ after reading ADZ=1

14.3.6 Slave Address Register (SAR)

SAR selects the communication format and sets the slave address. When the chip is in sla with the I^2C bus format, if the upper 7 bits of SAR match the upper 7 bits of the first fram received after a start condition, the chip operates as the slave device.

0: I²C bus format is selected.

1: Clocked synchronous serial format is selected

14.3.7 I²C Bus Transmit Data Register (ICDRT)

ICDRT is an 8-bit readable/writable register that stores the transmit data. When ICDRT space in the shift register (ICDRS), it transfers the transmit data which is written in ICD ICDRS and starts transferring data. If the next transfer data is written to ICDRT during transferring data of ICDRS, continuous transfer is possible. If the MLS bit of ICMR is s when the data is written to ICDRT, the MSB/LSB inverted data is read. The initial value ICDRT is H′FF.

14.3.8 I²C Bus Receive Data Register (ICDRR)

ICDRR is an 8-bit register that stores the receive data. When data of one byte is received transfers the receive data from ICDRS to ICDRR and the next data can be received. ICD receive-only register, therefore the CPU cannot write to this register. The initial value of H′FF.

14.3.9 I²C Bus Shift Register (ICDRS)

ICDRS is a register that is used to transfer/receive data. In transmission, data is transferr ICDRT to ICDRS and the data is sent from the SDA pin. In reception, data is transferred ICDRS to ICDRR after data of one byte is received. This register cannot be read directly CPU.

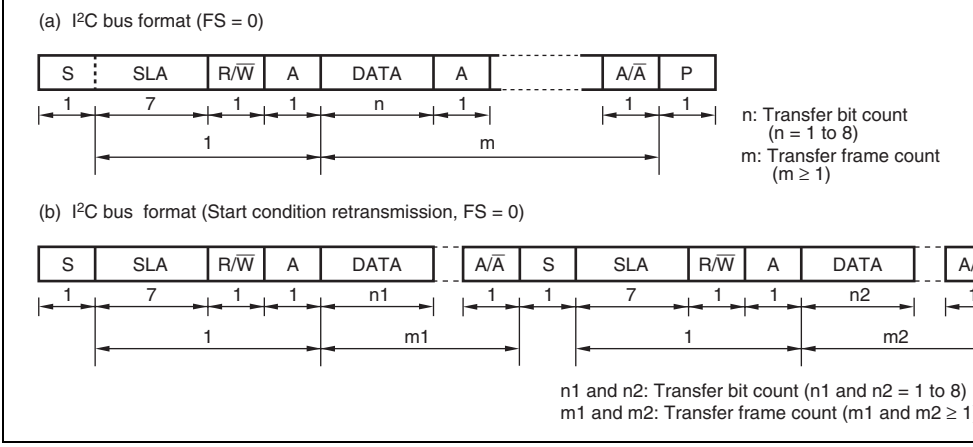


Figure 14.3 I²C Bus Formats

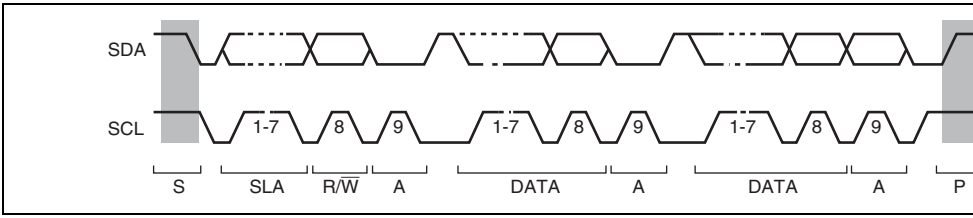


Figure 14.4 I²C Bus Timing

In master transmit mode, the master device outputs the transmit clock and transmit data, slave device returns an acknowledge signal. For master transmit mode operation timing, figures 14.5 and 14.6. The transmission procedure and operations in master transmit mo described below.

1. Set the ICE bit in ICCR1 to 1. Set the MLS and WAIT bits in ICMR and the CKS3 t bits in ICCR1 to 1. (Initial setting)
2. Read the BBSY flag in ICCR2 to confirm that the bus is free. Set the MST and TRS ICCR1 to select master transmit mode. Then, write 1 to BBSY and 0 to SCP using M instruction. (Start condition issued) This generates the start condition.
3. After confirming that TDRE in ICSR has been set, write the transmit data (the first b show the slave address and R/$\overline{W}$) to ICDRT. At this time, TDRE is automatically cle and data is transferred from ICDRT to ICDRS. TDRE is set again.
4. When transmission of one byte data is completed while TDRE is 1, TEND in ICSR i at the rise of the 9th transmit clock pulse. Read the ACKBR bit in ICIER, and confir slave device has been selected. Then, write second byte data to ICDRT. When ACK the slave device has not been acknowledged, so issue the stop condition. To issue the condition, write 0 to BBSY and SCP using MOV instruction. SCL is fixed low until transmit data is prepared or the stop condition is issued.
5. The transmit data after the second byte is written to ICDRT every time TDRE is set.
6. Write the number of bytes to be transmitted to ICDRT. Wait until TEND is set (the e byte data transmission) while TDRE is 1, or wait for NACK (NACKF in ICSR = 1) receive device while ACKE in ICIER is 1. Then, issue the stop condition to clear TE NACKF.
7. When the STOP bit in ICSR is set to 1, the operation returns to the slave receive mo

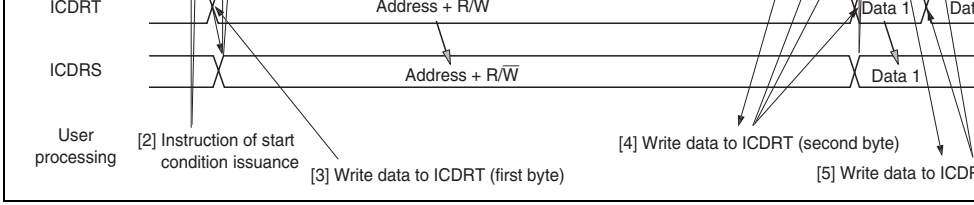


Figure 14.5 Master Transmit Mode Operation Timing (1)

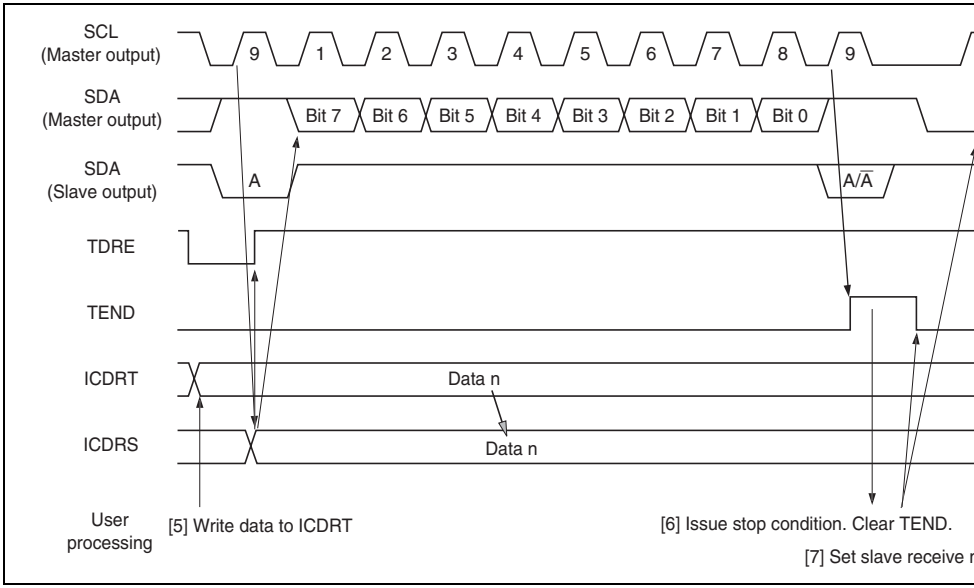


Figure 14.6 Master Transmit Mode Operation Timing (2)

level specified by ACKBT in ICIER to SDA, at the 9th receive clock pulse.

3. After the reception of first frame data is completed, the RDRF bit in ICST is set to 1 of 9th receive clock pulse. At this time, the receive data is read by reading ICDRR, a is cleared to 0.
4. The continuous reception is performed by reading ICDRR every time RDRF is set. I RDRF is set to 1, the reading of ICDRR is delayed by other processing and does not the falling edge of the 8th clock pulse, set RCVD to 1 and perform one-byte data tra
5. If next frame is the last receive data, set the RCVD bit in ICCR1 to 1 before reading This enables the issuance of the stop condition after the next reception.
6. When the RDRF bit is set to 1 at rise of the 9th receive clock pulse, issue the stage c
7. When the STOP bit in ICSR is set to 1, read ICDRR. Then clear the RCVD bit to 0.
8. The operation returns to the slave receive mode.

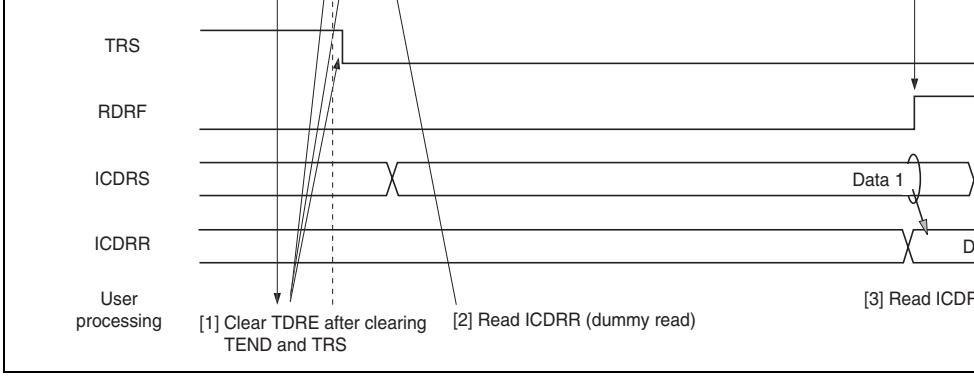


Figure 14.7 Master Receive Mode Operation Timing (1)

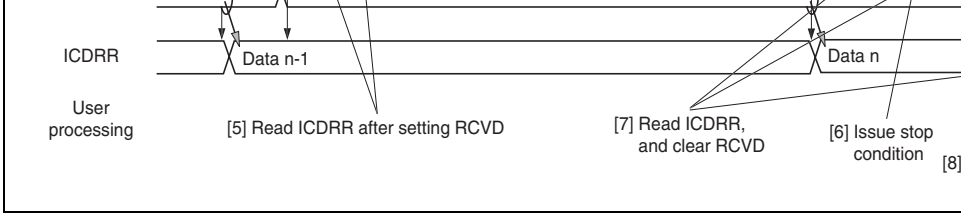


Figure 14.8 Master Receive Mode Operation Timing (2)

14.4.4 Slave Transmit Operation

In slave transmit mode, the slave device outputs the transmit data, while the master devi the receive clock and returns an acknowledge signal. For slave transmit mode operation see figures 14.9 and 14.10.

The transmission procedure and operations in slave transmit mode are described below.

1. Set the ICE bit in ICCR1 to 1. Set the MLS and WAIT bits in ICMR and the CKS3 t bits in ICCR1 to 1. (Initial setting) Set the MST and TRS bits in ICCR1 to select sla mode, and wait until the slave address matches.
2. When the slave address matches in the first frame following detection of the start co the slave device outputs the level specified by ACKBT in ICIER to SDA, at the rise clock pulse. At this time, if the 8th bit data (R/$\overline{W}$) is 1, the TRS and ICSR bits in IC set to 1, and the mode changes to slave transmit mode automatically. The continuous transmission is performed by writing transmit data to ICDRT every time TDRE is se
3. If TDRE is set after writing last transmit data to ICDRT, wait until TEND in ICSR i with TDRE = 1. When TEND is set, clear TEND.
4. Clear TRS for the end processing, and read ICDRR (dummy read). SCL is free.
5. Clear TDRE.

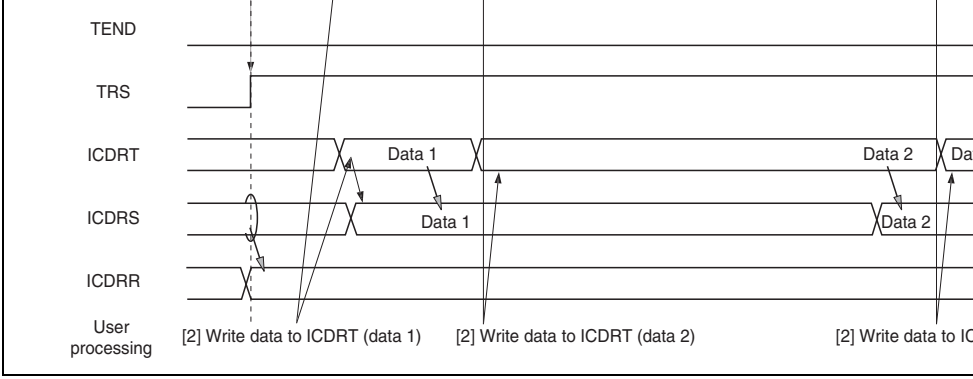


Figure 14.9 Slave Transmit Mode Operation Timing (1)

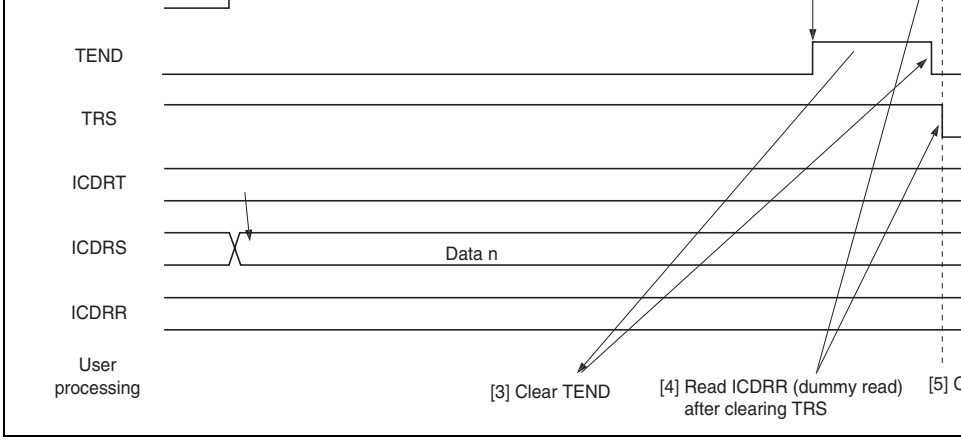


Figure 14.10 Slave Transmit Mode Operation Timing (2)

14.4.5 Slave Receive Operation

In slave receive mode, the master device outputs the transmit clock and transmit data, ar slave device returns an acknowledge signal. For slave receive mode operation timing, se 14.11 and 14.12. Since a flag may be set according to the port state, initialization should performed after all bits in ICSR have been cleared. The reception procedure and operati slave receive mode are described below.

1. Set the ICE bit in ICCR1 to 1. Set the MLS and WAIT bits in ICMR and the CKS3 t bits in ICCR1 to 1. (Initial setting) Set the MST and TRS bits in ICCR1 to select sla mode, and wait until the slave address matches.
2. When the slave address matches in the first frame following detection of the start co the slave device outputs the level specified by ACKBT in ICIER to SDA, at the rise

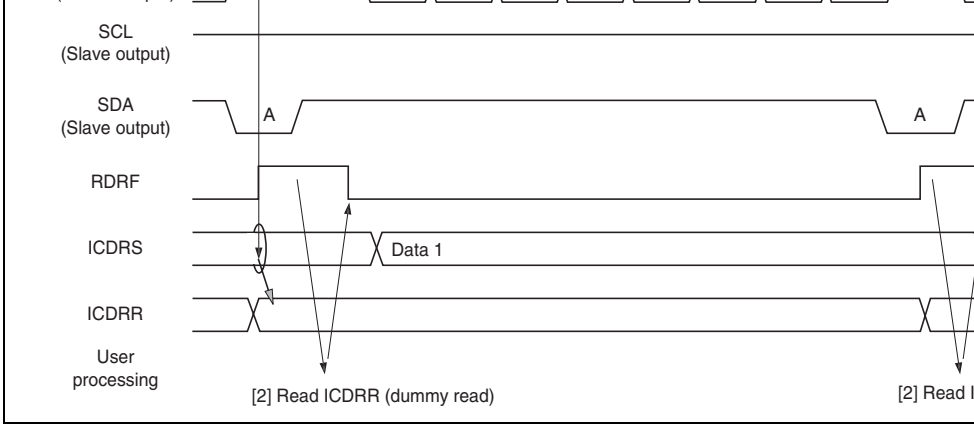


Figure 14.11 Slave Receive Mode Operation Timing (1)

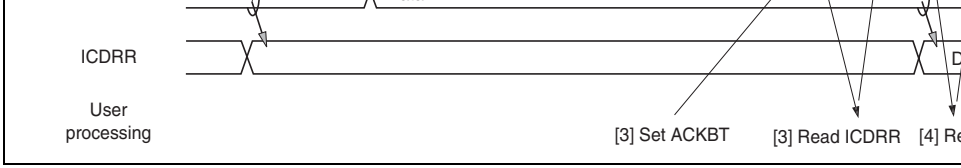


Figure 14.12 Slave Receive Mode Operation Timing (2)

14.4.6 Clocked Synchronous Serial Format

This module can be operated with the clocked synchronous serial format, by setting the
SAR to 1. When the MST bit in ICCR1 is 1, the transfer clock output from SCL is selec
MST is 0, the external clock input is selected.

Data Transfer Format: Figure 14.13 shows the clocked synchronous serial transfer for

The transfer data is output from the rise to the fall of the SCL clock, and the data at the r
of the SCL clock is guaranteed. The MLS bit in ICMR sets the order of data transfer, in
MSB first or LSB first. The output level of SDA can be changed during the transfer wai
SDAO bit in ICCR2.

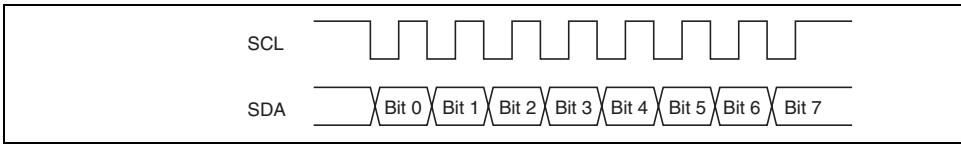


Figure 14.13 Clocked Synchronous Serial Transfer Format

Transmit Operation: In transmit mode, transmit data is output from SDA, in synchron
with the fall of the transfer clock. The transfer clock is output when MST in ICCR1 is 1.

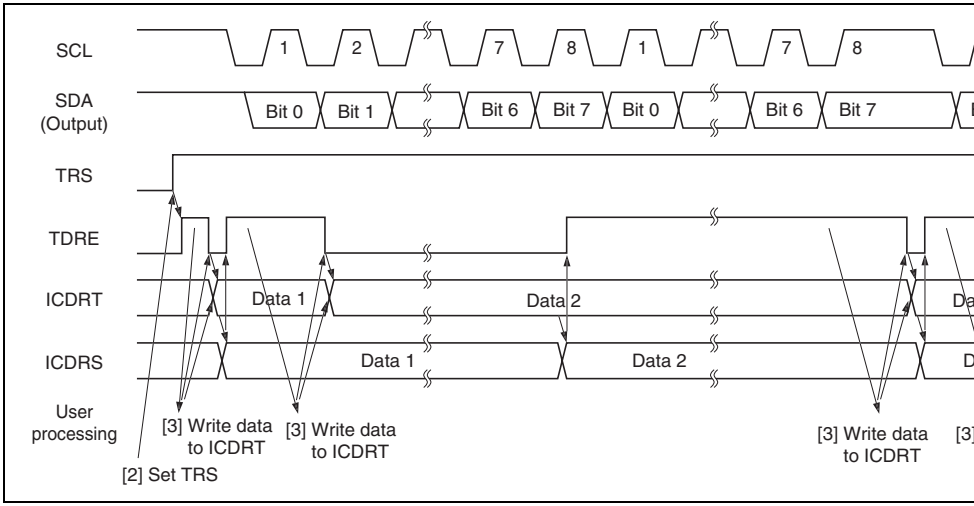


Figure 14.14 Transmit Mode Operation Timing

Receive Operation: In receive mode, data is latched at the rise of the transfer clock. The clock is output when MST in ICCR1 is 1, and is input when MST is 0. For receive mode timing, see figure 14.15. The reception procedure and operations in receive mode are des below.

1. Set the ICE bit in ICCR1 to 1. Set the MST and CKS3 to CKS0 bits in ICCR1 to 1. (I setting)
2. When the transfer clock is output, set MST to 1 to start outputting the receive clock.
3. When the receive operation is completed, data is transferred from ICDRS to ICDRR a RDRF in ICSR is set. When MST = 1, the next byte can be received, so the clock is continually output. The continuous reception is performed by reading ICDRR every ti

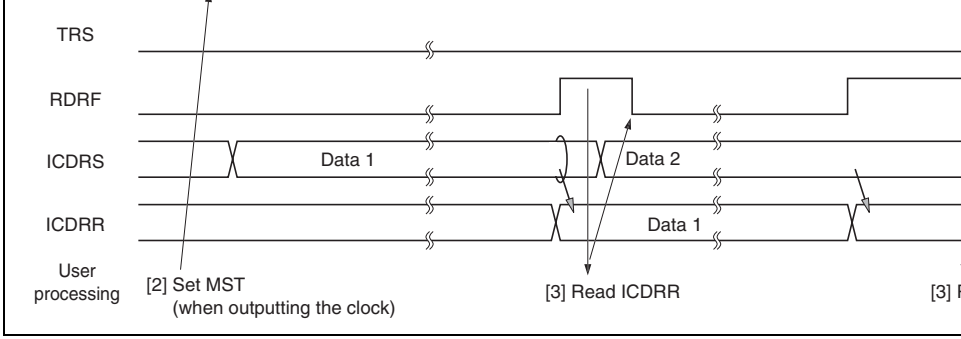


Figure 14.15 Receive Mode Operation Timing

14.4.7 Noise Canceler

The logic levels at the SCL and SDA pins are routed through noise cancelers before bei
internally. Figure 14.16 shows a block diagram of the noise canceler circuit.

The noise canceler consists of two cascaded latches and a match detector. The SCL (or S
input signal is sampled on the system clock, but is not passed forward to the next circuit
outputs of both latches agree. If they do not agree, the previous value is held.

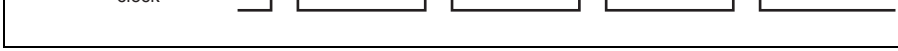


Figure 14.16 Block Diagram of Noise Canceler

14.4.8 Example of Use

Flowcharts in respective modes that use the I^2C bus interface are shown in figures 14.17 t

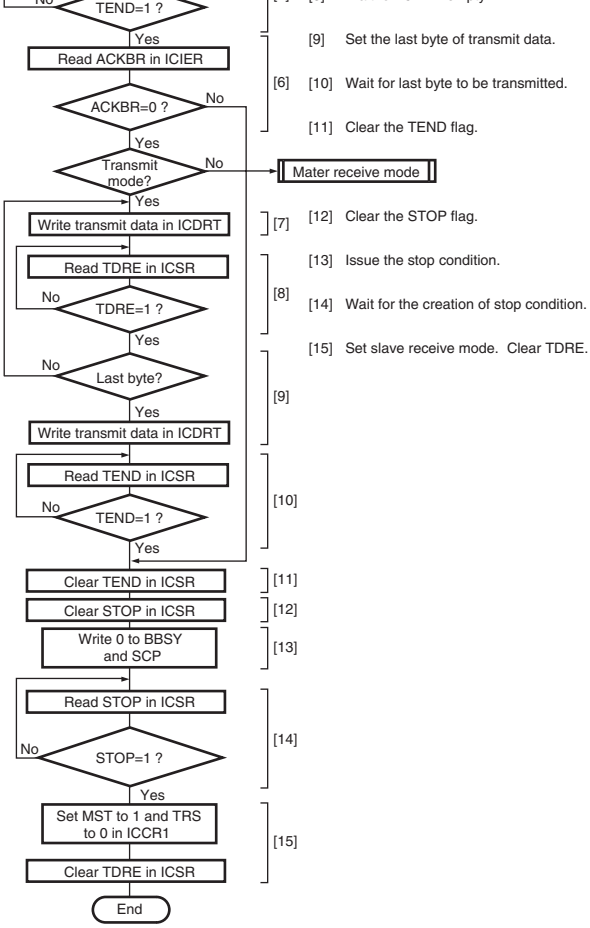
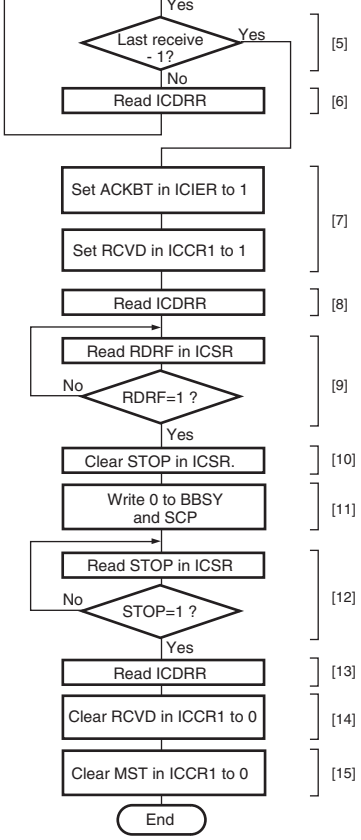


Figure 14.17 Sample Flowchart for Master Transmit Mode



[9] Wait for the last byte to be receive.

[10] Clear the STOP flag.

[11] Issue the stop condition.

[12] Wait for the creation of stop condition.

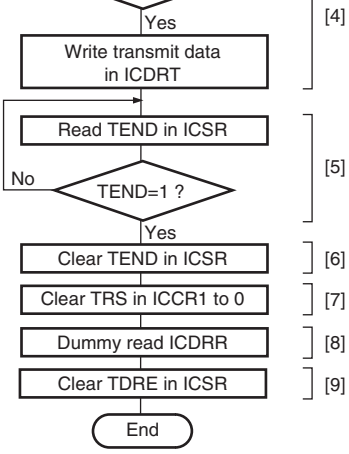
[13] Read the last byte of receive data.

[14] Clear RCVD.

[15] Set slave receive mode.

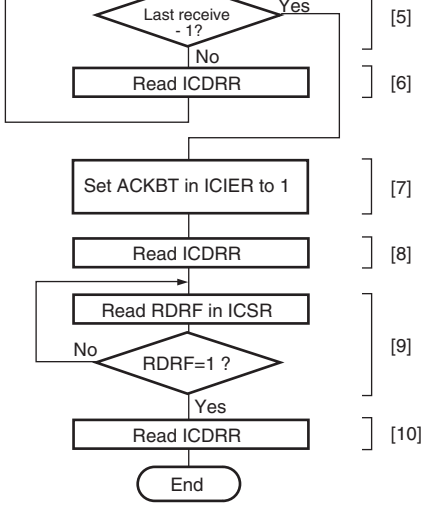
Notes: When receiving one byte, execute step [7] after step [1] without ex
steps [2] to [6]. The step [8] is ICDRR dummy read.
* Do not activate an interrupt during the execution of steps [1] to [3

Figure 14.18 Sample Flowchart for Master Receive Mode



[9] Clear the TDRE flag.

Figure 14.19 Sample Flowchart for Slave Transmit Mode



[8] Read the (last byte - 1) of receive data.

[9] Wait the last byte to be received.

[10] Read for the last byte of receive data.

Note: When receiving one byte, execute step [7] after step [1] witho executing steps [2] to [6]. The step [8] is ICDRR dummy read.

Figure 14.20 Sample Flowchart for Slave Receive Mode

Transmit data empty	TXI	(TDRE=1) • (TIE=1)	O	O
Transmit end	TEI	(TEND=1) • (TEIE=1)	O	O
Receive data full	RXI	(RDRF=1) • (RIE=1)	O	O
STOP recognition	STPI	(STOP=1) • (STIE=1)	O	X
NACK receive	NAKI	{(NACKF=1)+(AL=1)} • (NAKIE=1)	O	X
Arbitration lost/overrun Error			O	O

When interrupt conditions described in table 14.3 are 1 and the I bit in CCR is 0, the CP executes an interrupt exception processing. Interrupt sources should be cleared in the ex processing. TDRE and TEND are automatically cleared to 0 by writing the transmit data ICDRT. RDRF are automatically cleared to 0 by reading ICDRR. TDRE is set to 1 agai same time when transmit data is written to ICDRT. When TDRE is cleared to 0, then an data of one byte may be transmitted.

Figure 14.21 shows the timing of the bit synchronous circuit and table 14.4 shows the tim
SCL output changes from low to Hi-Z then SCL is monitored.

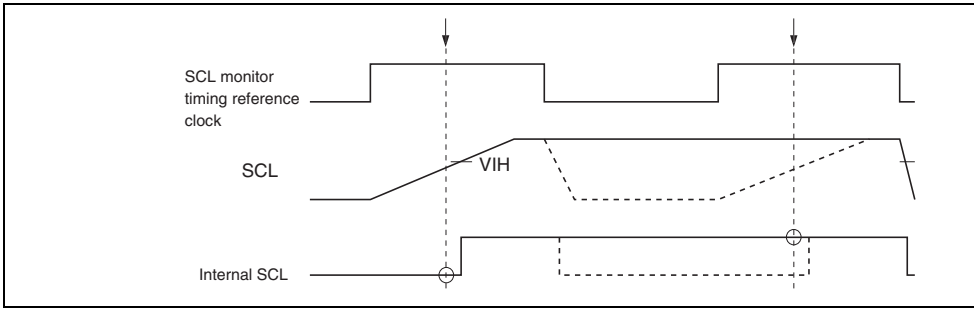


Figure 14.21 The Timing of the Bit Synchronous Circuit

Table 14.4 Time for Monitoring SCL

CKS3	CKS2	Time for Monitoring SCL
0	0	7.5 tcyc
	1	19.5 tcyc
1	0	14.5 tcyc
	1	41.5 tcyc

Confirm the ninth falling edge of the clock before issuing a stop or a repeated start cond

The ninth falling edge can be confirmed by monitoring the SCLO bit in the I²C bus cont
2 (ICCR2).

If a stop or a repeated start condition is issued at a certain timing in either of the followi
the stop or repeated start condition may be issued incorrectly.

- The rising time of the SCL signal exceeds the time given in section 14.6, Bit Synchr
  Circuit, because of the load on the SCL bus.
- The bit synchronous circuit is activated because a slave device holds the SCL bus lo
  the eighth clock.

14.7.3 WAIT Bit in I²C Bus Mode Register (ICMR)

The WAIT bit in the I²C bus mode register (ICMR) must be held 0.

If the WAIT bit is set to 1, when a slave device holds the SCL signal low more than one
clock cycle during the eights clock, the high level period of the ninth clock may be short
given period.

- Set RCVD to 1 in master receive mode and perform communication in units of one by

14.7.5 Restriction on Setting of Transfer Rate in Use of Multi-Master

In multi-master usage when the IIC transfer rate setting of this LSI is lower than those of masters, unexpected length of SCL may occasionally be output. To avoid this, the specifi must be greater than or equal to the value produced by multiplying the fastest transfer rat the other masters by 1/1.8. For example, when the transfer rate of the fastest bus master a other bus masters is 400 kbps, the transfer rate of the IIC of this LSI must be set to at leas kbps (= 400/1.8).

14.7.6 Restriction on Use of Bit Manipulation Instructions to Set MST and TRS v Multi-Master Is Used

When master transmission is selected by consecutively manipulating the MST and TRS b multi-master usage, an arbitration loss during execution of the bit-manipulation instructio TRS leads to the contradictory situation where AL in ICSR is 1 in master transmit mode (1, TRS = 1).

Ways to avoid this effect are listed below.

- Use the MOV instruction to set MST and TRS in multi-master usage.
- When arbitration is lost, confirm that MST = 0 and TRS = 0. If the setting of MST = 0 TRS = 0 is not confirmed, set MST = 0 and TRS = 0 again.

- Conversion time: 13.3 µs per channel (at 20-MHz operation), 10.1 µs per channel (at operation)
- Two operating modes
  Single mode: Single-channel A/D conversion
  Scan mode: Continuous A/D conversion on 1 to 4 channels
- Four data registers
  Conversion results are held in a 16-bit data register for each channel
- Sample and hold function
- Three methods conversion start
  Software
  16-bit timer pulse unit (TPU or TMR) conversion start trigger
  External trigger signal
- Interrupt request
  An A/D conversion end interrupt request (ADI) can be generated
- Module stop mode can be set
- Selectable range of voltages of analog inputs
  The range of voltages of analog inputs to be converted can be specified using the Vr as the analog reference voltage.

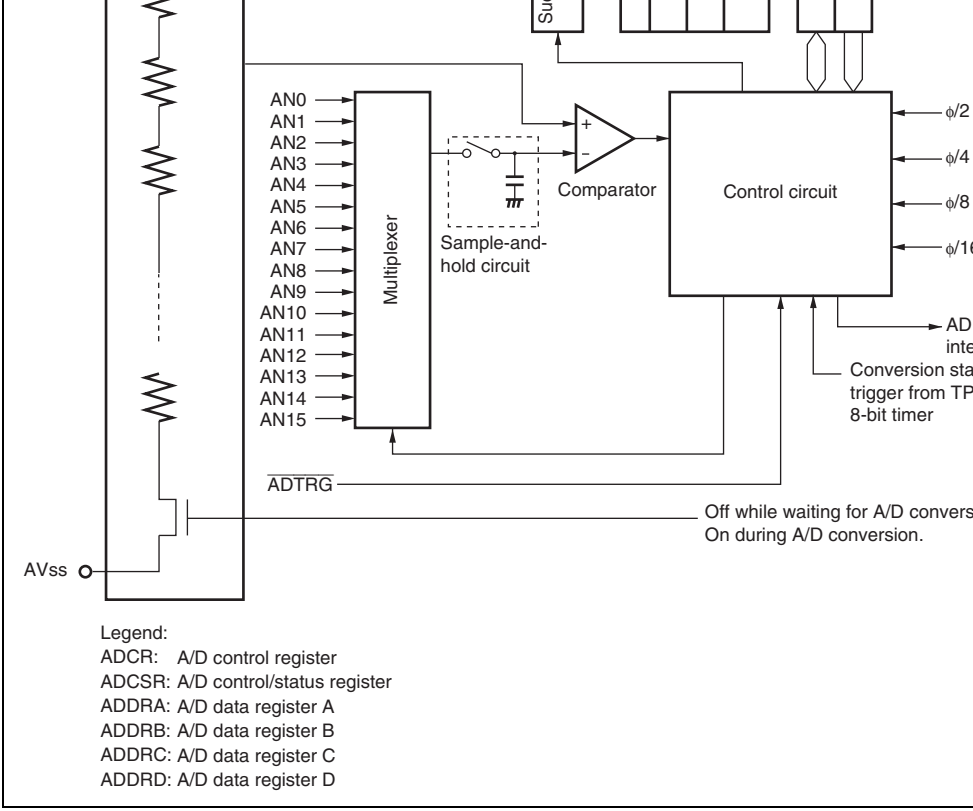


Figure 15.1 Block Diagram of A/D Converter

Pin Name	Symbol	I/O	Function
Analog power supply pin	AV_{CC}	Input	Analog block power supply pin
Analog ground pin	AV_{SS}	Input	Analog block ground and referenc
Reference voltage pin	Vref	Input	Reference voltage for A/D convers
Analog input pin 0	AN0	Input	Channel set 0 (CH3 = 0), group 0 input pins
Analog input pin 1	AN1	Input	
Analog input pin 2	AN2	Input	
Analog input pin 3	AN3	Input	
Analog input pin 4	AN4	Input	Channel set 0 (CH3 = 0), group 1 input pins
Analog input pin 5	AN5	Input	
Analog input pin 6	AN6	Input	
Analog input pin 7	AN7	Input	
Analog input pin 8	AN8	Input	Channel set 1 (CH3 = 1), group 0 input pins
Analog input pin 9	AN9	Input	
Analog input pin 10	AN10	Input	
Analog input pin 11	AN11	Input	
Analog input pin 12	AN12	Input	Channel set 1 (CH3 = 1), group 1 input pins
Analog input pin 13	AN13	Input	
Analog input pin 14	AN14	Input	
Analog input pin 15	AN15	Input	
A/D external trigger input	$\overline{\text{ADTRG}}$	Input	External trigger input pin for startin conversion

- A/D control register (ADCR)

15.3.1 A/D Data Registers A to D (ADDRA to ADDRD)

There are four 16-bit read-only ADDR registers; ADDRA to ADDRD, used to store the r
A/D conversion. The ADDR registers, which store a conversion result for each analog inp
channel, are shown in table 15.2.

The converted 10-bit data is stored in bits 6 to 15. The lower 6 bits are always read as 0.

The data bus between the CPU and the A/D converter is 8 bits wide. The upper byte can b
directly from the CPU, however the lower byte should be read via a temporary register. T
when reading ADDR, read only the upper byte, or read in word unit.

Table 15.2 Analog Input Channels and Corresponding ADDR Registers

Analog Input Channel				
Channel Set 0 (CH3 = 0)		Channel Set 1 (CH3 = 1)		A/D Data Registe
Group 0 (CH2 = 0)	Group 1 (CH2 = 1)	Group 0 (CH2 = 0)	Group 1 (CH2 = 1)	Stored Results o Conversion
AN0	AN4	AN8	AN12	ADDRA
AN1	AN5	AN9	AN13	ADDRB
AN2	AN6	AN10	AN14	ADDRC
AN3	AN7	AN11	AN15	ADDRD

				• When A/D conversion ends on all specified ch scan mode [Clearing conditions] • When 0 is written after reading ADF = 1 • When the DTC is activated by an ADI interrup DISEL bit in DTC is 0 with the transfer counte than 0
6	ADIE	0	R/W	A/D Interrupt Enable A/D conversion end interrupt (ADI) request enabl is set
5	ADST	0	R/W	A/D Start Clearing this bit to 0 stops A/D conversion, and th converter enters the wait state. Setting this bit to 1 starts A/D conversion. In singl this bit is cleared to 0 automatically when convers the specified channel is complete. In scan mode, conversion continues sequentially on the specifie channels until this bit is cleared to 0 by software, a transition to software standby mode, hardware mode, or module stop mode. The ADST bit can be set to 1 by software, a time conversion start trigger, or the A/D external trigge ($\overline{\text{ADTRG}}$).

Switches the analog channel allocated to group 0 a
group 1.

	Group 0	Group 1
0: Channel set 0	AN0 to AN3	AN4 to AN7
1: Channel set 1	AN 8 to AN11	AN12 to AN15

	100: AN4	100: AN4
	101: AN5	101: AN4, AN5
	110: AN6	110: AN4 to AN6
	111: AN7	111: AN4 to AN7
	Channel set 1 (CH3 = 1)	
	000: AN8	000: AN8
	001: AN9	001: AN8, AN9
	010: AN10	010: AN8 to AN10
	011: AN11	011: AN8 to AN11
	100: AN12	100: AN12
	101: AN13	101: AN12, AN13
	110: AN14	110: AN12 to AN14
	111: AN15	111: AN12 to AN15

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

				01: A/D conversion start by TPU conversion start t enabled
				10: A/D conversion start by 8-bit timer conversion trigger is enabled
				11: A/D conversion start by external trigger pin (AD enabled
5, 4	—	All 1	—	Reserved
				These bits are always read as 1 and cannot be mo
3	CKS1	0	R/W	Clock Select 0 and 1
2	CKS0	0	R/W	These bits specify the A/D conversion time. The conversion time should be changed only when the conversion stops (ADST = 0).
				The conversion time setting should exceed the co time shown in section 24.5, A/D Converter Charac
				00: Conversion time = 530 states (max.)
				01: Conversion time = 266 states (max.)
				10: Conversion time = 134 states (max.)
				11: Conversion time = 68 states (max.)
1, 0	—	All 1	—	Reserved
				These bits are always read as 1 and cannot be mo

order. When only the upper-byte data is read, the data is guaranteed. However, when on
lower-byte data is read, the data is not guaranteed.

Figure 15.2 shows data flow when accessing to ADDR.

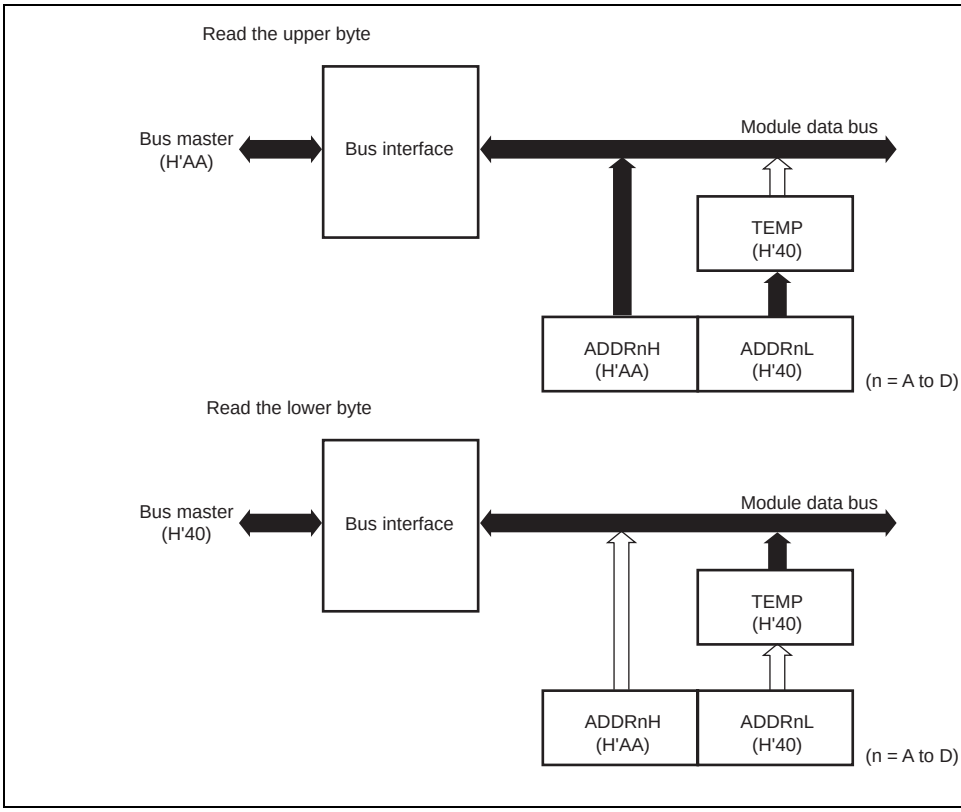


Figure 15.2 Access to ADDR (When Reading H'AA40)

operations are as follows.

1. A/D conversion is started when the ADST bit is set to 1, according to software, timer
   conversion start trigger, or external trigger input.
2. When A/D conversion is completed, the result is transferred to the corresponding A/D
   register to the channel.
3. On completion of conversion, the ADF bit in ADCSR is set to 1. If the ADIE bit is se
   this time, an ADI interrupt request is generated.
4. The ADST bit remains set to 1 during A/D conversion. When A/D conversion ends, t
   bit is automatically cleared to 0 and the A/D converter enters the wait state.

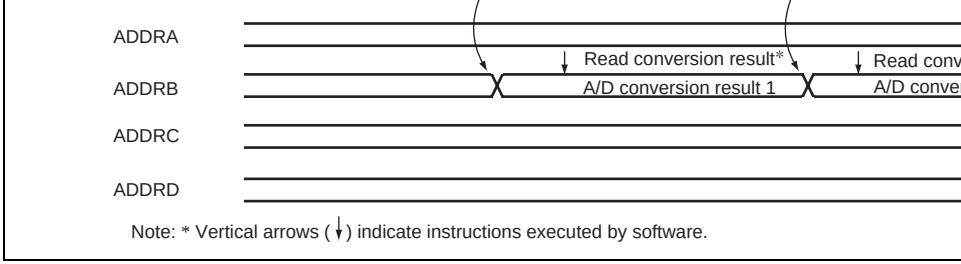
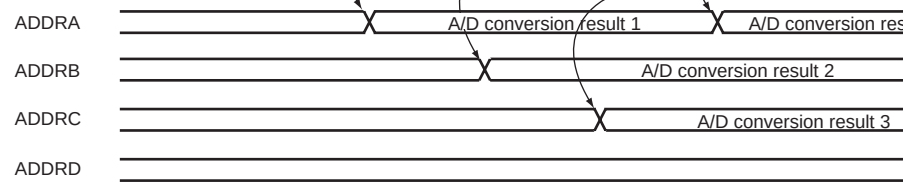


Figure 15.3 A/D Conversion Timing (Single-Chip Mode, Channel 1 Selecte

15.5.2 Scan Mode

In scan mode, A/D conversion is to be performed sequentially on the specified channels channels maximum). The operations are as follows.

1. When the ADST bit is set to 1 by software, timer conversion start trigger, or externa input, A/D conversion starts on the first channel in the group (AN0 when CH3 = 0 a 0, AN4 when CH3 = 0 and CH2 = 1, AN8 when CH3 = 1 and CH2 = 0, or AN12 wh 1 and CH2 = 1).
2. When A/D conversion for each channel is completed, the result is sequentially transf the A/D data register corresponding to each channel.
3. When conversion of all the selected channels is completed, the ADF flag is set to 1. ADIE bit is set to 1 at this time, an ADI interrupt is requested after A/D conversion e Conversion of the first channel in the group starts again.
4. Steps 2 to 3 are repeated as long as the ADST bit remains set to 1. When the ADST cleared to 0, A/D conversion stops and the A/D converter enters the wait state. After when the ADST bit is set to 1, conversion of the first channel in the group starts agai



Notes: 1. Vertical arrows (↓) indicate instructions executed by software.
2. Data currently being converted is ignored.

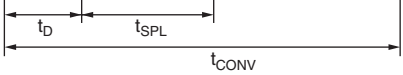
Figure 15.4 A/D Conversion Timing (Scan Mode, Channels AN0 to AN2 Selec

15.5.3 Input Sampling and A/D Conversion Time

The A/D converter has a built-in sample-and-hold circuit. The A/D converter samples the input when the A/D conversion start delay time (t_D) has passed after the ADST bit is set t starts conversion. Figure 15.5 shows the A/D conversion timing. Table 15.3 shows the A/ conversion time.

As indicated in figure 15.5, the A/D conversion time (t_{CONV}) includes t_D and the input samp (t_{SPL}). The length of t_D varies depending on the timing of the write access to ADCSR. The conversion time therefore varies within the ranges indicated in table 15.3. Specify the co time by setting bits CKS0 and CKS1 in ADCR with ADST cleared to 0. Note that the spe conversion time should be longer than the value described in section 24.5 A/D Conversio Characteristics.

In scan mode, the values given in table 15.3 apply to the first conversion time. The value table 15.4 apply to the second and subsequent conversions.



Legend:
(1) : ADCSR write cycle
(2) : ADCSR address
t_D : A/D conversion start delay
t_{SPL} : Input sampling time
t_{CONV} : A/D conversion time

Figure 15.5 A/D Conversion Timing

Table 15.3 A/D Conversion Time (Single Mode)

		CKS1 = 0						CKS1 = 1				
		CKS0 = 0			CKS0 = 1			CKS0 = 0			CK	
Item	**Symbol**	**Min.**	**Typ.**	**Max.**	**Min.**	**Typ.**	**Max.**	**Min.**	**Typ.**	**Max.**	**Min.**	
A/D conversion start delay time	t_D	18	—	33	10	—	17	6	—	9	4	
Input sampling time	t_{SPL}	—	127	—	—	63	—	—	31	—	—	
A/D conversion time	t_{CONV}	515	—	530	259	—	266	131	—	134	67	

Note: All values represent the number of states.

A/D conversion can be externally triggered. When the TRGS0 and TRGS1 bits are set to
ADCR, external trigger input is enabled at the $\overline{\text{ADTRG}}$ pin. A falling edge at the $\overline{\text{ADTRG}}$
the ADST bit to 1 in ADCSR, starting A/D conversion. Other operations, in both single a
modes, are the same as when the bit ADST has been set to 1 by software. Figure 15.6 sho
timing.

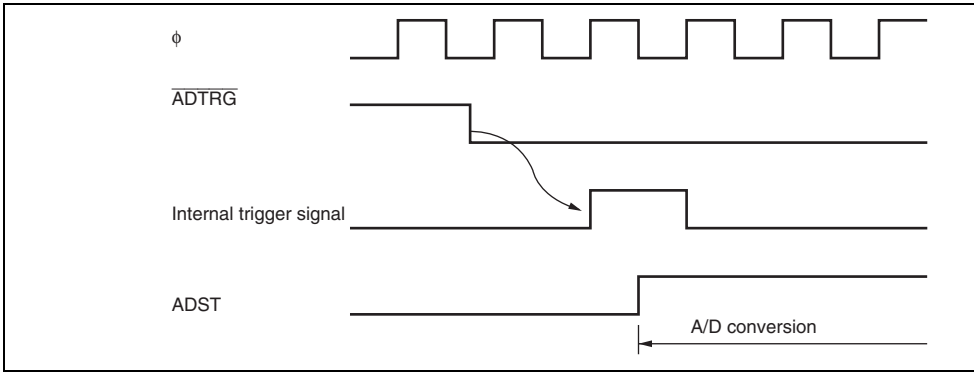


Figure 15.6 External Trigger Input Timing

ADI	A/D conversion end	ADF	Possible

15.7 A/D Conversion Accuracy Definitions

This LSI's A/D conversion accuracy definitions are given below.

- Resolution

  The number of A/D converter digital output codes
- Quantization error

  The deviation inherent in the A/D converter, given by 1/2 LSB (see figure 15.7).
- Offset error

  The deviation of the analog input voltage value from the ideal A/D conversion chara
  when the digital output changes from the minimum voltage value B'0000000000 (H'
  B'0000000001 (H'001) (see figure 15.8).
- Full-scale error

  The deviation of the analog input voltage value from the ideal A/D conversion chara
  when the digital output changes from B'1111111110 (H'3FE) to B'1111111111 (H'3
  figure 15.8).
- Nonlinearity error

  The error with respect to the ideal A/D conversion characteristic between zero voltag
  scale voltage. Does not include offset error, full-scale error, or quantization error (se
  15.8).
- Absolute accuracy

  The deviation between the digital value and the analog input value. Includes offset e
  scale error, quantization error, and nonlinearity error.

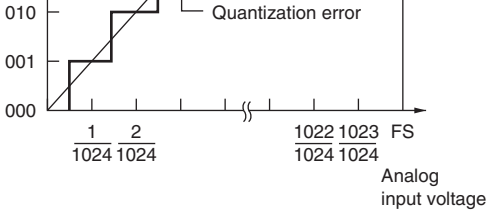


Figure 15.7 A/D Conversion Accuracy Definitions

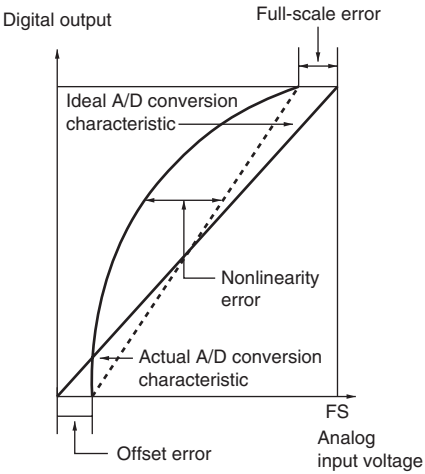


Figure 15.8 A/D Conversion Accuracy Definitions

This LSI's analog input is designed such that conversion accuracy is guaranteed for an i
for which the signal source impedance is 5 kΩ or less. This specification is provided to e
A/D converter's sample-and-hold circuit input capacitance to be charged within the samp
if the sensor output impedance exceeds 5 kΩ, charging may be insufficient and it may n
possible to guarantee A/D conversion accuracy. However, for A/D conversion in single
a large capacitance provided externally, the input load will essentially comprise only the
input resistance of 10 kΩ, and the signal source impedance is ignored. However, as a lo
filter effect is obtained in this case, it may not be possible to follow an analog signal wit
differential coefficient (e.g., 5 mV/µs or greater) (see figure 15.9). When converting a h
analog signal, a low-impedance buffer should be inserted.

15.8.3 Influences on Absolute Accuracy

Adding capacitance results in coupling with GND, and therefore noise in GND may adv
affect absolute accuracy. Be sure to make the connection to an electrically stable GND s
AVss.

Care is also required to insure that filter circuits do not communicate with digital signals
mounting board (i.e., acting as antennas).

15.8.4 Range of Analog Power Supply and Other Pin Settings

If the conditions below are not met, the reliability of the device may be adversely affecte

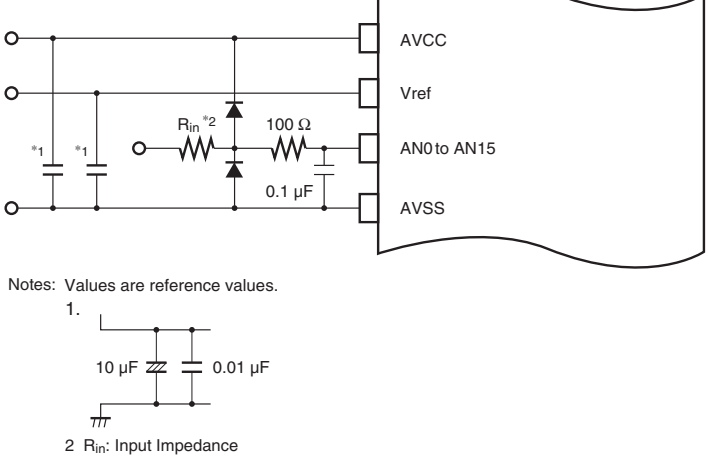
- Analog input voltage range
  The voltage applied to analog input pin ANn during A/D conversion should be in the
  AVss ≤ ANn ≤ AVcc.
- Relationship between AVcc, AVss and Vcc, Vss
  Set AVss = Vss as the relationship between AVcc, AVss and Vcc, Vss. If the A/D co
  not used, the AVcc and AVss pins must not be left open.
- Vref range
  The reference voltage input from the Vref pin should be set to AVcc or less.

15.8.5 Notes on Board Design

In board design, digital circuitry and analog circuitry should be as mutually isolated as po
and layout in which digital circuit signal lines and analog circuit signal lines cross or are
proximity should be avoided as far as possible. Failure to do so may result in incorrect op
of the analog circuitry due to inductance, adversely affecting A/D conversion values. Als
circuitry must be isolated from the analog input signals (AN0 to AN15), and analog powe
(AVcc) by the analog ground (AVss). Also, the analog ground (AVss) should be connecte
point to a stable digital ground (Vss) on the board.

15.8.6 Notes on Noise Countermeasures

A protection circuit should be connected in order to prevent damage due to abnormal volt
as an excessive surge at the analog input pins (AN0 to AN15), between AVcc and AVss,



Notes: Values are reference values.

1.

2 R_{in}: Input Impedance

Figure 15.10 Example of Analog Input Protection Circuit

Table 15.6 Analog Pin Specifications

Item	Min.	Max.	Unit
Analog input capacitance	—	20	pF
Permissible signal source impedance	—	5	kΩ

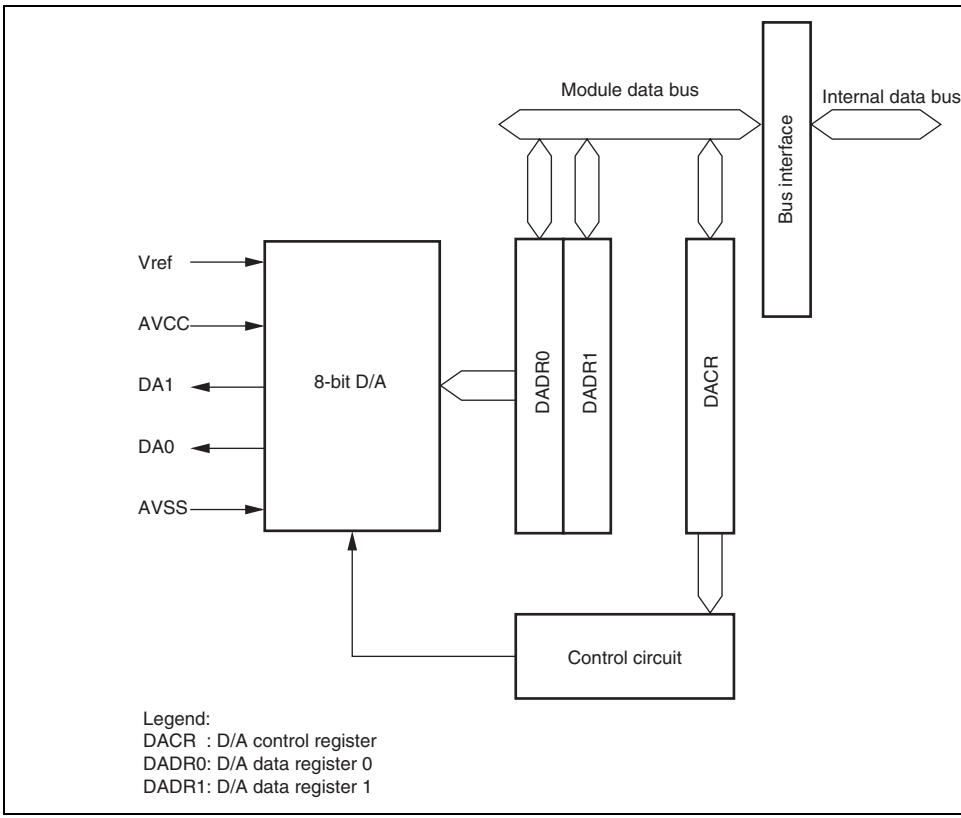


Figure 16.1 Block Diagram of D/A Converter

Analog output pin 1	DA1	Output	Channel 1 analog output pin
Reference voltage pin	Vref	Input	Analog block reference voltage

16.3 Register Descriptions

The D/A converter has the following registers. For details on the module stop control reg
section 22.1.2, Module Stop Control Registers A to C (MSTPCRA to MSTPCRC).

- D/A data register 0 (DADR0)
- D/A data register 1 (DADR1)
- D/A control register (DACR)

16.3.1 D/A Data Registers 0, 1 (DADR0, DADR1)

DADR0 and DADR1 are 8-bit readable/writable registers that store data for D/A convers
When analog output is permitted, D/A data register contents are converted and output to a
output pins.

				are enabled
6	DAOE0	0	R/W	D/A Output Enable 0 Controls D/A conversion and analog output 0: Analog output DA0 is disabled 1: D/A conversion for channel 0 and analog out are enabled
5	DAE	0	R/W	D/A Enable Controls D/A conversion in conjunction with the and DAOE1 bits. When the DAE bit is cleared t conversion for channels 0 and 1 are controlled individually. When DAE is set to 1, D/A convers channels 0 and 1 are controlled as one. Conve result output is controlled by the DAOE0 and D bits. For details, see table 16.2.
4 to 0	—	All 1	—	Reserved These bits are always read as 1 and cannot be

1	0	
	1	

16.4 Operation

Two channels of the D/A converter can perform conversion individually.

When the DAOE bit in DACR is set to 1, D/A conversion is enabled and the conversion r
output.

An example of D/A conversion of channel 0 is shown below. The operation timing is sho
figure 16.2.

1. Write conversion data to DADR0.
2. When the DAOE0 bit in DACR is set to 1, D/A conversion starts. After the interval o
   the conversion results are output from the analog output pin DA0. The conversion res
   output continuously until DADR0 is modified or DAOE0 bit is cleared to 0. The outp
   is calculated by the following formula:

   (DADR contents)/256 × Vref
3. Conversion starts immediately after DADR0 is modified. After the interval of t_{DCONV},
   conversion results are output.
4. When the DAOE bit is cleared to 0, analog output is disabled.

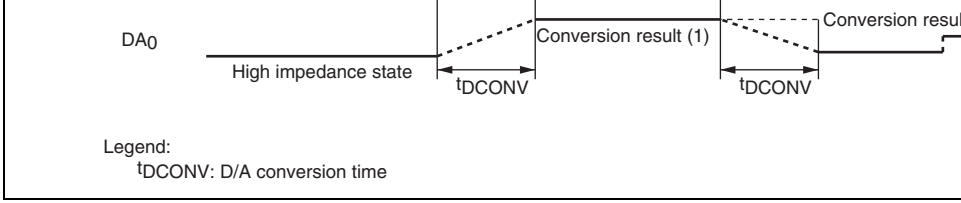


Figure 16.2 D/A Converter Operation Example

16.5 Usage Notes

16.5.1 Analog Power Supply Current in Power-Down Mode

If this LSI enters a power-down mode such as software standby, watch, and module stop while D/A conversion is enabled, the D/A cannot retain analog outputs within the given absolute accuracy although it retains digital values. The analog power supply current is approximately the same as that during D/A conversion. To reduce analog power supply power-down mode, clear the DAOE0, DAOE1 and DAE bits to 0 to disable D/A output entering the mode.

16.5.2 Setting for Module Stop Mode

It is possible to enable/disable the D/A converter operation using the module stop contro the D/A converter does not operate by the initial value of the register. The register can b by releasing the module stop mode. For more details, see section 22, Power-Down Mod

17.1 Features

- IEBus protocol control (layer 2) supported
  - Half duplex asynchronous communications
  - Multi-master system
  - Broadcast communications function
  - Selectable mode (three types) with different transfer speeds
- Data transfer by the data transfer controller (DTC)
  - Transfer buffer: 1 byte
  - Reception buffer: 1 byte
  - Up to 128 bytes of consecutive transfer/reception (maximum number of transfer
    mode 2)
- Operating frequency
  - 12 MHz, 12.58 MHz (IEB uses 1/2 divided external clock.)
  - 18 MHz, 18.87 MHz (IEB uses 1/3 divided external clock.)
  - 24 MHz, 25.16 MHz (IEB uses 1/4 divided external clock.)

Note: When selected communications mode 0 or mode 1 (± 1.5 %)
When selected communications mode 2 (± 0.5 %)

- Noise resistance is improved by mounting the IEBus driver/receiver (layer 1) externa
- Module stop mode can be set.

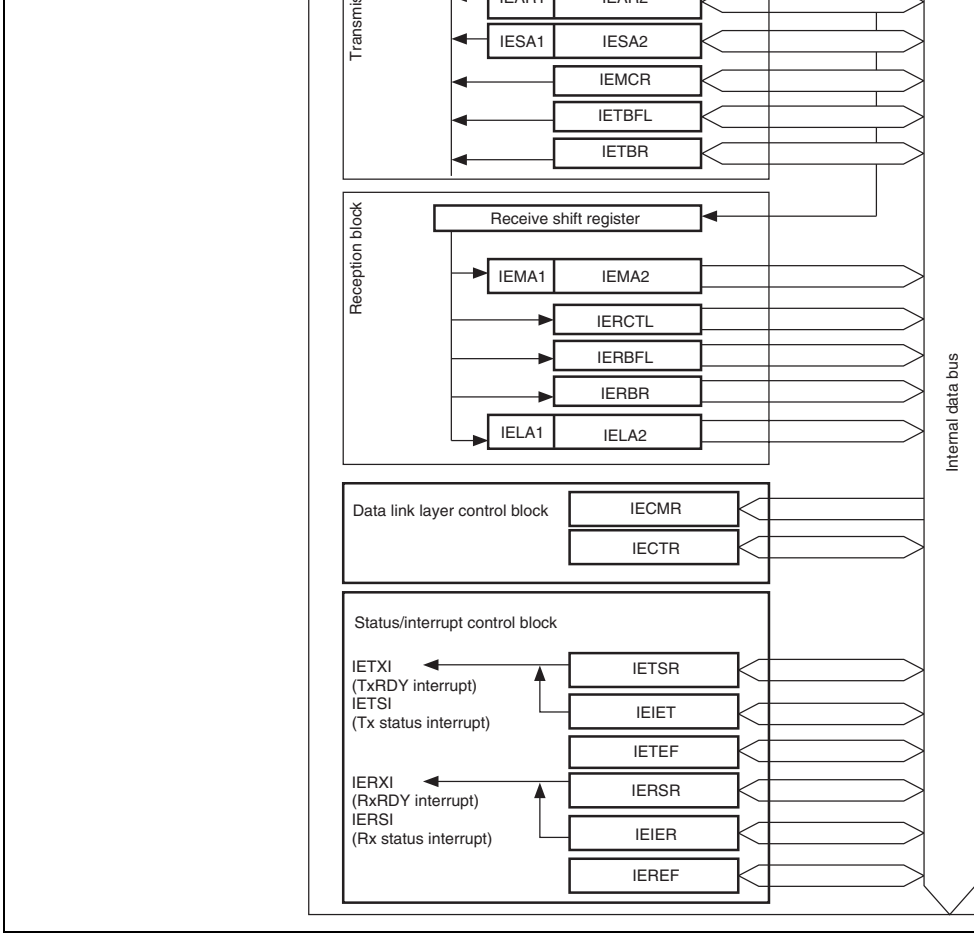


Figure 17.1 Block Diagram of IEB

- Mode is selectable (three modes with different transfer speeds).

Table 17.1 Mode Types

Mode	ϕ = 12, 18, 24 MHz	ϕ = 12.58, 18.87, 25.16 MHz	Maximum Number Transfer Bytes (byt
0	About 3.9 kbps	About 4.1 kbps	16
1	About 17 kbps	About 18 kbps	32
2	About 26 kbps	About 27 kbps	128

- Access control: CSMA/CD (Carrier Sense Multiple Access with Collision Detection
  Priority of bus mastership is as follows.
  - Broadcast communications (one-to-many communications) have priority rather t normal communications (one-to-one communications).
  - Smaller master address has priority.
- Communications scale
  - Number of units: Up to 50
  - Cable length: Up to 150 m (when using a twisted pair cable)

Note: The communications scale of the actual system depends on the externally mount driver/receiver characteristics and the characteristics of the cable to be used.

(1) Determination of Bus Mastership (Arbitration)

A unit connected to the IEBus performs an operation for getting the bus to control other This operation is called arbitration. In arbitration, when the multiple units start transfer simultaneously, the bus mastership is given to one unit among them.

priority, and a unit with HHHH has the lowest priority.

Note: When a unit loses arbitration, the unit can automatically enter retransfer mode
retransfer times can be selected by bits RN2 to RN0 in IEMCR).

(2) Communications Mode

The IEBus has three communications modes with different transfer speeds. Table 17.2 sh
transfer speed in each communications mode and the maximum number of transfer bytes
communications frame.

Table 17.2 Transfer speed and Maximum Number of Transfer Bytes in Each Communications Mode

Communications Mode	Maximum Number of Transfer Bytes (byte/frame)	Effective Transfer Speed*1 (kbps)	
		ϕ = 12, 18, 24 MHz*2	ϕ = 12.58, 18.87, 25.
0	16	About 3.9	About 4.1
1	32	About 17	About 18
2	128	About 26	About 27

Notes: Each unit connected to the IEBus should select a communications mode prior
performing communications. Note that correct communications is not guarante
master and slave units do not adopt the same communications mode.

In the case of communications between a unit with ϕ = 12 MHz and a unit with
12.58 MHz, correct communications is not possible even if the same communi
mode is adopted. This is similar to the case of communications between a unit
24 MHz and a unit with ϕ = 25.16 MHz, or between a unit with ϕ = 18 MHz and
with ϕ = 18.87 MHz. Communications must be performed at the same oscillatio
frequency.

1. An effective transfer speed when the maximum number of transfer bytes is tra
2. Oscillation frequency when this LSI is used

communicates with multiple slave units. Since there are multiple slave units, acknowled
not returned from the slave units during communications.

A broadcast bit decides whether broadcast or normal communications is performed. (Fo
the broadcast bit, see section 17.1.2 (1) (b), Broadcast Bit.

There are two types of broadcast communications.

(a) Group broadcast communications

Broadcast communications is performed to units with the same group number, mean
those units have the same upper four bits of the communications address.

(b) General broadcast communications

Broadcast communications is performed to all units regardless of the group number.

Group broadcast and general broadcast communications are identified by a slave add
details on the slave address, see section 17.1.2 (3), Slave Address Field.)

17.1.2 Communications Protocol

Figure 17.2 shows an IEBus transfer signal format.

Communications data is transferred as a series of signals referred to as a communicatio
The number of data which can be transmitted in a single communications frame and the
speed differ according to communications mode.

P: Parity bit (1 bit)
A: Acknowledge bit (1 bit)
When A = 0: ACK
When A = 1: NAK
N: Number of bytes

Note: The value of acknowledge bit is ignored in broadcast communications.

Figure 17.2 Transfer Signal Format

(1) Header

Header is comprised of a start bit and a broadcast bit.

(a) Start Bit

The start bit is a signal for informing a start of data transfer to other units. A unit, whi attempts to start data transfer, outputs a low-level signal (start bit) for a specified peri then outputs the broadcast bit.

If another unit is already outputting a start bit when a unit attempts to output a start bi waits for completion of output of the start bit from the other unit without outputting th bit, and then outputs the broadcast bit synchronized with the completion timing.

Other units enter the receive state after detecting the start bit.

(b) Broadcast Bit

The broadcast bit is a bit to identify the type of communications: broadcast or normal.

When this bit is cleared to 0, it indicates the broadcast communications. When it is se indicates the normal communications. Broadcast communications includes group broa and general broadcast, which are identified by a value of the slave address. (For detail slave address, see section 17.1.2 (3), Slave Address Field.)

Since there are multiple slave units, which are communications destination units, in th broadcast communications, the acknowledge bit is not returned from each field descri and below.

In the master address field, self-output data and data on the bus are compared for every c
transfer. If the self-output master address and data on the bus are different, the unit that l
arbitration, stops transfer, and enters the receive state.

Since the IEBus is configured with wired AND, a unit having the smallest master addres
units in arbitration (arbitration master) wins in arbitration.

Finally, only a single unit remains in the transfer state as a master unit after outputting 1
master address.

Next, this master unit outputs a parity bit*, defines the master address to other units, and
enters the slave address field output state.

Note: Since even parity is used, when the number of one bits in the master address is o
parity bit is 1.

(3) Slave Address Field

The slave address field is a field to transmit an address (slave address) of a unit (slave un
which a master transmit data. The slave address field is comprised of slave address bits,
bit, and an acknowledge bit.

The slave address has 12 bits and is output MSB first. The parity bit is output after the 1
address is transmitted in order to avoid receiving the slave address accidentally. The ma
then detects the acknowledgement from the slave unit in order to confirm that the slave
on the bus. When the acknowledgement is detected, the master unit enters the control fie
state. However, the master unit enters the control field output state without detecting the
acknowledgement in broadcast communications.

Note: The group number is the upper 4-bit value of the slave address in group broadcas communications.

(4) Control Field

The control field is a field for transmitting the type and direction of the following data fie control field is comprised of control bits, a parity bit, and an acknowledge bit.

The control bits include four bits and are output MSB first.

The parity bit is output following the control bits. When the parity is correct, and the slav can implement the function required from the master unit, the slave unit returns the acknowledgement and enters the message length field output state. However, if the slave cannot implement the requirements from the master unit even though the parity is correct parity is not correct, the slave unit does not return the acknowledgement, and returns to th waiting (monitor) state.

The master unit enters the subsequent message length field output state after confirming t acknowledgement.

When the acknowledgement is not confirmed, the master unit enters the waiting (monitor and communications end. However, in the case of broadcast communications, the master enters the following message length field output state without confirming the acknowledg For details of the contents of the control bit, see table 17.4.

(5) Message Length Field

The message length field is a field for specifying the number of transfer bytes. The messa field is comprised of message length bits, a parity bit, and an acknowledge bit.

H'00	256 bytes

Note: * If a number greater than the maximum number of transfer bytes in one frame specified, communications are performed in multiple frames depending on the communications mode. In this case, the message length bits indicate the num remaining communications data after the first transfer. In this LSI, after the fir the message length bits must be specified to the number of remaining commu data by a program, since these bits are not automatically specified by the har

This field operation differs depending on the value of bit 3 in the control field: master transmission (bit 3 in the control bits is 1) or master reception (bit 3 in the control bits is

(a) Master Transmission

The master unit outputs the message length bits and parity bit. When the parity is cor slave unit returns the acknowledgement and enters the following data field. Note tha unit does not return the acknowledgement in broadcast communications.

In addition, when the parity is not correct, the slave unit decides that the message len is not correctly received, does not return the acknowledgement, and returns to the wa (monitor) state. In this case, the master unit also returns to the waiting state, and communications end.

(b) Master Reception

The slave unit outputs the message length bits and parity bit. When the parity is corr master unit returns the acknowledgement.

When the parity is not correct, the master unit decides that the message length bits ar correctly received, does not return the acknowledgement, and returns to the waiting s this case, the slave unit also returns to the waiting state, and communications end.

the acknowledge bit is ignored. Operations in master transmission and master reception a described below.

(a) Master Transmission

The master unit transmits the data bits and parity bit to the slave unit to write data fro master unit to the slave unit. The slave unit receives the data bits and parity bit, and re acknowledgement if the parity bit is correct and the receive buffer is empty. If the par not correct or the receive buffer is not empty, the slave unit rejects acceptance of corresponding data and does not return the acknowledgement.

When the slave unit does not return the acknowledgement, the master unit retransmits data. This operation is repeated until either the acknowledgement from the slave unit i detected or the maximum number of data transfer bytes is exceeded.

When the parity is correct and the acknowledgement is output from the slave unit, the unit transmits the subsequent data if data remains and the maximum number of transf is not exceeded.

In the case of broadcast communications, the slave unit does not return the acknowled and the master unit transfers data byte by byte.

(b) Master Reception

The master unit outputs synchronous signals corresponding to all data bits to be read f slave unit.

The slave unit outputs the data bits and parity bit on the bus in accordance with the synchronous signals from the master unit.

The parity bit is added to respective data of the master address, slave address, control, m
length, and data bits.

The even parity is used. When the number of one bits in data is odd, the parity bit is 1. W
number of one bits in data is even, the parity bit is 0.

(8) Acknowledge bit

In normal communications (a single unit to a single unit communications), the acknowle
added to the following position in order to confirm that data is correctly accepted.

- At the end of the slave address field
- At the end of the control field
- At the end of the message length field
- At the end of the data field

The acknowledge bit is defined below.

- 0: indicates that the transfer data is acknowledged. (ACK)
- 1: indicates that the transfer data is not acknowledged. (NAK)

Note that the acknowledge bit is ignored in the case of broadcast communications.

(a) Acknowledge bit at the End of the Slave Address Field

The acknowledge bit at the end of the slave address field becomes NAK in the follow
and transfer is stopped.

— When the parity of the master address or slave address bits is incorrect
— When a timing error (an error in bit format) occurs
— When there is no slave unit

— When the control bits are the locked address read (H'4, H'5) although the unit is no
— When a timing error occurs
— When the control bits are undefined

Note: See section 17.1.3 (1), Slave Status Read (Control Bits: H'0, H'6).

(c) Acknowledge Bit at the End of the Message Length Field

The acknowledge bit at the end of the message length field becomes NAK in the follo cases and transfer is stopped.

— When the parity of the message length bits is incorrect
— When a timing error occurs

(d) Acknowledge Bit at the End of the Data Field

The acknowledge bit at the end of the data field becomes NAK in the following cases transfer is stopped.

— When the parity of the data bits is incorrect*
— When a timing error occurs after the previous transfer of the acknowledge bit
— When the receive buffer becomes full and cannot accept further data

Note: In this case, data field is transferred repeatedly until the number of data reaches th maximum number of transfer bytes if the number of data does not exceed the max number of transfer bytes in one frame.

H'3	0	0	1	1	Reads data and locks
H'4	0	1	0	0	Reads locked address (lower 8 bits)
H'5	0	1	0	1	Reads locked address (upper 4 bits)
H'6	0	1	1	0	Reads slave status (SSR) and unlocks
H'7	0	1	1	1	Reads data
H'8	1	0	0	0	Undefined. Setting prohibited.
H'9	1	0	0	1	Undefined. Setting prohibited.
H'A	1	0	1	0	Writes command and locks
H'B	1	0	1	1	Writes data and locks
H'C	1	1	0	0	Undefined. Setting prohibited.
H'D	1	1	0	1	Undefined. Setting prohibited.
H'E	1	1	1	0	Writes command
H'F	1	1	1	1	Writes data

Notes: 1. According to the value of bit 3 (MSB), the transfer directions of the message l in the following message length field and data in the data field vary.
When bit 3 is 1: Data is transferred from the master unit to the slave unit.
When bit 3 is 0: Data is transferred from the slave unit to the master unit.
2. H'3, H'6, H'A, and H'B are control bits to specify lock setting and cancellation.
When the undefined values of H'1, H'2, H'8, H'9, H'C, and H'D are transmitted acknowledge bit is not returned.

When the control bits received from another unit which locked are not included in table slave unit which has been locked by the master unit rejects acceptance of the control bits not return the acknowledge bit.

by reading the slave status (H'0, H'6). The slave status indicates the result of the last communications that the slave unit performs. All slave units can provide slave status info Figure 17.3 shows bit configuration of the slave status.

Bit 2	0	Unit is unlocked
	1	Unit is locked
Bit 1*3	0	Slave receive buffer is empty
	1	Slave receive buffer is not empty
Bit 0*4	0	Slave transmit buffer is empty
	1	Slave transmit buffer is not empty

Notes: 1. Since this LSI can support up to mode 2, bits 6 and 7 are fixed to 10.
2. The value of bit 4 can be selected by the STE bit in the IEBus master unit address register 1 (IE
3. The slave receive buffer is a buffer which is accessed during data write (control bits: H'8, H'A, H'B, H'E, H'F).
In this LSI, the slave receive buffer corresponds to the IEBus receive buffer register (IERBR); and bit 2 is the value of the RxRDY flag in the IEBus receive status register (IERSR).
4. The slave transmit buffer is a buffer which is accessed during data read (control bits: H'3, H'7).
In this LSI, the slave transmit buffer corresponds to the IEBus transmit buffer register (IETBR) when SRQ = 1 in the IEBus general flag register (IEFLG); and bit 1 is a value which reverses the TxRDY flag in the IEBus transmit/runaway status register (IETSR).

Figure 17.3 Bit Configuration of Slave Status (SSR)

(2) Data Command Transfer (Control Bits: Read (H'3, H'7), Write (H'A, H'B, H

In the case of data read (H'3, H'7), data in the data buffer of the slave unit is read in the m unit. In the case of data write (H'B or H'F) or command write (H'A or H'E), data receive slave unit is processed in accordance with the operation specification of the slave unit.

Notes: 1. The user can select data and commands freely in accordance with the system
2. H'3, H'A, or H'B may lock depending on the communications condition and s

(3) Locked Address Read (Control Bits: H′4, H′5)

In the case of the locked address read (H'4 or H'5), the address (12 bits) of the master un issues lock instruction is configured in bytes shown in figure 17.4.

receives data only from the unit which has locked.

Locking and unlocking are described below.

- Locking

  When the acknowledge bit of 0 in the message length field is transmitted/received wit control bits indicating the lock operation, and then the communications frame is comp before completion of data transmission/reception for the number of bytes specified by message length bits, the slave unit is locked by the master unit. In this case, the bit (bi relevant to lock in the byte data indicating the slave status is set to 1.

  Lock is set only when the number of data exceeds the maximum number of transfer b one frame. Lock is not set by other error termination.

- Unlocking

  When the control bits indicate the lock (H'3, H'A, or H'B) or unlock (H'6) operation a byte data for the number of bytes specified by the message length bits are transmitted/ in a single communications frame, the slave unit is unlocked by the master unit. In thi bit (bit 2) relevant to lock in the byte indicating the slave status is cleared to 0.

Note that locking and unlocking are not performed in broadcast communications.

Note: * There are three methods to unlock by a locked unit itself.
- Perform hardware reset
- Enter module stop mode
- Issue unlock command by the IEBus command register (IECMR)

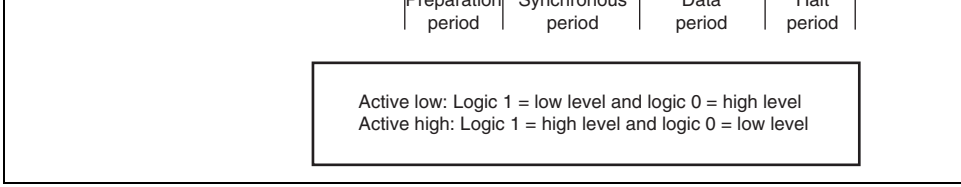


Figure 17.5 IEBus Bit Format (Conceptual Diagram)

Each period of bit format for use of active high signals is described below.

- Preparation period: first logic 1 period (high level)
- Synchronous period: subsequent logic 0 period (low level)
- Data period: period indicating bit value (logic 1: high level, logic 0: low level)
- Halt period: last logic 1 cycle (high level)

For use of active low signals, levels are reversed from the active high signals.

The synchronous and data periods have approximately the same length.

The IEBus is synchronized bit by bit. The specifications for the time of all bits and the p
allocated to the bits differ depending on the type of transfer bits and the unit (master or s

17.3 Register Descriptions

The IEB has the following registers. For the module stop control register, see section 22.1 Module Stop Control Registers A to C (MSTPCRA to MSTPCRC).

- IEBus control register (IECTR)
- IEBUS command register (IECMR)
- IEBus master control register (IEMCR)
- IEBus master unit address register 1 (IEAR1)
- IEBus master unit address register 2 (IEAR2)
- IEBus slave address setting register 1 (IESA1)
- IEBus slave address setting register 2 (IESA2)
- IEBus transmit message length register (IETBFL)
- IEBus transmit buffer register (IETBR)
- IEBus reception master address register 1 (IEMA1)
- IEBus reception master address register 2 (IEMA2)
- IEBus receive control field register (IERCTL)
- IEBus receive message length register (IERBFL)
- IEBus receive buffer register (IERBR)
- IEBus lock address register 1 (IELA1)
- IEBus lock address register 2 (IELA2)
- IEBus general flag register (IEFLG)
- IEBus transmit/runaway status register (IETSR)
- IEBus transmit/runaway interrupt enable register (IEIET)
- IEBus transmit error flag register (IETEF)
- IEBus receive status register (IERSR)

				Switches IEB pin and port functions. 0: The PG3/$\overline{Rx}$/$\overline{CS1}$ and PG2/$\overline{Tx}$/$\overline{CS2}$ pins functi PG3/$\overline{CS1}$ and PG2/$\overline{CS2}$ pins. 1: The PG3/$\overline{Rx}$/$\overline{CS1}$ and PG2/$\overline{Tx}$/$\overline{CS2}$ pins functi $\overline{Tx}$ and $\overline{Rx}$ pins.
6	IOL	0	R/W	Input/Output Level Selects input/output pin level (polarity) for the $\overline{Rx}$ pins. 0: Pin input/output is set to active low. (Logic 1 is and logic 0 is high level.) 1: Pin input/output is set to active high. (Logic 1 i level and logic 0 is low level.)

				time, the master address is stored in IEMA1 and IE The receive data is not stored in IERCTL. While this bit is 0, a reception error interrupt does when the receive buffer is not in the receive enabl and the reception stops and enters the wait state. master address is not saved. 0: A broadcast receive error is not generated up to control field. 1: A broadcast receive error is generated up to the field.
4	CKS1	0	R/W	Input Clock Select Selects clock used by the IEB. See table 17.7.
3	RE	0	R/W	Receive Enable Enables/disables IEB reception. This bit must be s initial setting before frame reception. Changing this before receiving the control field is valid, however, changing this bit after receiving the control field is and the value before the change is validated. 0: Reception is disabled. 1: Reception is enabled.

				transmission. When the DTC is not used, set this generate an underrun error at the last byte transr 0: An underrun error does not occur at the last by transmission (when using the DTC) 1: An underrun error does not occur at the last by transmission (when not using the DTC)
1	CKS0	0	R/W	Input Clock Select Selects clock used by the IEB. See table 17.7.
0	—	0	—	Reserved This bit is always read as 0 and cannot be modifi

Table 17.7 List of System Clock Division Ratio

Bit 4	Bit 1	
CKS1	**CKS0**	**Function**
0	0	1/4 system clock is used (ϕ = 24MHz, 25.16 MHz)
0	1	1/3 system clock is used (ϕ = 18MHz, 18.87 MHz)
1	0	1/2 system clock is used (ϕ = 12MHz, 12.58 MHz)
1	1	Setting prohibited

				These bits cannot be modified.
2	CMD2	0	W	Command Bits
1	CMD1	0	W	These bits issue a command to control IEB communications. When the CMX flag in IEFLG is the command issuance, the command is indicate execution. When the CMX flag becomes 0, the op state is entered. The read value is undefined. Do a bit manipulation instruction that causes malfunc
0	CMD0	0	W	000: No operation. Operation is not affected. 001: Unlock (required from other units)*[1] 010: Requires communications as the master 011: Stops master communications*[2] 100: Undefined bits. Operation is not affected by command. 101: Requires data transfer from the slave. 110: Stops data transfer from the slave*[3]. 111: Undefined bits. Operation is not affected by command.

Notes: 1. Do not execute this command in slave communications. Execute this comman slave communications ends or in master communications. If this command is i slave communications, this command is ignored.

2. This command is valid during master communications (MRQ = 1). In other stat command issuance is ignored. If this command is issued in master communica communications controller immediately enters the wait state. At this time, the is master transmission request ends (MRQ = 0).

Bit	Bit Name	Initial Value	R/W	Description
7	SS	1	R/W	Broadcast/Normal Communications Select Selects broadcast or normal communications for communications. 0: Broadcast communications 1: Normal communications
6	RN2	0	R/W	Retransmission Counts
5	RN1	0	R/W	Set the number of times retransmission is perform
4	RN0	0	R/W	arbitration is lost in master communications. If arb lost for a specified number of times, the AL flag in and the TxE bit in IETSR is set and transmission a transmit error. If arbitration is won during retran the retransmission count is automatically restored initial setting after master address transfer. 000: 0 001: 1 010: 2 011: 3 100: 4 101: 5 110: 6 111: 7

0101: Reads locked address (upper 4 bits)

0110: Reads slave status and unlocks*2

0111: Reads data

1000: Undefined. Setting prohibited.

1001: Undefined. Setting prohibited.

1010: Writes command and locks*2

1011: Writes data and locks*2

1100: Undefined. Setting prohibited.

1101: Undefined. Setting prohibited.

1110: Writes command

1111: Writes data

Notes: 1. CTL3 decides the data transfer direction of the message length bits in the mes
length field and data bits in the data field:

CTL3 = 1: Transfer is performed from master unit to slave unit

CTL3 = 0: Transfer is performed from slave unit to master unit

2. Control bits to lock and unlock

5	IAR1	0	R/W	
4	IAR0	0	R/W	
3	IMD1	0	R/W	IEBus Communications Mode
2	IMD0	0	R/W	Set IEBus communications mode. 00: Communications mode 0 01: Communications mode 1 10: Communications mode 2 11: Setting prohibited
1	—	0	—	Reserved This bit is always read as 0 and cannot be modifi
0	STE	0	R/W	Slave Transmission Setting Sets bit 4 in the slave status register. Transmittin slave status register informs the master unit that transmission enabled state is entered by setting t 1. Note that this bit only sets the slave status reg and does not affect slave transmission directly. 0: Bit 4 in the slave status register is 0 (slave tran stop state) 1: Bit 4 in the slave status register is 1 (slave tran enabled state)

4	IAR8	0	R/W	
3	IAR7	0	R/W	
2	IAR6	0	R/W	
1	IAR5	0	R/W	
0	IAR4	0	R/W	

17.3.6 IEBus Slave Address Setting Register 1 (IESA1)

IESA1 sets the lower 4 bits of the communications destination slave unit address. For sla
communications, it is not necessary to set this register.

Bit	Bit Name	Initial Value	R/W	Description
7	ISA3	0	R/W	Lower 4 Bits of IEBus Slave Address
6	ISA2	0	R/W	These bits set the lower 4 bits of the communications destination slave unit ad
5	ISA1	0	R/W	
4	ISA0	0	R/W	
3 to 0	—	All 0	—	Reserved These bits are always read as 0 and can modified.

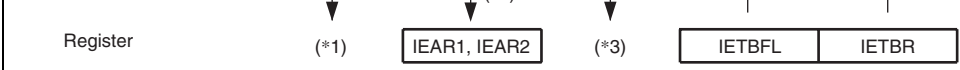
3	ISA7	0	R/W	
2	ISA6	0	R/W	
1	ISA5	0	R/W	
0	ISA4	0	R/W	

17.3.8 IEBus Transmit Message Length Register (IETBFL)

IETBFL sets the message length for master or slave transmission.

Bit	Bit Name	Initial Value	R/W	Description
7	TBFL7	0	R/W	Transmit Message Length
6	TBFL6	0	R/W	Set the message length for master or sla transmission.
5	TBFL5	0	R/W	
4	TBFL4	0	R/W	If a value exceeding the maximum transr for one frame is set in IETBFL, communi are performed with two or more frames i communications modes. In this case, in c second frame, the message length value be the number of bytes of the remaining communications data, however, the initia setting remains unchanged. Therefore, fo second frame or after, re-set the number of the remaining communications data.
3	TBFL3	0	R/W	
2	TBFL2	0	R/W	
1	TBFL1	0	R/W	
0	TBFL0	0	R/W	

7	TBR7	0	R/W	Data to be transmitted is written to this 1-b
6	TBR6	0	R/W	buffer.
5	TBR5	0	R/W	
4	TBR4	0	R/W	
3	TBR3	0	R/W	
2	TBR2	0	R/W	
1	TBR1	0	R/W	
0	TBR0	0	R/W	



Notes: 1. In slave transmission, the received master address is not saved. If the unit is loc address comparison performed.
2. The received slave address is compared with IEAR1 and IEAR2, and if these ac match, operation continues.
3. In slave transmission, the received control bits are not saved. The received con are decoded to decide the subsequent operation.

Figure 17.6 Transmission Signal Format and Registers in Data Transfe

17.3.10 IEBus Reception Master Address Register 1 (IEMA1)

IEMA1 indicates the lower four bits of the communications destination master unit addr slave/broadcast reception. This register is enabled when slave/broadcast reception starts. contents are changed at the timing of setting the RxS flag in IERSR.

If a broadcast receive error interrupt is selected by the DEE bit in IECTR and the receiv not in the receive enabled state on control field reception, a receive error interrupt is gen the lower 4 bits of the master address are stored in IEMA1. This register cannot be modi

Bit	Bit Name	Initial Value	R/W	Description
7	IMA3	0	R	Lower 4 Bits of IEBus Reception Master A
6	IMA2	0	R	Indicates the lower 4 bits of the communi
5	IMA1	0	R	destination master unit address in slave/b reception.
4	IMA0	0	R	
3 to 0	—	All 0	R	Reserved These bits are always read as 0.

Bit	Bit Name	Initial Value	R/W	Description
7	IMA11	0	R	Upper 8 Bits of IEBus Reception Master A
6	IMA10	0	R	Indicates the upper 8 bits of the communi
5	IMA9	0	R	destination master unit address in slave/b
4	IMA8	0	R	reception.
3	IMA7	0	R	
2	IMA6	0	R	
1	IMA5	0	R	
0	IMA4	0	R	

17.3.12 IEBus Receive Control Field Register (IERCTL)

IERCTL indicates the control field value in slave/broadcast reception. This register is ena
when slave/broadcast receive starts, and the contents are changed at the timing of setting
flag in IERSR.

This register cannot be modified.

Bit	Bit Name	Initial Value	R/W	Description
7 to 4	—	All 0	R	Reserved These bits are always read as 0.
3	RCTL3	0	R	IEBus Receive Control Field
2	RCTL2	0	R	Indicates the control field value in slave/b
1	RCTL1	0	R	reception.
0	RCTL0	0	R	

5	RBFL5	0	R	slave/broadcast reception.
4	RBFL4	0	R	
3	RBFL3	0	R	
2	RBFL2	0	R	
1	RBFL1	0	R	
0	RBFL0	0	R	

6	RBR6	0	R	received in master or slave reception
5	RBR5	0	R	
4	RBR4	0	R	
3	RBR3	0	R	
2	RBR2	0	R	
1	RBR1	0	R	
0	RBR0	0	R	

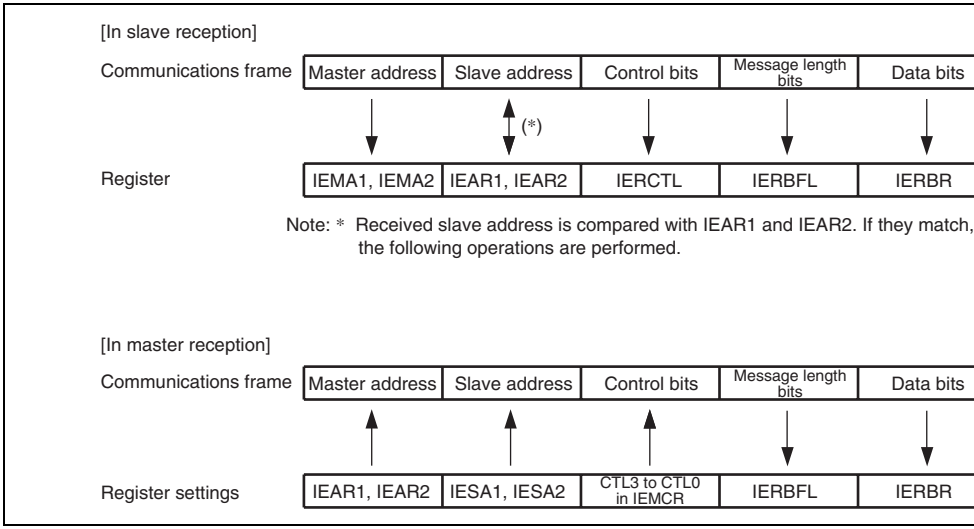


Figure 17.7 Relationship between Transmission Signal Format and Register in IEBus Data Reception

3	ILA3	0	R	
2	ILA2	0	R	
1	ILA1	0	R	
0	ILA0	0	R	

17.3.16 IEBus Lock Address Register 2 (IELA2)

IELA2 is an 8-bit read-only register that specifies the upper 4 bits of a locked address w is locked. Data in this register is valid when the LCK flag in IEFLG is set to 1. This regi cannot be modified.

Bit	Bit Name	Initial Value	R/W	Description
7 to 4	—	All 0	R	Reserved These bits are always read as 0.
3	ILA11	0	R	Upper 4 Bits of IEBus Locked Address
2	ILA10	0	R	Stores the upper 4 bits of the master unit when a unit is locked.
1	ILA9	0	R	
0	ILA8	0	R	

				[Setting condition] • When a master communications request or sla transmit request command is issued while the SRQ, or SRE flag is set to 1 0: A command execution is completed [Clearing condition] • When a command execution has been comple
6	MRQ	0	R	Master Communications Request Indicates whether or not the unit is in communicati request state as a master unit. 1: The unit is in communications request state as a unit [Setting condition] • When the CMX flag is cleared to 0 after the ma communications request command is issued 0: The unit is not in communications request statu master unit [Clearing condition] • When the master communications have been completed

				[Clearing condition] • When a slave transmission has been complet
4	SRE	0	R	Slave Receive Status Indicates the execution status in slave/broadcast 1: Slave/broadcast reception is being executed [Setting condition] • When the slave/broadcast reception is started RE bit in IECTR is set to 1. 0: Slave/broadcast reception is not being execute [Clearing condition] • When the slave/broadcast reception has beer completed.

				that make the unit locked are received from the unit. (The LCK flag is set to 1 only when the m length exceeds the maximum number of transf in one frame. This flag is not set by completion errors.) 0: A unit is unlocked [Clearing condition] • When an unlock condition is satisfied or when unlock command is issued.
2	—	0	R	Reserved This bit is always read as 0.
1	RSS	0	R	Receive Broadcast Bit Status Indicates the received broadcast bit value. This fla when the slave/broadcast reception is started. (Th changed at the timing of setting the RxS flag in IEI The previous value remains unchanged until the n slave/broadcast reception is started.

[Setting condition]

- When H'FFF is acknowledged in the slave fie broadcast reception

[Clearing conditions]

- A unit is in slave reception
- When H'FFF is not acknowledged in slave fie broadcast reception

				DTC* data transfer. When data is transmitted by this flag must be cleared by software. This flag is by writing 0 after reading a 1 from this flag. [Setting conditions] • Immediately after reset • When data can be written to IETBR (when IE loaded data from IETBR to the transmit shift r [Clearing conditions] • When writing 0 after reading TxRDY = 1 • When data is written to TBR by the DTC by a request. Note: This flag is not cleared on the end byte of transfer.
6 to 4	—	All 0	—	Reserved These bits are always read as 0 and cannot be m
3	IRA	0	R/W	IEBus Runaway State Indicates that the on-chip microprogram for IEBus is in the runaway states. This flag is set to 1 when runaway occurs during either IEBus transmission reception. (This flag is not a transfer specific flag also set for a reception runaway.) [Setting condition] • When the on-chip microprogram is in the runa states [Clearing condition] • When writing 0 after reading IRA = 1

				that data transfer is requested [Clearing condition] • When writing 0 after reading TxS = 1
1	TxF	0	R/W	Transmit Normal Completion Indicates that data for the number of bytes specif message length bits has been transmitted with no [Setting condition] • When data for the number of bytes specified message length bits has been transmitted no [Clearing condition] • When writing 0 after reading TxF = 1

	In master reception, an error (arbitration loss, timi or NAK reception) generated after a master communications command is issued before maste reception starts will be detected as a transmit erro [Setting condition] • When the data for the number of bytes specifie message length bits is not completed and whe transmission is terminated [Clearing condition] • When writing 0 after reading TxE = 1

				1: Enables a transmit data ready (TxRDY) interr
6 to 4	—	All 0	—	Reserved These bits are always read as 0 and cannot be
3	IRAE	0	R/W	IEBus Runaway State Interrupt Enable Enables/disables an IEBus runaway state interr 0: Disables an IEBus runaway state interrupt (IF 1: Enables an IEBus runaway state interrupt (IR
2	TxSE	0	R/W	Transmit Start Interrupt Enable Enables/disables a transmit start (TxS) interrupt 0: Disables a transmit start (TxS) interrupt 1: Enables a transmit start (TxS) interrupt
1	TxFE	0	R/W	Transmit Normal Completion Interrupt Enable Enables/disables a transmit normal completion interrupt. 0: Disables a transmit normal completion (TxF) 1: Enables a transmit normal completion (TxF) i
0	TxEE	0	R/W	Transmit Error Termination Interrupt Enable Enables/disables a transmit error termination (T interrupt. 0: Disables a transmit error termination (TxE) in 1: Enables a transmit error termination (TxE) int

4	AL	0	R/W	Arbitration Loss The IEB retransmits from the start bit for the num times specified by bits RN2 to Rn0 in IEMCR if th arbitration has been lost in master communicatio arbitration has been lost for the specified number the AL and TxE flags are set to enter the wait sta arbitration has been won within retransmit for the specified number of times, this flag is not set to 1 flag is set only when the arbitration has been lost wait state is entered. [Setting condition] • When the arbitration has been lost during dat transmission and the transmission has been terminated [Clearing condition] • When writing 0 after reading AL = 1

				[Setting condition] • When the IEB loads data from IETBR to the shift register while the TxRDY flag is set to 1 [Clearing condition] • When writing 0 after reading UE = 1
2	TTME	0	R/W	Timing Error Set to 1 if data is not transmitted at the timing sp the IEBus protocol during data transmission. Th the TxE flag and enters the wait state. [Setting condition] • When a timing error occurs during data trans [Clearing condition] • When writing 0 after reading TTME = 1

[Setting condition]

- When the transmit has not been completed altho maximum number of bytes defined by communic mode have been transmitted

[Clearing condition]

- When writing 0 after reading RO = 1

0	ACK	0	R/W	Acknowledge bit Status Indicates the data received in the acknowledge bit c data field. • Acknowledge bit other than in the data field The IEB terminates the transmission and enters state if a NAK is received. In this case, this bit a TxE flag are set to 1. • Acknowledge bit in the data field The IEB retransmits data up to the maximum nu bytes defined by communications mode until an received from the receive unit if a NAK is receive the receive unit during data field transmission. In case, when an ACK is received from the receive during retransmission, this flag is not set and transmission will be continued. When transmissi terminated without receiving an ACK, this flag is Note: This flag is invalid in broadcast communicati [Setting condition] • When the acknowledge bit of 1 (NAK) is detecte [Clearing condition] • When writing 0 after reading ACK = 1

				cleared by DTC* data transfer. When data is tra by the CPU, this flag must be cleared by softwa [Setting condition] • When data reception has been completed n and receive data has been loaded to IERBR [Clearing conditions] • When writing 0 after reading RxRDY = 1 • When IERBR data is read by the DTC by a request. Note: This flag cannot be cleared on the end b DTC transfer.
6 to 3	—	All 0	—	Reserved These bits are always read as 0 and cannot be
2	RxS	0	R/W	Receive Start Detection Indicates that the IEB starts reception. [Setting conditions] • Master reception: When the message length been received from the slave unit correctly a arbitration is won and the control field transm completed • Slave reception: When the message length been received from the master unit correctly [Clearing condition] • When writing 0 after reading RxS = 1

0	RxE	0	R/W	Receive Error Completion Indicates that data for the number of bytes specifie message length bits is not completed and that the reception is terminated. The source of this error ca checked by the contents of IEREF. This flag is set timing that an error indicated by IEREF occurs. Th flag can be cleared even when the error source fla IEREF is set to 1 because the RxE flag is not logic ORed with the flags in IEREF. [Setting condition] • When the data for the number of bytes specifie message length bits is not completed and whe reception is terminated. [Clearing condition] • When writing 0 after reading RxE = 1

				1: Enables a receive data ready (RxRDY) interru
6 to 3	—	All 0	—	Reserved These bits are always read as 0 and cannot be m
2	RxSE	0	R/W	Receive Start Interrupt Enable Enables/disables a receive start (RxS) interrupt. 0: Disables a receive start (RxS) interrupt 1: Enables a receive start (RxS) interrupt
1	RxFE	0	R/W	Receive Normal Completion Enable Enables or disables a receive normal completion interrupt. 0: Disables a receive normal completion (RxF) in 1: Enables a receive normal completion (RxF) int
0	RxEE	0	R/W	Receive Error Termination Interrupt Enable Enables or disables a receive error termination (R interrupt. 0: Disables a receive error termination (RxE) inte 1: Enables a receive error termination (RxE) inter

7 to 4	—	All 0	—	Reserved These bits are always read as 0 and cannot be m
3	OVE	0	R/W	Overrun Control Flag Used to control the overrun during data reception sets the OVE and RxE flags when the IEB receive next byte data while the receive data has not bee (the RxRDY flag is not cleared) and when the par reception has been started. If this flag remains se acknowledge bit transfer, the IEB assumes that a overrun error has occurred and returns a NAK to t communications destination unit. The communications destination unit retransmits d to the maximum number of transmit bytes. The IE however, returns a NAK when this flag remains se because the IEB assumes that the overrun error h been cleared. If this flag is cleared to 0, the IEB decides that the error has been cleared, returns an ACK, and rece next data. In broadcast reception, if this flag is set during acknowledge bit transmission, the IEB immediate the wait state. [Setting condition] • When the next byte data is received while the flag is not cleared and when the parity bit of th received. [Clearing condition] • When writing 0 after reading OVE = 1

1	DLE	0	R/W	Overflow of Maximum Number of Receive Bytes Frame Indicates that the maximum number of bytes def communications mode have been received beca parity error or overrun error occurred, or that the has not be completed because the message len exceeds the maximum number of receive bytes frame. The IEB sets the RxE flag and enters the state. [Setting condition] • When the reception has not been completed the maximum number of bytes defined by communications mode have been received. [Clearing condition] • When writing 0 after reading DLE = 1

destination unit continues retransfer up to the max
number of receive bytes in one frame and if the re
has been completed normally by clearing the parit
the PE flag is not set. If the parity error is not clear
the reception is terminated before receiving data fo
number of bytes specified by the message length,
flag is set.

In broadcast reception, if a parity error occurs duri
field reception, the IEB enters the wait state imme
after setting the PE flag.

[Setting condition]

- When the parity bit of last data of the data field
  correct after the maximum number of receive b
  been received

[Clearing condition]

- When writing 0 after reading PE = 1

the LULL bit to 0 since the transfer is performed by the DTC.

(b) Setting the IEBus Master Unit Address Registers 1 and 2 (IEAR1 and IEAR2)
Specify the master unit address and specify the communications mode in IEAR1.

(c) Setting the IEBus Slave Address Setting Registers 1 and 2 (IESA1 and IESA2)
Specify the communications destination slave unit address.

(d) Setting the IEBus Master Control register (IEMCR)
Select broadcast/normal communications, specify the number of retransfer counts at loss, and specify the control bits.

(e) Setting the IEBus Transmit Message Length Register (IETBFL)
Specify the message length bits.

(f) Setting the IEBus Transmit/Runaway Interrupt Enable Register (IEIET)
Enable TxRDY (IETxI), TxS, TxF, and TxE (IETSI) interrupts.

The above registers can be specified in any order. (The register specification order does the IEB operation.)

(2) DTC Initialization

1. Set the start address of the RAM which stores the register information necessary for transfer in the vector address (H'000004D4) to be accessed when a DTC transfer req generated.
2. Set the following data from the start address of the RAM.
   — Transfer source address (SAR): Start address of the RAM which stores data to be transmitted in the data field.
   — Transfer destination address (DAR): Address (H'FFF808) of the IEBus transmit register (IETBR)
   — Transfer count (CRA): The same value as the IETBFL contents

1. After the IEB and DTC have been initialized, a master communications request comm issued from IECMR. During slave reception, the command execution status flag (CM IEFLG is set and the master communications request will not be issued.
2. When the slave reception has been completed, the CMX flag is cleared, the master communications command is executed, and the MRQ flag is set.
3. The transmit start detection flag (TxS) in IETSR is set when arbitration is won and th address has been transmitted. In this case, one of the transmit status interrupts (IETSI requested to the CPU, and the TxS flag is cleared in the interrupt handling routine.
4. The IEB loads data to be transmitted in the data field from IETBR when the control a message length fields have been transmitted and an ACK is received in each field. Af the TxRDY flag is set. A DTC transfer request is generated by IETxI and the second b written to the transmit buffer.
5. Similarly, the data field load and transmission are repeated.
6. The DTC completes the data transfer for the number of specified bytes when data to b transmitted in the last byte is written to. At this time, the DTC does not clear the TxR It, however, clears bit DTCEG5 in the DTC enable register G (DTCERG) so as not to more DTC transfer request.
7. A TxRDY interrupt (IETxI) is issued to the CPU when the DTC transfer is completed interrupt handling routine, the TxRDY flag can be cleared. However, since a TxRDY will be generated again after the last byte transfer, the TxRDY flag remains set. (Note LUEE bit must be cleared to 0 because an underrun error occurs to terminate the trans LUEE bit in IECTR is set to 1.) Note, however, that the TxRDY interrupt must be dis because the TxRDY interrupt is always generated.
8. A transmit normal completion (TxF) interrupt (IETSI) occurs after the last data transf completed. In this case, the CPU clears the TxF flag and completes the normal compl interrupt and clears the MRQ flag to 0.

Note: As a transmit status interrupt (IETSI), the transmit error termination (TxE) interru well as the transfer start detection (TxS) and transmit normal completion (TxF) in

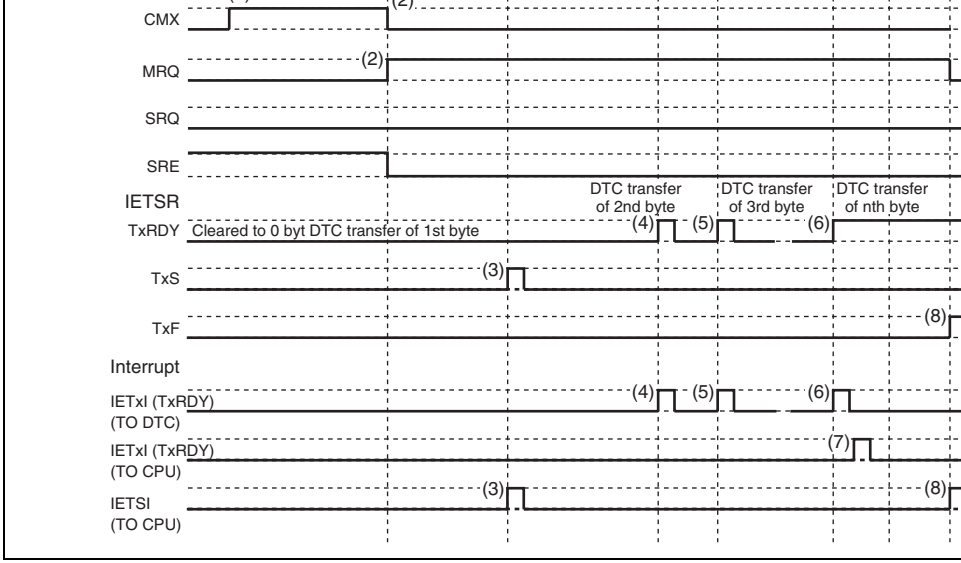


Figure 17.8 Master Transmit Operation Timing

17.4.2 Slave Receive Operation

This section describes an example of performing a slave reception using the DTC.

(1) IEB Initialization

(a) Setting the IEBus Control Register (IECTR)

Enable the IEBus pins, select the signal polarity, and select a clock supplied to the IE
RE bit to 1 to perform reception. The LUEE bit does not need to be specified.

(b) Setting the IEBus Master Unit Address Registers 1 and 2 (IEAR1 and IEAR2)

transfer in the vector address (H'000004D2) to be accessed when a DTC transfer requ
generated.

2. Specify the following from the start address of the RAM.
   — Transfer source address (SAR): Address (H'FFF80D) of the IEBus receive buffer
   (IERBR).
   — Transfer destination address (DAR): Start address of the RAM which stores data r
   from the data field.
   — Transfer count (CRA): Maximum number of transfer bytes in one frame in the tra
   mode.
3. Set DTCEG6 in the DTC enabler register G (DTCERG) to enable the RxRDY interru
   (IETxI).

Because the above settings are performed before the frame reception, the length of data to
received cannot be decided. Accordingly, the maximum number of transfer bytes in one f
specified as the DTC transfer count.

If the DTC is specified after reception starts, the above settings are performed in the recei
(RxS) interrupt handling routine. In this case, the transfer count must be the same value a
contents of the IEBus receive message length register (IERBFL).

(3) Slave Reception Flow

Figure 17.9 shows the slave reception flow. Numbers in the following description corresp
the number in Figure 17.9. In this example, the DTC is specified when the frame receptio

1. After the broadcast reception has been completed, the slave reception is performed. T
   receive broadcast bit status flag (RSS) in IEFLG retains the previous frame informatio
   1) until the receive start detection flag (RxS) is set to 1. If the RSS flag changes at the
   of header reception, the interrupt handling of the broadcast reception completion mus

bytes after loading the receive data to the RAM. In this case, the DTC does not clear RxRDY flag. It, however, clears the DTC enable register G (DTCEG). Accordingly, no transfer request will be issued to the DTC.

6. When the DTC transfer has been completed, an RxRDY interrupt (IERxI) is issued t CPU. In this interrupt handling routine, the RxRDY flag is cleared.
7. When the last data is received, a receive normal completion (RxF) interrupt (IERSI) this case, the CPU clears the RxF flag in order to complete the normal completion in The SRE flag is cleared to 0.

Notes: 1. As a receive status interrupt (IERSI), the receive error termination (RxE) inte well as the receive start detection (RxS) and receive normal completion (RxF interrupts must be enabled. If an error termination interrupt is disabled, no in generated even if the reception is terminated by an error.

2. The interrupt occurs after the DTC transfer has been completed. Accordingly interrupt described in item 6 actually occurs after item 7 above.

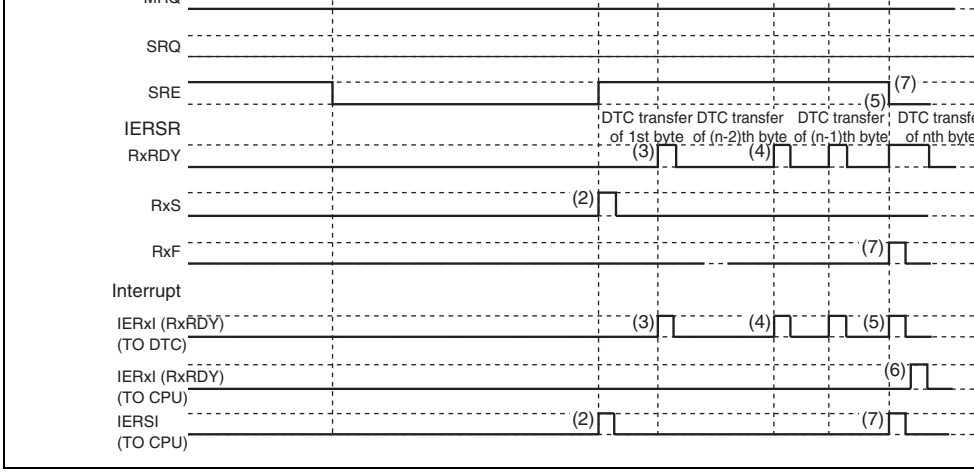


Figure 17.9 Slave Reception Operation Timing

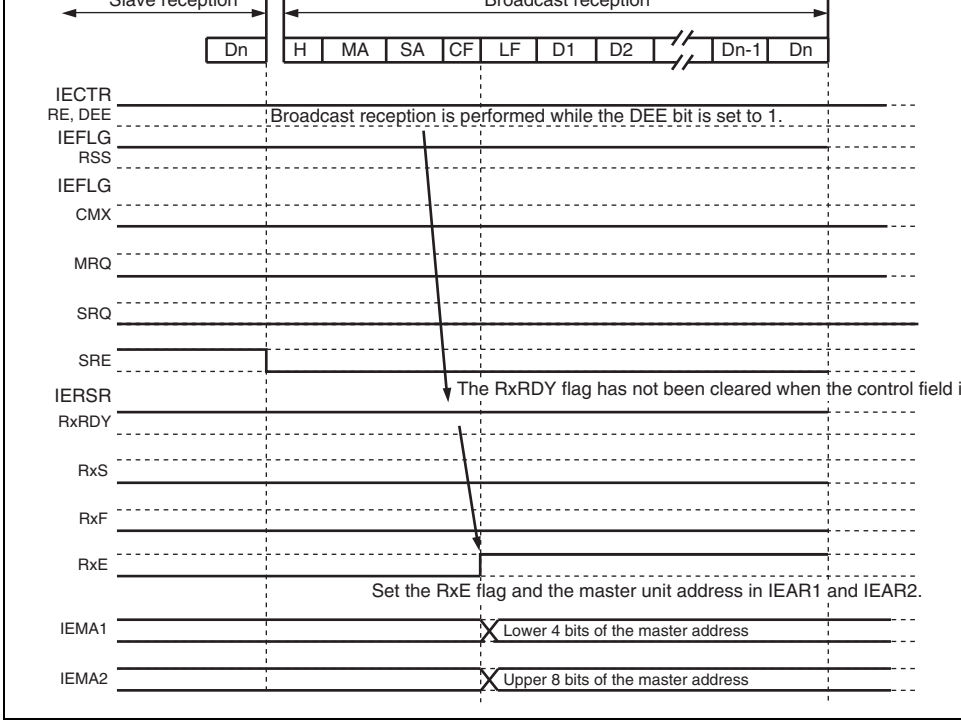


Figure 17.10 Error Occurrence in the Broadcast Reception (DEE = 1)

Specify the master unit address and specify the communications mode in IEAR1. Con
with the slave address in the communications frame and receive the frame if matched.

(c) Setting the IEBus Slave Address Setting Registers 1 and 2 (IESA1 and IESA2)

Specify the communications destination slave unit address.

(d) Setting the IEBus Master Control Register (IEMCR)

Select broadcast/normal communications, specify the number of retransfer counts at a
loss, and specify the control bits.

(e) Setting the IEBus Receive Interrupt Enable Register (IEIER)

Enable the RxRDY (IERxI), RxS, RxF, and RxE (IERSI) interrupts.

The above registers can be specified in any order. (The register specification order does n
the IEB operation.)

(2) DTC Initialization

1. Set the start address of the RAM which stores the register information necessary for th
transfer in the vector address (H'000004D2) to be accessed when a DTC transfer requ
generated.
2. Set the following data from the start address of the RAM.
   — Transfer source address (SAR): Address (H'FFF80D) of the IEBus receive buffer
   (IERBR).
   — Transfer destination address (DAR): Start address of the RAM which stores data t
   received from the data field.
   — Transfer count (CRA): Maximum number of transfer bytes in one frame in the tra
   mode.
3. Set bit DTCEG6 in the DTC enabler register G (DTCERG), and enable the RxRDY in
(IERxI).

the number in figure 17.11. In this example, the DTC is specified when the frame recept

1. After the IEB has been initialized, a master communications request command is iss IECMR. During slave reception, the command execution status flag (CMX) in IEFL and the master communications request will not be issued.
2. The CMX flag is cleared when the slave reception is completed, the master commun command is executed, and the MRQ flag is set.
3. If the arbitration is won, the master address, slave address, and control field will be transmitted. An error generated before the control field transmission will be handled transmission error. In this case, the TxE flag is set and the error contents will be refl IETEF.
4. The message length field is received from the slave unit. If no parity error is detecte reception is performed correctly, the receive start detection flag (RxS) is set to 1. If a error occurs, it is handled as a receive error. A receive start detection (RxS) interrupt status interrupt (IERSI)) occurs and the DTC initialization described in (2) is perfor DTC initialization, the RxS flag is cleared to 0.
5. When the first data is received, the RxRDY flag is set to 1. A DTC transfer request b occurs and the DTC loads data from the IEBus receive buffer register (IERBR) and c RxRDY flag.
6. Similarly, the above data field receive and load operations are repeated.
7. When the last data is received, the DTC completes the data transfer for the specified bytes after loading the receive data to the RAM. In this case, the DTC does not clear RxRDY flag. It, however, clears the DTC enable register G (DTCEG). Accordingly, no transfer request will be issued to the DTC.
8. When the DTC transfer has been completed, an RxRDY interrupt (IERxI) is issued t CPU. In this interrupt handling routine, the RxRDY flag is cleared.

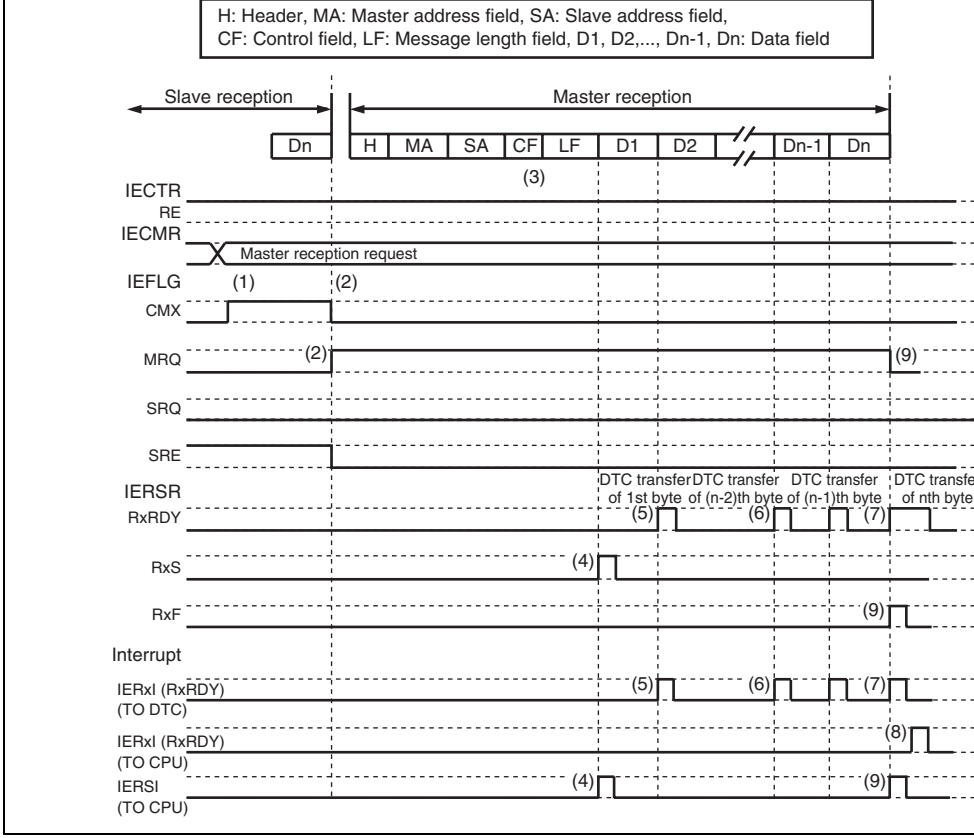


Figure 17.11 Master Receive Operation Timing

Specify the master unit address and specify the communications mode in IEAR1. Co
with the slave address in the communications frame and receive the frame if matched

(c) Setting the IEBus Transmit Message Length Register (IETBFL)

Specify the message length bits.

(d) Setting the IEBus Transmit/Runaway Interrupt Enable Register (IEIET)

Enable the TxRDY (IETxI), TxS, and TxE (IETSI) interrupts.

The above registers can be specified in any order. (The register specification order does
the IEB operation.)

(2) DTC Initialization

1. Set the start address of the RAM which stores the register information necessary for
transfer in the vector address (H'000004D4) to be accessed a DTC transfer request is
generated.
2. Set the following data from the start address of the RAM.
   - Transfer source address (SAR): Start address of the RAM which stores data to be
transmitted from the data field.
   - Transfer destination address (DAR): Address (H'FFF808) of the IEBus transmit b
register (IETBR)
   - Transfer count (CRA): The same value as IETBFL
3. Set bit DTCEG5 in the DTC enabler register G (DTCERG), and enable the TxRDY
(IETxI).

   Because the TxRDY flag is retained after reset, the DTC transfer is executed when th
enabled and the first data field data is written to IETBR. The DTC negates the TxRD
and the DTC transfer of the first byte is completed.

is H'3 or H'7, the transmit start detection flag (TxS) in IETSR register is set to 1. In th the TxS flag is cleared in the TxS interrupt handling routine.

4. The slave then transmits the message length field, and the IEB loads the transmit data data field from IETBR when the ACK is received. Then the TxRDY flag is set to 1. A transfer request by IETxI is generated and the second byte data is written to the transr buffer.
5. Similarly, the above data field load and transmission operations are repeated.
6. The DTC completes the data transfer for the number of specified bytes when data to b transmitted in the last byte is written to. At this time, the DTC does not clear the TxR It, however, clears bit DTCEG5 in the DTC enable register G (DTCERG) not to gene DTC transfer request.
7. A TxRDY interrupt (IETxI) is issued to the CPU when the DTC transfer is completed interrupt handling routine, the TxRDY flag can be cleared. However, since the TxRD interrupt will be generated again after the last byte transfer, the TxRDY flag remains s that the LUEE bit should be cleared to 0 because an underrun error occurs to terminat transfer if the LUEE bit in IECTR is set to 1.) Note, however, that the TxRDY interru be disabled because the TxRDY interrupt is always generated.
8. After the last data transfer has been completed, a transmit normal completion (TxF) in occurs. In this case, the CPU clears the TxF flag and completes the normal completion interrupt and clears the SRQ flag to 0.

Notes: 1. As a transmit status interrupt (IETSI), a transmit error termination (TxE) inter well as the transmit start detection (TxS) and transmit normal completion (Tx interrupts must be enabled. If a transmit error completion interrupt is disabled interrupt is generated even if the transfer is terminated by an error.

2. If the control bits sent from the master unit is H'0, H'4, H'5, or H'6 in slave transmission, the IEB automatically performs processing and the TxS and TxF not set.

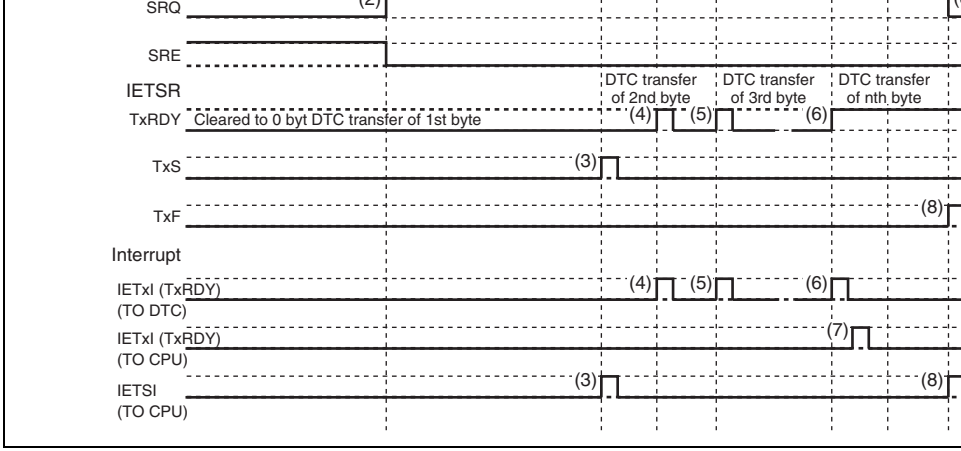
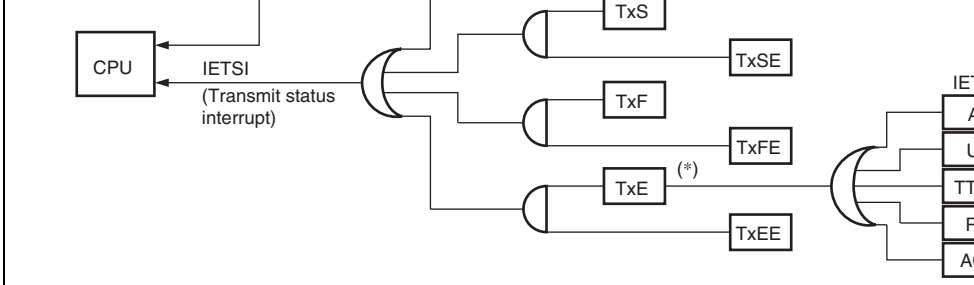
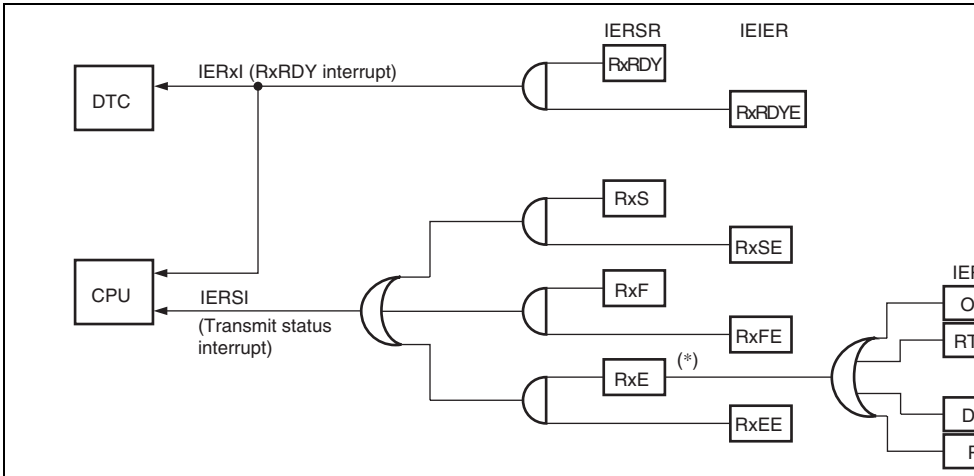


Figure 17.12 Slave Transmit Operation Timing



Note: * The TxE flag is set at the timing when an error source of IETEF occurs. The TxE flag can be cleared e
when the error source flag in IETEF is set to 1 because the TxE flag is not logically ORed with flags in

Figure 17.13 Relationships among Transfer Interrupt Sources



Note: * The RxE flag is set at the timing when an error source of IEREF occurs. The RxE flag can be cleared e
when the error source flag in IEREF is set to 1 because the RxE flag is not logically ORed with flags in

Figure 17.14 Relationships among Receive Interrupt Sources

1. The TxRDY flag indicates that IETBR is empty. Writing to IETBR by the DTC clea TxRDY flag. Meanwhile, the TxRDY flag must be cleared by software since writing by the CPU does not clear the TxRDY flag.
2. If the CPU fails to write to IETBR by the timing of the frame transmission or if the n transfer words is less than the length specified by the message length bits, an underru occurs.
3. The IEB decides that an underrun error occurred when the data is loaded from IETB transmit shift register while the TxRDY flag is set to 1. In this case, the IEB sets the in IETSR and enters the wait state. The UE flag in IETEF is also set to 1.
4. On the receive side, the unit decides that a timing error has occurred because the communications are terminated.
5. In data transfer using the DTC, the TxRDY flag in IETSR is not cleared after the las is transferred to IETBR and a CPU interrupt caused by the DTC interrupt will occur. If the TxRDY flag is not cleared in this CPU interrupt handling routine, an underrun occur when the last byte data is loaded from IETBR to the transmit shift register. In if the LUEE bit is cleared to 0 (initial value), no underrun error occurs and the last b data field is transmitted correctly. (if the LUEE bit is set to 1, an underrun error occu
6. Although the DTC is used as described in item 5, if the number of DTC transfer wor than the length specified by the message length bits, the LUEE bit setting is invalid. LUEE bit is valid only when data is transmitted for the number of bytes specified by message length bits has been transmitted.) In this case, an underrun error occurs, dat transmitted for one byte less than the DTC transfer words, and the transfer is termina transmit error.

shift register.

4. On the transmit side, the unit continues retransfer until an ACK is received because it a NAK.
5. If the OVE flag is cleared without loading the receive data from IERBR in the RxE in handling routine caused when the OVE flag is set to 1, the IEB decides that the overru has been cleared and sends an ACK to other units. In this case, the transmit unit comp communications correctly. However, no receive data is loaded from the IERBR and t unit continues reception. Accordingly, in an interrupt handling routine caused by the flag, receive data must be loaded from IERBR, the RxRDY flag must be cleared. The thus, should be ready to receive the next byte, and then the OVE flag must be cleared.
6. Item 5 above will not occur when the DTC transfer words is specified as the IERBFL

17.6.4 Error Flag s in the IETEF

(1) AL Flag

The AL Flag is set to 1 when arbitration is lost even if retransfer is performed for the nun times specified by IEMCR after arbitration has been lost. The AL flag is not set when arb is won during retransfer. If the AL flag is set to 1, the TxE flag is set and the wait state is

(2) UE Flag

If the UE flag is set to 1, the TxE flag is set and the wait state is entered. For details, see s 17.6.2, TxRDY Flag and Underrun Error.

(3) TTME Flag

If a timing error occurs during data transfer, the TTME and TxE flags are set, and the wai entered.

(5) ACK Flag

- If a NAK is received in an acknowledge bit before the message length field transmis ACK flag is set, the TxE flag is set, and then the wait state is entered.
- If a NAK is received in an acknowledge bit of the data field, data is automatically retransmitted up to the maximum number of transfer bytes defined by the protocol. I is received in an acknowledge bit during retransfer and the following data is transmi correctly, the ACK flag is not set. If a NAK is received in the last data transfer durin retransfer for the maximum number of transfer bytes, the ACK flag is set to 1 and th state is entered.

Note: Even if a NAK is received from the receive side during the data field transmissi retransfer is performed up to the maximum number of transfer bytes defined by protocol, and the number of transferred bytes is less than that of bytes specified message length bits, an ACK may be received in the acknowledge bit in the last transfer. In this case, the ACK flag is not set although the RO flag is set.

17.6.5 Error Flags in IEREF

(1) OVE Flag

When the OVE flag is set, the RxE flag is also set. If an overrun error is cleared and the is also cleared, the IEBus receive operation is continued. For details, see section 17.6.3, Flag and Overrun Error.

(2) RTME Flag

If a timing error occurs during data reception after reception starts (the RxS flag is set to RTME flag is set to 1, RxE flag is set to 1, and the wait state is entered. When a timing e occurs before reception starts, this flag is not set and the reception frame is discarded.

(4) PE Flag

If a parity error occurs after reception starts (the RxS flag is set to 1), a NAK is sent to pe
reception.

If a parity error is not cleared when the maximum number of transfer bytes specified by t
protocol is received, the PE flag is set to 1, the RxE flag is set to 1 and the wait state is en
a parity error is cleared during the rereception and if the following data is received correc
PE flag is not set.

Notes: 1. If the reception is performed up to the maximum number of transfer bytes defi
the protocol because of a parity or an overrun error during data field reception
number of receive bytes is less than that of bytes specified by the message len
no parity error or overrun error may occur at the last byte reception. In this cas
DLE flag is set. However, the OVE and PE flags are not set.

2. The flags in IEREF are set after reception starts. Accordingly, the RxE flag is
set after the RxS flag has been set. If an error occurs before reception starts, th
is discarded and no interrupt occurs.

17.6.6 Notes on Slave Transmission

When the slave unit transmits the slave status and upper and lower locked addresses, a pa
overrun error occurs in the master reception side and the data cannot be received. Accord
even if a NAK is returned, the slave unit is not capable of retransfer.

In this case, the master unit must discard the frame in which an error occurred and reques
above operation in the master reception to receive the correct frame.

already transferred to the transmit buffer by the DTC and the TxRDY flag that is the DT initiation source is already cleared to 0 (2).

In this case, if retransfer is performed, data remained in the transmit buffer (previous fra is transmitted as the first byte data of the data field (3).

To avoid this error, in master transmission, the first byte data in the data field should be the transmit buffer by software instead of using the DTC. After that, data can be transfe DTC. In this case, the SAR (transfer source address) and CRA (transfer counter) should specified as follows.

- An address of the on-chip memory that stores the second byte data → SAR
- The number of bytes specified by message length –1 → CRA

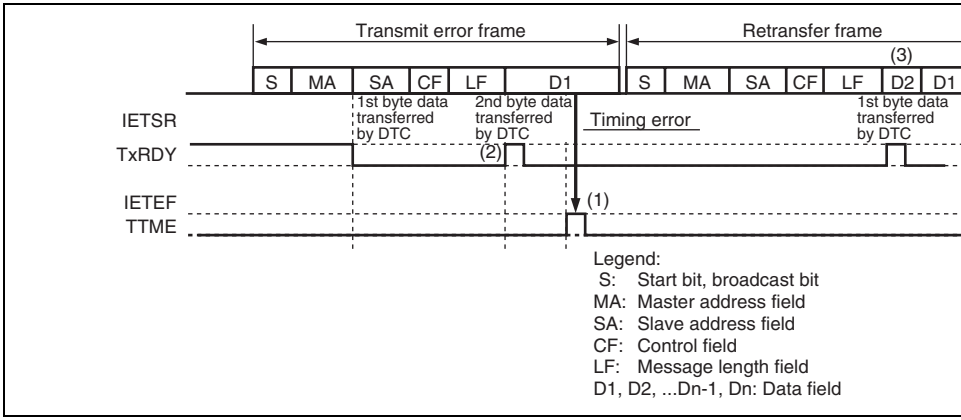


Figure 17.15 Error Processing in Transfer

17.6.11 Notes on Register Access

The IEB registers can be accessed in bytes. The IEB registers must not be accessed in wo
longwords.

- CAN version: Conforming to Bosch 2.0B active
  Communication systems: NRZ (Non-Return to Zero) system (with bit-stuffing funct
  Broadcast communication system
  Transmission path: Bidirectional 2-wire serial communication
  Communication speed: Max. 1 Mbps
  Data length: 0 to 8 bytes
- Number of channels: 1
- Data buffers: 16 (one receive-only buffer and 15 buffers settable for transmission/re
- Data transmission: Two methods
  Mailbox (buffer) number order (low-to-high)
  Message priority (identifier) reverse-order (high-to-low)
- Data reception: Two methods
  Message identifier match (transmit/receive-setting buffers)
  Reception with message identifier masked (receive-only)
- CPU interrupts: 12
  Error interrupt
  Reset processing interrupt
  Message reception interrupt
  Message transmission interrupt
- HCAN operating modes
- Support for various modes
  Hardware reset
  Software reset
  Normal status (error-active, error-passive)

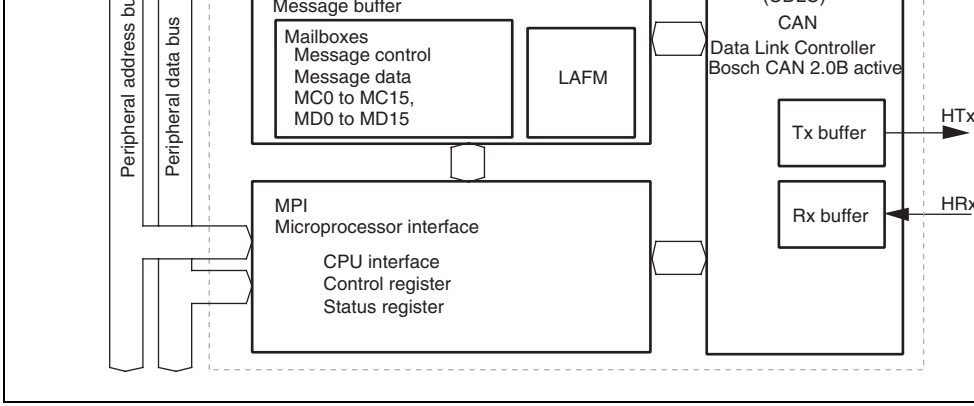


Figure 18.1 HCAN Block Diagram

- Message Buffer Interface (MBI)

  The MBI, consisting of mailboxes and a local acceptance filter mask (LAFM), stores transmit/receive messages (identifiers, data, etc.) Transmit messages are written by th For receive messages, the data received by the CDLC is stored automatically.

- Microprocessor Interface (MPI)

  The MPI, consisting of a bus interface, control register, status register, etc., controls H internal data, status, and so forth.

- CAN Data Link Controller (CDLC)

  The CDLC transmits and receives of messages conforming to the Bosch CAN Ver. 2. standard (data frames, remote frames, error frames, overload frames, inter-frame spac well as CRC checking, bus arbitration, and other functions.

HCAN receive data pin	HRxD	Input	CAN bus receptio

A bus driver is necessary for the interface between the pins and the CAN bus. A Philips PCA82C250 compatible model is recommended.

18.3 Register Descriptions

The HCAN has the following registers.

- Master control register (MCR)
- General status register (GSR)
- Bit configuration register (BCR)
- Mailbox configuration register (MBCR)
- Transmit wait register (TXPR)
- Transmit wait cancel register (TXCR)
- Transmit acknowledge register (TXACK)
- Abort acknowledge register (ABACK)
- Receive complete register (RXPR)
- Remote request register (RFPR)
- Interrupt register (IRR)
- Mailbox interrupt mask register (MBIMR)
- Interrupt mask register (IMR)
- Receive error counter (REC)
- Transmit error counter (TEC)
- Unread message status register (UMSR)
- Local acceptance filter mask H (LAFMH)

	MCR4	0	R/W	HCAN Sleep Mode Release When this bit is set to 1, the HCAN automatically e HCAN sleep mode on detection of CAN bus opera
6	—	0	R	Reserved This bit is always read as 0. The write value shoul be 0.
5	MCR5	0	R/W	HCAN Sleep Mode When this bit is set to 1, the HCAN transits to HCA mode. When this bit is cleared to 0, HCAN sleep m released.
4, 3	—	All 0	R	Reserved These bits are always read as 0. The write value s always be 0.
2	MCR2	0	R/W	Message Transmission Method 0: Transmission order determined by message ide priority 1: Transmission order determined by mailbox num priority (TXPR1 > TXPR15)
1	MCR1	0	R/W	Halt Request When this bit is set to 1, the HCAN transits to HCA mode. When this bit is cleared to 0, HCAN HALT r released.

				• Module stop mode • 1-write (software reset) [Clearing condition] • When 0 is written to this bit while the GSR3 b is 1

18.3.2 General Status Register (GSR)

GSR indicates the status of the HCAN.

Bit	Bit Name	Initial Value	R/W	Description
7 to 4	—	All 0	R	Reserved These bits are always read as 0. The write value always be 0.
3	GSR3	1	R	Reset Status Bit Indicates whether the HCAN module is in the nor operating state or the reset state. This bit cannot modified. [Setting conditions] • When entering configuration mode after the H internal reset has finished • Sleep mode [Clearing condition] • When entering normal operation mode after t bit in MCR is cleared to 0 (Note that there is a between clearing of the MCR0 bit and the GS

				This bit cannot be modified. [Setting condition] When TEC ≥ 96 or REC ≥ 96 [Clearing conditions] • When TEC < 96 and REC < 96 • When TEC ≥ 256 (bus off state)
0	GSR0	0	R	Bus Off Flag This bit cannot be modified. [Setting condition] • When TEC ≥ 256 (bus off state) [Clearing condition] • Recovery from bus off state

				01: 2 time quanta 10: 3 time quanta 11: 4 time quanta
13	BCR5	0	R/W	Baud Rate Prescaler (BRP)
12	BCR4	0	R/W	Set the length of time quantum.
11	BCR3	0	R/W	000000: 2 × system clock
10	BCR2	0	R/W	000001: 4 × system clock
9	BCR1	0	R/W	000010: 6 × system clock
8	BCR0	0	R/W	: 111111: 128 × system clock
7	BCR15	0	R/W	Bit Sample Point (BSP) Sets the point at which data is sampled. 0: Bit sampling at one point (end of time segment (TSEG1)) 1: Bit sampling at three points (end of TSEG1 and preceding and following one time quantum)

				110: 7 time quanta
				111: 8 time quanta
3	BCR11	0	R/W	Time Segment 1 (TSEG1)
2	BCR10	0	R/W	Set the TSEG1 (PRSEG + PHSEG1) width to betw
1	BCR9	0	R/W	and 16 time quanta.
0	BCR8	0	R/W	0000: Setting prohibited
				0001: Setting prohibited
				0010: Setting prohibited
				0011: 4 time quanta
				0100: 5 time quanta
				0101: 6 time quanta
				0110: 7 time quanta
				0111: 8 time quanta
				1000: 9 time quanta
				1001: 10 time quanta
				1010: 11 time quanta
				1011: 12 time quanta
				1100: 13 time quanta
				1101: 14 time quanta
				1110: 15 time quanta
				1111: 16 time quanta

10	MBCR2	0	R/W	Bit 8 is reserved. This bit is always read a
9	MBCR1	0	R/W	the write value should always be 1.
8	—	1	R	
7	MBCR15	0	R/W	
6	MBCR14	0	R/W	
5	MBCR13	0	R/W	
4	MBCR12	0	R/W	
3	MBCR11	0	R/W	
2	MBCR10	0	R/W	
1	MBCR9	0	R/W	
0	MBCR8	0	R/W	

11	TXPR3	0	R/W	• Completion of message transmission • Completion of transmission cancellation
10	TXPR2	0	R/W	
9	TXPR1	0	R/W	
8	—	0	R	Bit 8 is reserved. This bit is always read as 0. The write value should always be 0.
7	TXPR15	0	R/W	
6	TXPR14	0	R/W	
5	TXPR13	0	R/W	
4	TXPR12	0	R/W	
3	TXPR11	0	R/W	
2	TXPR10	0	R/W	
1	TXPR9	0	R/W	
0	TXPR8	0	R/W	

10	TXCR2	0	R/W	message is canceled normally
9	TXCR1	0	R/W	Bit 8 is reserved. This bit is always read a
8	—	0	R	the write value should always be 0.
7	TXCR15	0	R/W	
6	TXCR14	0	R/W	
5	TXCR13	0	R/W	
4	TXCR12	0	R/W	
3	TXCR11	0	R/W	
2	TXCR10	0	R/W	
1	TXCR9	0	R/W	
0	TXCR8	0	R/W	

11	TXACK3	0	R/(W)*	[Setting condition]
10	TXACK2	0	R/(W)*	• Completion of message transmission fo corresponding mailbox
9	TXACK1	0	R/(W)*	
8	—	0	R	[Clearing condition]
7	TXACK15	0	R/(W)*	• Writing 1
6	TXACK14	0	R/(W)*	Bit 8 is reserved. This bit is always read as the write value should always be 0.
5	TXACK13	0	R/(W)*	
4	TXACK12	0	R/(W)*	
3	TXACK11	0	R/(W)*	
2	TXACK10	0	R/(W)*	
1	TXACK9	0	R/(W)*	
0	TXACK8	0	R/(W)*	

Note: * Only 1 can be written to this bit for clearing the flag.

11	ABACK3	0	R/(W)*	[Setting condition]
10	ABACK2	0	R/(W)*	• Completion of transmit message canc corresponding mailbox
9	ABACK1	0	R/(W)*	
8	—	0	R	[Clearing condition]
7	ABACK15	0	R/(W)*	• Writing 1
6	ABACK14	0	R/(W)*	Bit 8 is reserved. This bit is always read a write value should always be 0.
5	ABACK13	0	R/(W)*	
4	ABACK12	0	R/(W)*	
3	ABACK11	0	R/(W)*	
2	ABACK10	0	R/(W)*	
1	ABACK9	0	R/(W)*	
0	ABACK8	0	R/(W)*	

Note: * Only 1 can be written to this bit for clearing the flag.

12	RXPR4	0	R/(W)*	• Completion of message (data frame or frame) reception in corresponding mailb [Clearing condition] • Writing 1
11	RXPR3	0	R/(W)*	
10	RXPR2	0	R/(W)*	
9	RXPR1	0	R/(W)*	
8	RXPR0	0	R/(W)*	
7	RXPR15	0	R/(W)*	
6	RXPR14	0	R/(W)*	
5	RXPR13	0	R/(W)*	
4	RXPR12	0	R/(W)*	
3	RXPR11	0	R/(W)*	
2	RXPR10	0	R/(W)*	
1	RXPR9	0	R/(W)*	
0	RXPR8	0	R/(W)*	

Note: * Only 1 can be written to this bit for clearing the flag.

12	RFPR4	0	R/(W)*	• Completion of remote frame reception in corresponding mailbox [Clearing condition] • Writing 1
11	RFPR3	0	R/(W)*	
10	RFPR2	0	R/(W)*	
9	RFPR1	0	R/(W)*	
8	RFPR0	0	R/(W)*	
7	RFPR15	0	R/(W)*	
6	RFPR14	0	R/(W)*	
5	RFPR13	0	R/(W)*	
4	RFPR12	0	R/(W)*	
3	RFPR11	0	R/(W)*	
2	RFPR10	0	R/(W)*	
1	RFPR9	0	R/(W)*	
0	RFPR8	0	R/(W)*	

Note: * Only 1 can be written to this bit for clearing the flag.

				[Clearing condition] • Writing 1
14	IRR6	0	R/(W)*	Bus Off Interrupt Flag Status flag indicating the bus off state caused by t transmit error counter. [Setting condition] • When TEC ≥ 256 [Clearing condition] • Writing 1
13	IRR5	0	R/(W)*	Error Passive Interrupt Flag Status flag indicating the error passive state cause transmit/receive error counter. [Setting condition] • When TEC ≥ 128 or REC ≥ 128 [Clearing condition] • Writing 1
12	IRR4	0	R/(W)*	Receive Overload Warning Interrupt Flag Status flag indicating the error warning state cause receive error counter. [Setting condition] • When REC ≥ 96 [Clearing condition] • Writing 1

				Status flag indicating that a remote frame has be received in a mailbox when MBIMR = 0. [Setting condition] • When remote frame reception is completed, v corresponding MBIMR = 0 [Clearing condition] • Clearing of all bits in RFPR (remote request r
9	IRR1	0	R	Receive Message Interrupt Flag Status flag indicating that a mailbox receive mess been received normally when MBIMR = 0. [Setting condition] • When data frame or remote frame reception i completed, when corresponding MBIMR = 0 [Clearing condition] • Clearing of all bits in RXPR (receive complete

				• When the reset operation has finished after en power-on reset or software standby mode, wat or module stop mode. [Clearing condition] • Writing 1
7 to 5	—	All 0	—	Reserved These bits are always read as 0. The write value s always be 0.
4	IRR12	0	R/(W)*	Bus Operation Interrupt Flag Status flag indicating detection of a dominant bit d operation when the HCAN module is in HCAN slee [Setting condition] • Bus operation (dominant bit) detection in HCAI mode [Clearing condition] • Writing 1
3, 2	—	All 0	—	Reserved These bits are always read as 0. The write value s always be 0.

0	IRR8	0	R/(W)*	Mailbox Empty Interrupt Flag Status flag indicating that the next transmit mess be stored in the mailbox. [Setting condition] • When TXPR (transmit wait register) is cleared completion of transmission or completion of transmission abort [Clearing condition] • Writing 1

Note: * Only 1 can be written to this bit for clearing the flag.

10	MBIMR2	1	R/W	transmission cancellation. The interrupt sou
9	MBIMR1	1	R/W	receive mailbox is RXPR setting on recepti
8	MBIMR0	1	R/W	
7	MBIMR15	1	R/W	
6	MBIMR14	1	R/W	
5	MBIMR13	1	R/W	
4	MBIMR12	1	R/W	
3	MBIMR11	1	R/W	
2	MBIMR10	1	R/W	
1	MBIMR9	1	R/W	
0	MBIMR8	1	R/W	

18.3.13 Interrupt Mask Register (IMR)

IMR enables or disables requests by individual interrupt sources of IRR. The interrupt fla
be masked.

Bit	Bit Name	Initial Value	R/W	Description
15	IMR7	1	R/W	Overload Frame Recovery Interrupt Mask When this bit is cleared to 0, OVR0 (interrupt requ IRR7) is enabled. When set to 1, OVR0 is masked
14	IMR6	1	R/W	Bus Off Interrupt Mask When this bit is cleared to 0, ERS0 (interrupt requ IRR6) is enabled. When set to 1, ERS0 is masked

				IRR3) is enabled. When set to 1, OVR0 is maske
10	IMR2	1	R/W	Remote Frame Request Interrupt Mask When this bit is cleared to 0, OVR0 (interrupt req IRR2) is enabled. When set to 1, OVR0 is maske
9	IMR1	1	R/W	Receive Message Interrupt Mask When this bit is cleared to 0, RM1 (interrupt requ IRR1) is enabled. When set to 1, RMI is masked.
8	—	0	R	Reserved This bit is always read as 0. The write value shou be 0.
7 to 5	—	All 1	R	Reserved These bits are always read as 1. The write value always be 1.
4	IMR12	1	R/W	Bus Operation Interrupt Mask When this bit is cleared to 0, OVR0 (interrupt req IRR12) is enabled. When set to 1, OVR0 is mask
3, 2	—	All 1	R	Reserved These bits are always read as 1. The write value always be 1.
1	IMR9	1	R/W	Unread Interrupt Mask When this bit is cleared to 0, OVR0 (interrupt req IRR9) is enabled. When set to 1, OVR0 is maske
0	IMR8	1	R/W	Mailbox Empty Interrupt Mask When this bit is cleared to 0, SLE0 (interrupt requ IRR8) is enabled. When set to 1, SLE0 is masked

18.3.16 Unread Message Status Register (UMSR)

UMSR is a status register that indicates, for individual mailboxes, that a received messag
been overwritten by a new receive message before being read. When overwritten by a ne
message, data in the unread receive message is lost.

Bit	Bit Name	Initial Value	R/W	Description
15	UMSR7	0	R/(W)*	Indicates that a received massage has been overw
14	UMSR6	0	R/(W)*	a new message before being read.
13	UMSR5	0	R/(W)*	[Setting condition]
12	UMSR4	0	R/(W)*	When a new message is received before RXPR is
11	UMSR3	0	R/(W)*	[Clearing condition]
10	UMSR2	0	R/(W)*	Writing 1
9	UMSR1	0	R/(W)*	
8	UMSR0	0	R/(W)*	
7	UMSR15	0	R/(W)*	
6	UMSR14	0	R/(W)*	
5	UMSR13	0	R/(W)*	
4	UMSR12	0	R/(W)*	
3	UMSR11	0	R/(W)*	
2	UMSR10	0	R/(W)*	
1	UMSR9	0	R/(W)*	
0	UMSR8	0	R/(W)*	

Note: * Only 1 can be written to this bit for clearing the flag.

14	LAFML6	0	R/W	When this bit is set to 1, ID-6 of the receive identifier is not compared.
13	LAFML5	0	R/W	When this bit is set to 1, ID-5 of the receive identifier is not compared.
12	LAFML4	0	R/W	When this bit is set to 1, ID-4 of the receive identifier is not compared.
11	LAFML3	0	R/W	When this bit is set to 1, ID-3 of the receive identifier is not compared.
10	LAFML2	0	R/W	When this bit is set to 1, ID-2 of the receive identifier is not compared.
9	LAFML1	0	R/W	When this bit is set to 1, ID-1 of the receive identifier is not compared.
8	LAFML0	0	R/W	When this bit is set to 1, ID-0 of the receive identifier is not compared.
7	LAFML15	0	R/W	When this bit is set to 1, ID-15 of the receiv message identifier is not compared.
6	LAFML14	0	R/W	When this bit is set to 1, ID-14 of the receiv message identifier is not compared.
5	LAFML13	0	R/W	When this bit is set to 1, ID-13 of the receiv message identifier is not compared.
4	LAFML12	0	R/W	When this bit is set to 1, ID-12 of the receiv message identifier is not compared.

- LAFMH

Bit	Bit Name	Initial Value	R/W	Description
15	LAFMH7	0	R/W	When this bit is set to 1, ID-20 of the receiv message identifier is not compared.
14	LAFMH6	0	R/W	When this bit is set to 1, ID-19 of the receiv message identifier is not compared.
13	LAFMH5	0	R/W	When this bit is set to 1, ID-18 of the receiv message identifier is not compared.
12 to 10	—	All 0	R	Reserved These bits are always read as 0. The write should always be 0.
9	LAFMH1	0	R/W	When this bit is set to 1, ID-17 of the receiv message identifier is not compared.
8	LAFMH0	0	R/W	When this bit is set to 1, ID-16 of the receiv message identifier is not compared.
7	LAFMH15	0	R/W	When this bit is set to 1, ID-28 of the receiv message identifier is not compared.
6	LAFMH14	0	R/W	When this bit is set to 1, ID-27 of the receiv message identifier is not compared.
5	LAFMH13	0	R/W	When this bit is set to 1, ID-26 of the receiv message identifier is not compared.
4	LAFMH12	0	R/W	When this bit is set to 1, ID-25 of the receiv message identifier is not compared.
3	LAFMH11	0	R/W	When this bit is set to 1, ID-24 of the receiv message identifier is not compared.

Mail box 3	MC3[1]	MC3[2]	MC3[3]	MC3[4]	MC3[5]	MC3[6]	MC3[7]	MC3[8
Mail box 15	MC15[1]	MC15[2]	MC15[3]	MC15[4]	MC15[5]	MC15[6]	MC15[7]	MC15[8

Figure 18.2 Message Control Register Configuration

The setting of message control registers are shown in the following. Figures 18.3 and 18.4 the correspondence between the identifiers and register bit names.

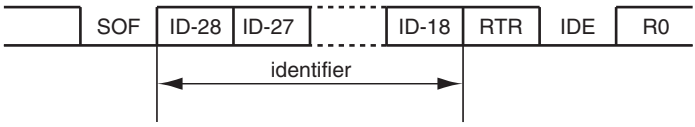


Figure 18.3 Standard Format

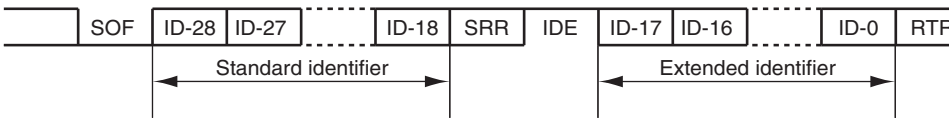


Figure 18.4 Extended Format

				0100: 4 bytes 0101: 5 bytes 0110: 6 bytes 0111: 7 bytes 1000: 8 bytes : : 1111: 8 bytes
MCx[2]	7 to 0	—	R/W	The initial value of these bits is undefined must be initialized (by writing 0 or 1).
MCx[3]	7 to 0	—	R/W	
MCx[4]	7 to 0	—	R/W	
MCx[5]	7 to 5	ID-20 to ID-18	R/W	Sets ID-20 to ID-18 in the identifier.
	4	RTR	R/W	Remote Transmission Request Used to distinguish between data frames remote frames. 0: Data frame 1: Remote frame
	3	IDE	R/W	Identifier Extension Used to distinguish between the standard and extended format of data frames and frames. 0: Standard format 1: Extended format
	2	—	R/W	The initial value of this bit is undefined. It initialized by writing 0 or 1.
	1, 0	ID-17 to ID-16	R/W	Sets ID-17 and ID-16 in the identifier.

sets of these registers. Because message data registers are in RAM, their initial values aft
on are undefined. Be sure to initialize them by writing 0 or 1. Figure 18.5 shows the regis
names for each mailbox.

Mail box 0	MD0[1]	MD0[2]	MD0[3]	MD0[4]	MD0[5]	MD0[6]	MD0[7]	MD0[8
Mail box 1	MD1[1]	MD1[2]	MD1[3]	MD1[4]	MD1[5]	MD1[6]	MD1[7]	MD1[8
Mail box 2	MD2[1]	MD2[2]	MD2[3]	MD2[4]	MD2[5]	MD2[6]	MD2[7]	MD2[8
Mail box 3	MD3[1]	MD3[2]	MD3[3]	MD3[4]	MD3[5]	MD3[6]	MD3[7]	MD3[8
Mail box 15	MD15[1]	MD15[2]	MD15[3]	MD15[4]	MD15[5]	MD15[6]	MD15[7]	MD15[8

Figure 18.5 Message Data Configuration

registers, except for message control and message data registers, are initialized by a
reset.

- Software Reset

  The HCAN can be reset by setting the MCR reset request bit (MCR0) in MCR via so
  a software reset, the error counters (TEC and REC) are initialized, however other reg
  not. If the MCR0 bit is set while the CAN controller is performing a communication
  (transmission or reception), the initialization state is not entered until message transf
  been completed. The reset status bit (GSR3) in GSR is set on completion of initializa

18.4.2 Initialization after Hardware Reset

After a hardware reset, the following initialization processing should be carried out:

1. Clearing of IRR0 bit in the interrupt register (IRR)
2. Bit rate setting
3. Mailbox transmit/receive settings
4. Mailbox (RAM) initialization
5. Message transmission method setting

These initial settings must be made while the HCAN is in bit configuration mode. Confi
mode is a state in which the GSR3 bit in GSR is set to 1 by a reset. Configuration mode
by clearing the MCR0 bit in MCR to 0; when the MCR0 bit is cleared to 0, the HCAN
automatically clears the GSR3 bit in GSR. There is a delay between clearing the MCR0
clearing the GSR3 bit because the HCAN needs time to be internally reset, there is a del
between clearing of the MCR0 bit and GSR3 bit. After the HCAN exits configuration m
power-up sequence begins, and communication with the CAN bus is possible as soon as
consecutive recessive bits have been detected.

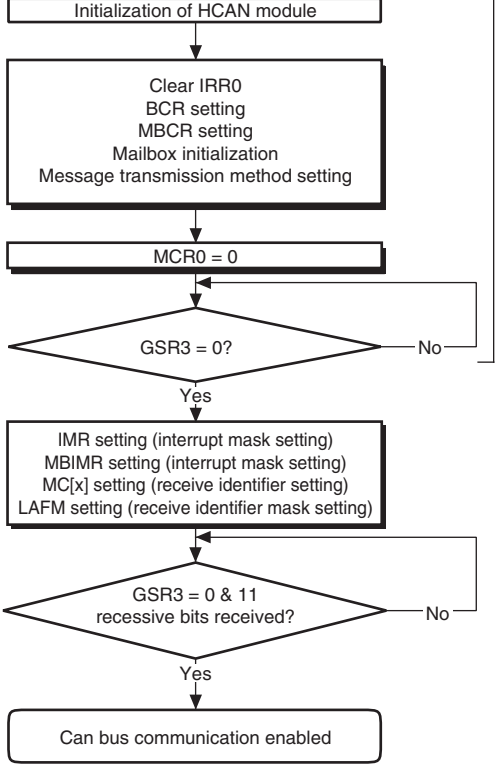


Figure 18.6 Hardware Reset Flowchart

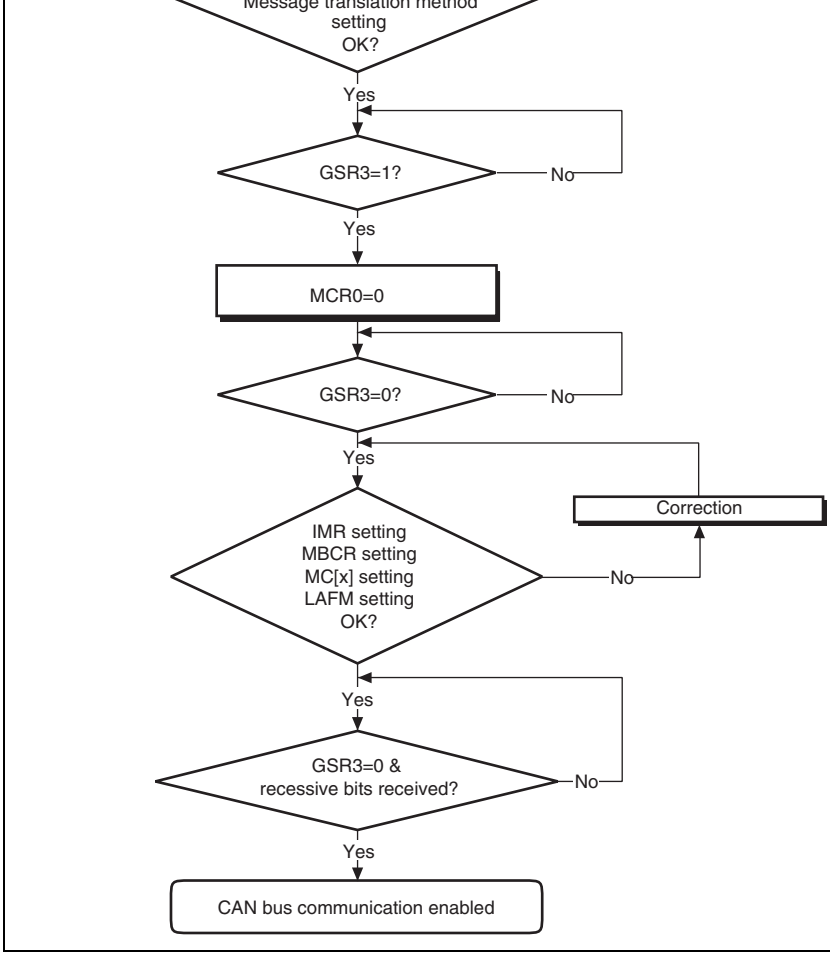


Figure 18.7 Software Reset Flowchart

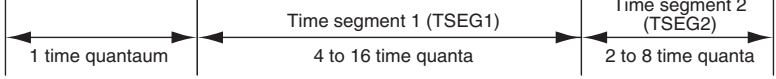


Figure 18.8 Detailed Description of One Bit

SYNC_SEG is a segment for establishing the synchronization of nodes on the CAN bus. bit edge transitions occur in this segment. PRSEG is a segment for compensating for the delay between networks. PHSEG1 is a buffer segment for correcting phase drift (positive segment is extended when synchronization (resynchronization) is established. PHSEG2 is segment for correcting phase drift (negative). This segment is shortened when synchroniz (resynchronization) is established. Limits on the settable value (TSEG1, TSEG2, BRP, B SJW) are shown in table 18.2.

Table 18.2 Limits for Settable Value

Name	Abbreviation	Min. Value	Max. Va
Time segment 1	TSEG1	B'0011*[3]	B'1111
Time segment 2	TSEG2	B'001*[2]	B'111
Baud rate prescaler	BRP	B'000000	B'11111
Bit sample point	BSP	B'0	B'1
Re-synchronization jump width	SJW*[1]	B'00	B'11

Notes: 1. SJW is stipulated in the CAN specifications:
$3 \geq SJW \geq 0$
2. The minimum value of TSEG2 is stipulated in the CAN specifications:
$TSEG2 \geq SJW$
3. The minimum value of TSEG1 is stipulated in the CAN specifications:
$TSEG1 > TSEG2$

Note: I_{CLK} = φ (system clock)
BCR value is used for BRP, TSEG1, and TSEG2.

Example: With a system clock of 20 MHz, a BRP setting of B'000000, a TSEG1 setting
B'0100, and a TSEG2 setting of B'011:

Bit rate = 20/{2 × (0 + 1) × (3 + 4 + 3)} = 1 Mbps

	1000	Yes*	Yes	Yes	Yes	Yes	Yes
	1001	Yes*	Yes	Yes	Yes	Yes	Yes
	1010	Yes*	Yes	Yes	Yes	Yes	Yes
	1011	Yes*	Yes	Yes	Yes	Yes	Yes
	1100	Yes*	Yes	Yes	Yes	Yes	Yes
	1101	Yes*	Yes	Yes	Yes	Yes	Yes
	1110	Yes*	Yes	Yes	Yes	Yes	Yes
	1111	Yes*	Yes	Yes	Yes	Yes	Yes

Note: The time quantum value for TSEG1 and TSEG2 is the TSEG value + 1.

* When baud rate prescaler (BRP), BCR[13:8], is not B'000000 (2 × system cloc can be set.

Mailbox Transmit/Receive Settings: The HCAN has 16 mailboxes. Mailbox 0 is receiv while mailboxes 1 to 15 can be set for transmission or reception. The Initial status of mai to 15 is for transmission. Mailbox transmit/receive settings are not initialized by a softwa

Clearing a bit to 0 in the mailbox configuration register (MBCR) designates the correspo mailbox for transmission use, whereas a setting of 1 in MBCR designates the correspondi mailbox for reception use. When setting mailboxes for reception, in order to improve mes reception efficiency, high-priority messages should be set in low-to-high mailbox order.

Mailbox (Message Control/Data) Initial Settings: Message control/data are held in RA so their initial values are undefined after power is supplied. Initial values must therefore b all the mailboxes (by writing 0s or 1s).

Setting the Message Transmission Method: The following two kinds of message transm methods are available.

When messages are set to be transmitted according to the mailbox number priority, if se
messages are designated as waiting for transmission (TXPR = 1), messages are stored in
transmit buffer in low-to-high mailbox order. CAN bus arbitration is then carried out for
message stored in the transmit buffer, and the message is transmitted when the transmiss
is acquired.

18.4.3 Message Transmission

Messages are transmitted using mailboxes 1 to 15. The transmission procedure after init
is described below, and a transmission flowchart is shown in figure 18.9.

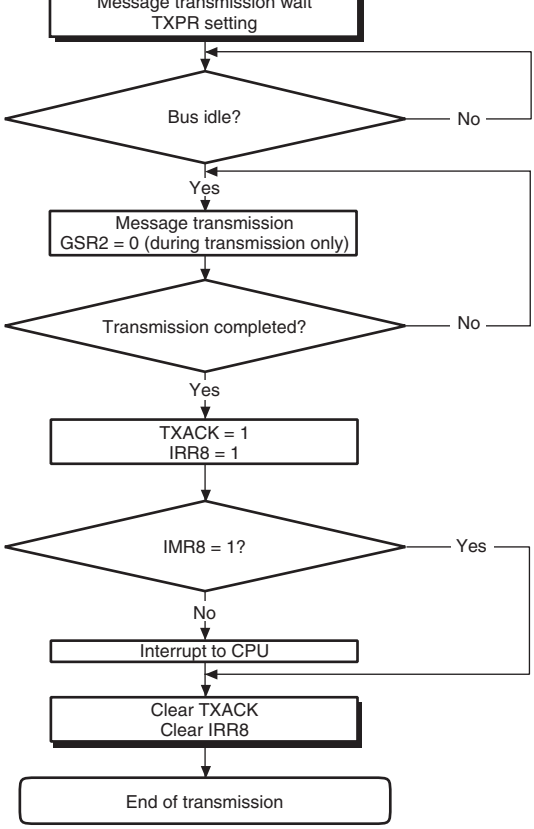


Figure 18.9 Transmission Flowchart

CPU Interrupt Source Settings: The CPU interrupt source is set by the interrupt mask r
(IMR) and mailbox interrupt mask register (MBIMR). Transmission acknowledge and
transmission abort acknowledge interrupts can be generated for individual mailboxes in t

Data Field Setting: In the data field, the data to be transmitted is set within the range ze
eight. The registers to be set are the message data registers MDx[1] to MDx[8]. The byt
the data to be transmitted is determined by the data length code in the control field. Ever
exceeding the value set in the control field is set in the data field, up to the byte length s
control field will actually be transmitted.

Message Transmission: If the corresponding mailbox transmit wait bit (TXPR1 to TXF
the transmit wait register (TXPR) is set to 1 after message control and message data regi
been set, the message enters transmit wait state. If the message is transmitted error-free,
corresponding acknowledge bit (TXACK1 to TXACK15) in the transmit acknowledge r
(TXACK) is set to 1, and the corresponding transmit wait bit (TXPR1 to TXPR15) in th
wait register (TXPR) is automatically cleared to 0. Also, if the corresponding bit (MBIM
MBIMR15) in the mailbox interrupt mask register (MBIMR) and the mailbox empty int
(IRR8) in the interrupt mask register (IMR) are both simultaneously set to enable interru
interrupts may be sent to the CPU.

If transmission of a transmit message is aborted in the following cases, the message is
retransmitted automatically:

- CAN bus arbitration failure (failure to acquire the bus)
- Error during transmission (bit error, stuff error, CRC error, frame error, or ACK erro

Message Transmission Cancellation: Transmission cancellation can be specified for a
stored in a mailbox as a transmit wait message. A transmit wait message is canceled by
bit for the corresponding mailbox (TXCR1 to TXCR15) to 1 in the transmit cancel regis
(TXCR). Clearing the transmit wait register (TXPR) does not cancel transmission. Wher
cancellation is executed, the transmit wait register (TXPR) is automatically reset, and th
corresponding bit is set to 1 in the abort acknowledge register (ABACK). An interrupt to
can be requested, and if the mailbox empty interrupt (IRR8) is enabled for the bits (MBI

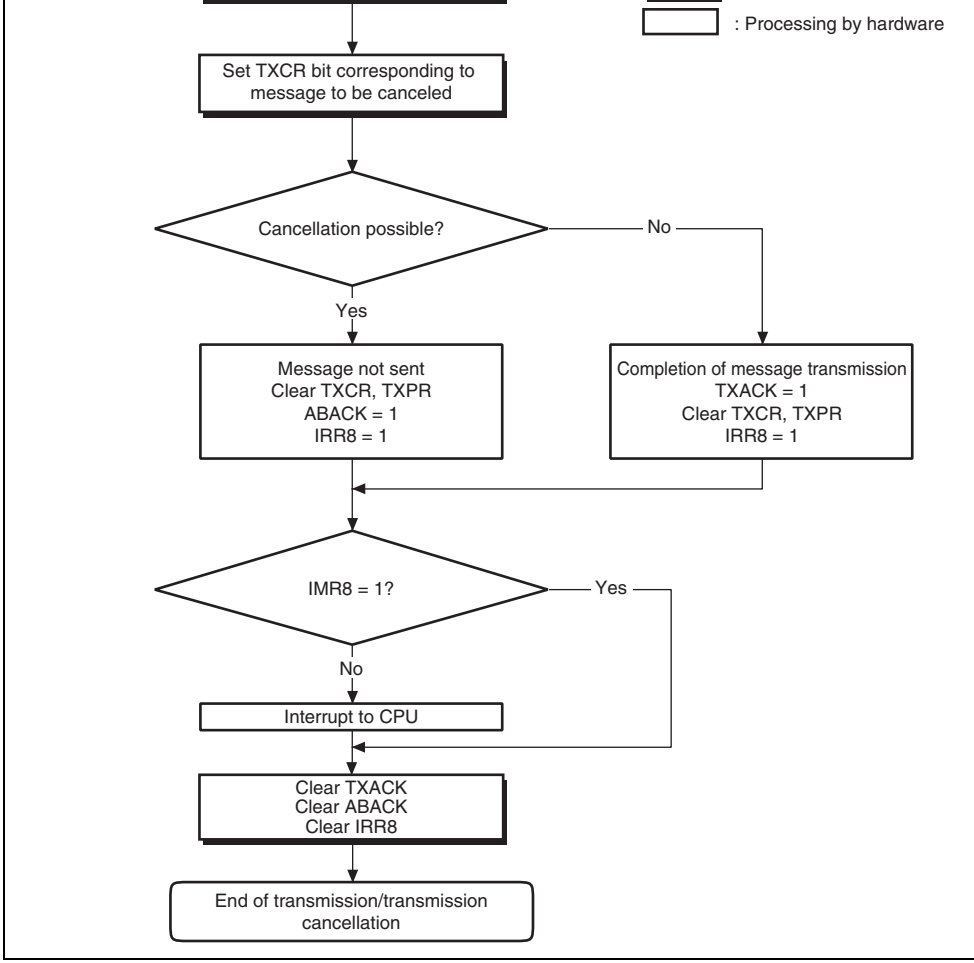


Figure 18.10 Transmit Message Cancellation Flowchart

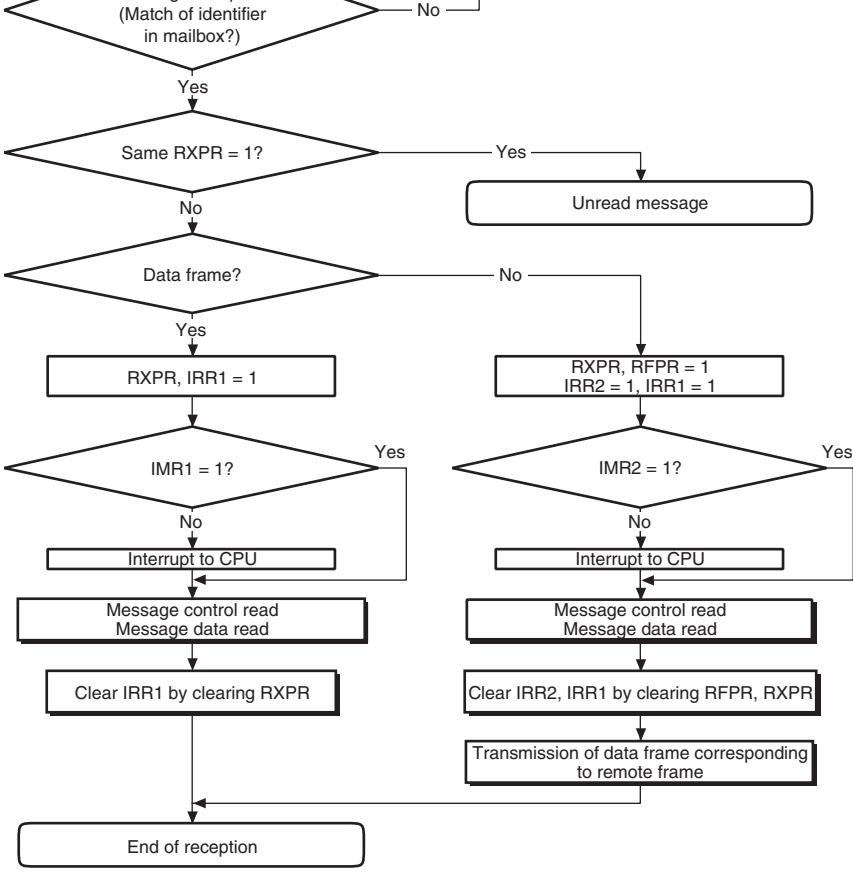


Figure 18.11 Reception Flowchart

be made. The LAFM setting can be made only for mailbox 0. By making the Don't Care for all the bits in the receive message identifier, messages of multiple identifiers can be r

Examples:

- When the identifier of mailbox 1 is 010_1010_1010 (standard format), only one kind message identifier can be received by mailbox 1:

  Identifier 1: 010_1010_1010
- When the identifier of mailbox 0 is 010_1010_1010 (standard format) and the LAFM 000_0000_0011 (0: Care, 1: Don't Care), a total of four kinds of message identifiers received by mailbox 0:

  Identifier 1: 010_1010_1000

  Identifier 2: 010_1010_1001

  Identifier 3: 010_1010_1010

  Identifier 4: 010_1010_1011

Message Reception: When a message is received, a CRC check is performed automatic result of the CRC check is normal, ACK is transmitted in the ACK field irrespective of v the message can be received or not.

- Data frame reception

  If the received message is confirmed to be error-free by the CRC check, the identifie mailbox (and also LAFM in the case of mailbox 0 only) and the identifier of the rece message, are compared. If a complete match is found, the message is stored in the m The message identifier comparison is carried out on each mailbox in turn, starting w mailbox 0 and ending with mailbox 15. If a complete match is found, the comparison that point, the message is stored in the matching mailbox, and the corresponding rec complete bit (RXPR0 to RXPR15) is set in the receive complete register (RXPR). H when a mailbox 0 LAFM comparison is carried out, even if the identifier matches, th

data frame must be stored in the data length code (DLC) in the control field.

When a remote frame (RTR = recessive) is received, the corresponding bit is set in the request wait register (RFPR). If the corresponding bit (MBIMR0 to MBIMR15) in the interrupt mask register (MBIMR) and the remote frame request interrupt mask (IRR2) interrupt mask register (IMR) are set to the interrupt enable value at this time, an inter be sent to the CPU.

Unread Message Overwrite: If the receive message identifier matches the mailbox iden receive message is stored in the mailbox regardless of whether the mailbox contains an u message or not. If a message overwrite occurs, the corresponding bit (UMSR0 to UMSR in the unread message register (UMSR). In overwriting an unread message, when a new is received before the corresponding bit in the receive complete register (RXPR) has been the unread message register (UMSR) is set. If the unread interrupt flag (IRR9) in the inter mask register (IMR) is set to the interrupt enable value at this time, an interrupt can be se CPU. Figure 18.12 shows a flowchart for unread message overwriting.

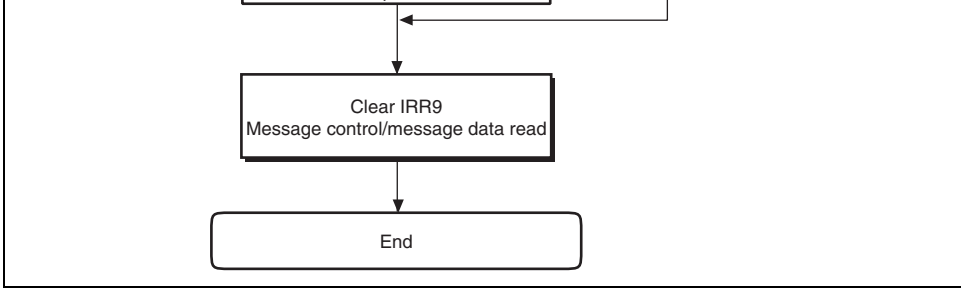


Figure 18.12 Unread Message Overwrite Flowchart

18.4.5 HCAN Sleep Mode

The HCAN is provided with an HCAN sleep mode that places the HCAN module in the state in order to reduce current dissipation. Figure 18.13 shows a flowchart of the HCAN mode.

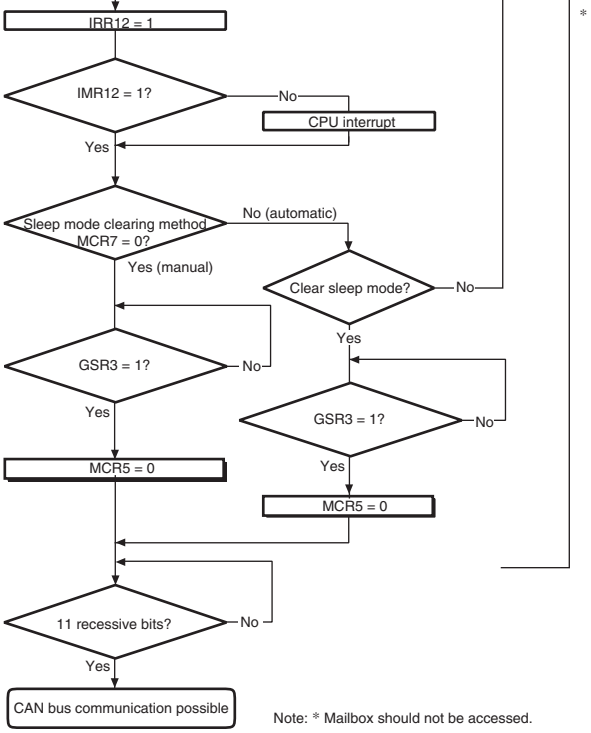


Figure 18.13 HCAN Sleep Mode Flowchart

HCAN sleep mode is entered by setting the HCAN sleep mode bit (MCR5) to 1 in the ma control register (MCR). If the CAN bus is operating, the transition to HCAN sleep mode delayed until the bus becomes idle.

Either of the following methods of clearing HCAN sleep mode can be selected:

messages will be received normally from the second message onward. When a change is on the CAN bus in HCAN sleep mode, the bus operation interrupt flag (IRR12) is set in interrupt register (IRR). If the bus interrupt mask (IMR12) in the interrupt mask register set to the interrupt enable value at this time, an interrupt can be sent to the CPU.

18.4.6 HCAN Halt Mode

The HCAN halt mode is provided to enable mailbox settings to be changed without perf HCAN hardware or software reset. Figure 18.14 shows a flowchart of the HCAN halt m

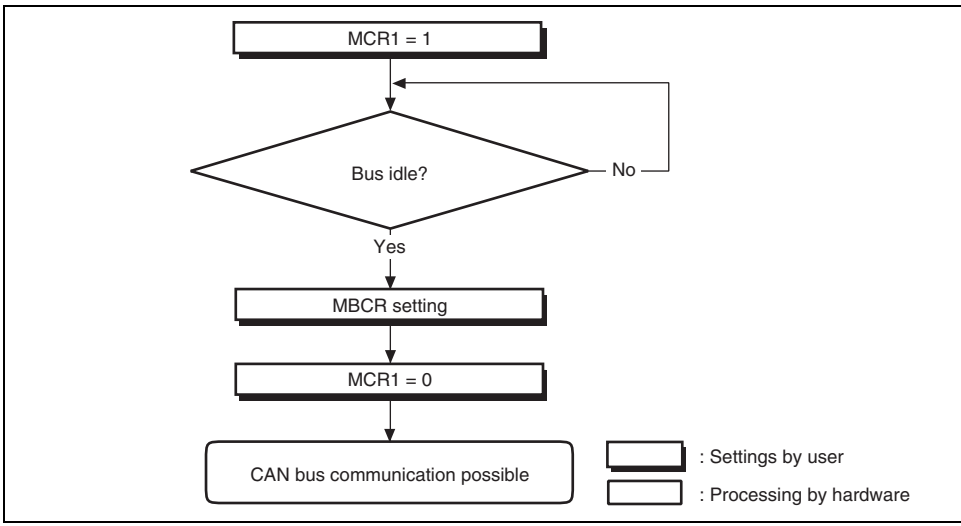


Figure 18.14 HCAN Halt Mode Flowchart

interrupt source, see section 5, Interrupt Controller.

Table 18.4 HCAN Interrupt Sources

Name	Description	Interrupt Flag	DTC Act
ERS0/OVR0	Error passive interrupt (TEC ≥ 128 or REC ≥ 128)	IRR5	Not
	Bus off interrupt (TEC ≥ 256)	IRR6	
	Reset process interrupt by power-on reset	IRR0	
	Remote frame reception interrupt	IRR2	
	Error warning interrupt (TEC ≥ 96)	IRR3	
	Error warning interrupt (REC ≥ 96)	IRR4	
	Overload frame transmission interrupt	IRR7	
	Unread message overwrite interrupt	IRR9	
	CAN bus operation in HCAN sleep mode interrupt	IRR12	
RM0	Mailbox 0 message reception interrupt	IRR1	Pos
RM1	Mailbox 1 to 15 message reception interrupt	IRR1	Not
SLE0	Message transmission/cancellation interrupt	IRR8	

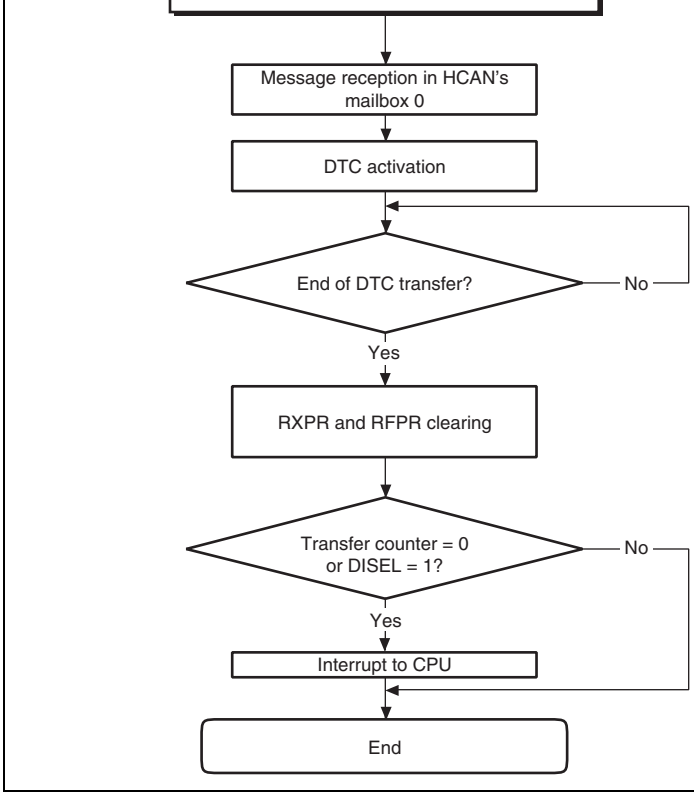


Figure 18.15 DTC Transfer Flowchart

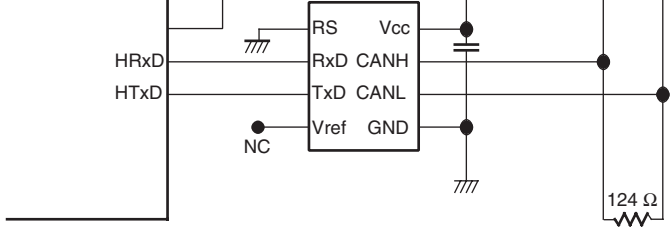


Figure 18.16 High-Speed Interface Using PCA82C250

18.8 Usage Notes

18.8.1 Module Stop Mode Setting

HCAN operation can be disabled or enabled using the module stop control register. The i
setting is for HCAN operation to be halted. Register access is enabled by clearing module
mode. For details, see section 22, Power-Down Modes.

18.8.2 Reset

The HCAN is reset by a power-on reset or in hardware standby mode, software standby m
watch mode, or module stop mode. All the registers are initialized in a reset, however ma
(message control (MCx[x])/message data (MDx[x])) are not. After power-on, mailboxes
control (MCx[x])/message data (MDx[x])) are initialized, and their values are undefined.
Therefore, mailbox initialization must always be carried out after a power-on reset or rec
from hardware standby mode, software standby mode, watch mode, or module stop mode
reset interrupt flag (IRR0) is always set after a power-on reset or recovery from software

18.8.4 Interrupts

When the mailbox interrupt mask register (MBIMR) is set, the interrupt register (IRR8,
set by reception completion, transmission completion, or transmission cancellation for t
mailboxes.

18.8.5 Error Counters

In the case of error active and error passive, REC and TEC normally count up and down
bus-off state, 11-bit recessive sequences are counted (REC + 1) using REC. If REC reac
during the count, IRR4 and GSR1 are set.

18.8.6 Register Access

Byte or word access can be used on all HCAN registers. Longword access cannot be use

18.8.7 HCAN Medium-Speed Mode

In medium-speed mode, neither read nor write is possible for the HCAN registers.

18.8.8 Register Hold in Standby Modes and Watch Mode

All HCAN registers except the message control and message data are initialized in hardw
standby mode, software standby mode, watch mode, or module stop mode.

- The HRxD pin is stacked to 1 because of a CAN bus error, etc.
- There is at least one mailbox waiting for transmission or being transmitted.
- The message transmission in a mailbox being transmitted is canceled by TXCR.

  If this occurs, transmission is canceled. However, since TXPR and TXCR states are in wrongly that a message is being cancelled, transmission cannot be restarted even if th state of the HRxD pin is canceled and the CAN bus recovers the normal state. If there least two transmission messages, a message which is not being transmitted is canceled message being transmitted retains its state.

To avoid this, one of the following countermeasures must be executed.

- Transmission must not be canceled by TXCR. When transmission is normally comple the CAN bus has recovered, TXPR is cleared and the HCAN recovers the normal stat
- To cancel transmission, the corresponding bit to TXCR must be written to 1 continuo the bit becomes 0. TXPR and TXCR are cleared and the HCAN recovers the normal s

2. When the bus-off state is entered while TXPR is set and the transmit wait state is ente internal state machine does not operate even if TXCR is set during the bus-off state. T transmission cannot be canceled. The message can be canceled when one message is transmitted or a transmission error occurs after the bus-off state is recovered. To clear message after the bus-off state is recovered, the following countermeasure must be ex

- A transmit wait message must be cleared by resetting the HCAN during the bus-off p

  To reset the HCAN, the module stop bit (MSTPC2 in MSTPCRC) must be set or clea this case, the HCAN is entirely reset. Therefore the initial settings must be made agai

is cancelled (TXPR and TXCR are set), and then transmitting a message of the highe
is cancelled in the first transmission setting.

When TXPR and another TXPR are set, or TXPR and TXCR are set under the all above
conditions, the message ID of the highest priority, which is selected in the second transn
setting or the set transmission cancellation, is damaged. That is, upper five bits of the m
which is selected in the first transmission, is set in the upper five bits of the ID. After th
five bits of the transmit message ID selected in the first transmission setting is transmitte
CAN bus, the lower six bits of the transmit message ID selected in the second transmiss
or the set transmission cancellation and transmit data (maximum eight bytes) are output
CAN bus. The CRC error is not occurred, since CRC transmits reconstructed transmit m
with ID.

Note that after the damaged transmit message is transmitted, the transmit message of the
highest priority, which is selected in the second transmission setting or the set transmiss
cancellation, is output. The message of the highest priority, which is selected in the seco
transmission setting or the set transmission cancellation, is not output.

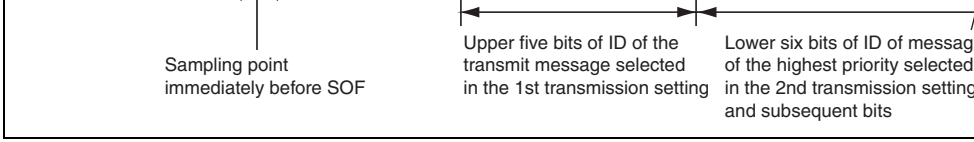


Figure 18.17 HCAN Transmit Procedure

When the interval between two transmissions, or between transmission and transmission cancellation is same or shorter than following, condition 2 described in above is satisfied consequently error may occur.

Table 18.5 Interval Limitation between TXPR and TXPR or between TXPR and T

Baud Rate (bps)	Set Interval (μs)
1 M	50
500 k	50
250 k	50

Interval (which is for preventing error) is changed depending on the following conditions

- The number of buffers (HCAN has 16 buffers)
- Data transmission: Mailbox (hereafter MB) number order, ID priority order
- HCAN operating clock
- CAN bus baud rate
- Bit timing (TSEG1, TSEG2)
- The number of MBs transmitted to the first TXPR and second TXPR
- The number of CPU accesses to MB in words or bytes after TXPR or TXCR set

When canceling an HCAN software reset or HCAN sleep mode (MCR0 = 0 or MCR5 =
confirm that the reset status bit (GSR3) is set to 1.

18.8.13 Accessing Mailboxes in HCAN Sleep Mode

Mailboxes should not be accessed in HCAN sleep mode. If mailboxes are accessed in H
sleep mode, the CPU may halt. When registers are accessed in HCAN sleep mode, the C
not halt. Also, when mailboxes are accessed not in HCAN sleep mode, the CPU does no

MAT is initiated first. The MAT can be switched by using the bank-switching metho
initiation.

— The user memory MAT is initiated at a power-on reset in user mode:

512 kbytes (H8S/2556, H8S/2552, and H8S/2506)

384 kbytes (H8S/2551 and H8S/2505)

— The user boot memory MAT is initiated at a power-on reset in user boot mode: 8

- Three on-board programming modes and one off-board programming mode
  — Boot mode

  This mode is a program mode that uses an on-chip SCI interface. The user MAT
  boot MAT can be programmed. This mode can automatically adjust the bit rate b
  host and this LSI.

  — User program mode

  The user MAT can be programmed by using the optional interface.

  — User boot mode

  The user boot program of the optional interface can be made and the user MAT c
  programmed.
- One off-board programming mode
  — Programmer mode

  This mode uses the PROM programmer. The user MAT and user boot MAT can
  programmed.
- Programming/erasing interface by the download of on-chip program

  This LSI has a dedicated programming/erasing program. After downloading this pro
  the on-chip RAM, programming/erasing can be performed by setting the argument p
  The user branch is also supported.

Software protect by register setting is provided to set the protect status of programmin flash memory

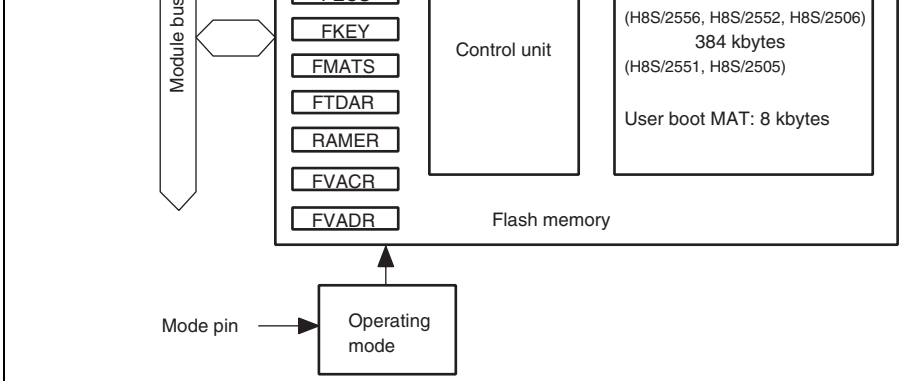
When an error is detected, the mode is changed to error protect mode to abort programming/erasing processing.

- Programming/erasing time

  The flash memory programming time is 3 ms (typ) in 128-byte simultaneous program 25 µs per byte. The erasing time is 1000 ms (typ) per 64-kbyte block.

- Number of programming

  The number of flash memory programming can be minimum 100 times.



Legend:
FCCS: Flash code control status register
FPCS: Flash program code select register
FECS: Flash erase code select register
FKEY: Flash key code register
FMATS: Flash MAT select register
FTDAR: Flash transfer destination address register
RAMER: RAM emulation register
FVACR: Flash vector address control register
FVADR: Flash vector address data register

Note: To read from or write to the above registers except RAMER, the FLSHE bit in the system control register 2 (SYSCR2) must be set to 1. When FLSHE is 1, some TPU control registers (H'FFFE80 to H'FFFEB1) cannot be accessed. The FLSHE bit should be cleared before the TPU registers are accessed.

Figure 20.1 Block Diagram of Flash Memory

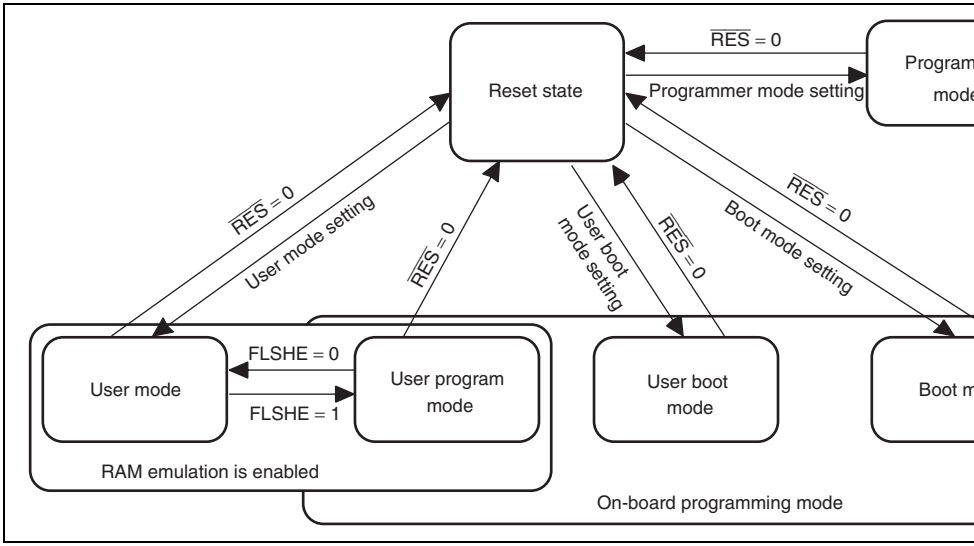


Figure 20.2 Mode Transition of Flash Memory

3. In case of On-chip ROM valid mode, User program mode and Boot mode, wh
MD0 pin sets to 0, the mode will be Expanded mode, otherwise, when the pin
the mode will be Single chip mode. However, in case of User boot mode, ther
Expanded mode.

Programming/ erasing enable MAT	User MAT User boot MAT	User MAT	User MAT	User MA User boo
All erasure	○ (Automatic)	○	○	○ (Autom
Block division erasure	○ *1	○	○	×
Program data transfer	From host via SCI	From optional device via RAM	From optional device via RAM	Via progr
User branch	×	×	×	×
RAM emulation	×	○	×	×
Reset initiation MAT	Embedded program storage MAT	User MAT	User boot MAT*2	—
Transition to user mode	Changing mode setting and reset	Changing FLSHE setting	Changing mode setting and reset	—

Notes: 1. All-erasure is performed. After that, the specified block can be erased.
2. Firstly, the reset vector is fetched from the embedded program storage MAT. A flash memory related registers are checked, the reset vector is fetched from th boot MAT.

- The user boot MAT can be programmed or erased only in boot mode and programme
- The user MAT and user boot MAT are erased in boot mode. Then, the user MAT and MAT can be programmed by means of the command method. However, the contents MAT cannot be read until this state.
  Only user boot MAT is programmed and the user MAT is programmed in user boot m only user MAT is programmed because user boot mode is not used.
- The boot operation of the optional interface can be performed by the mode pin setting from user program mode in user boot mode.

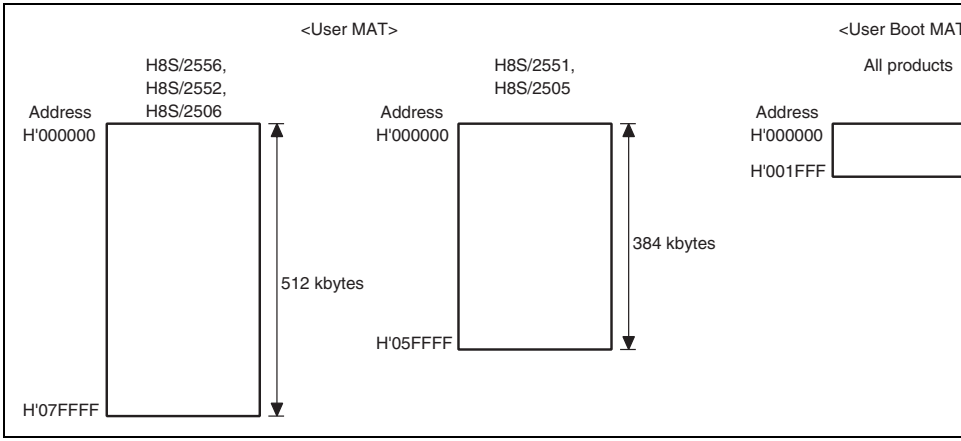
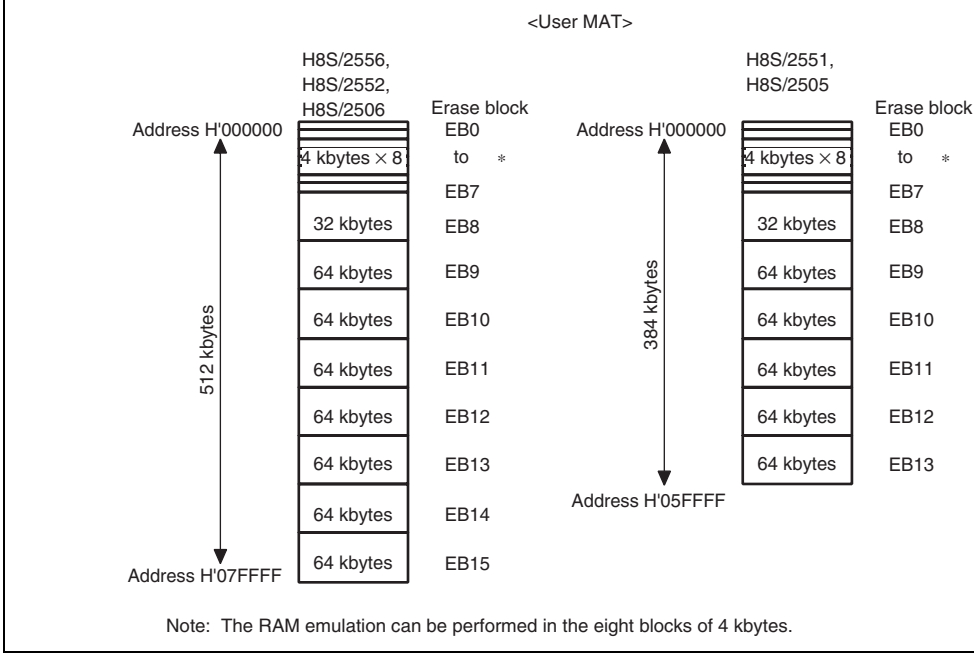


Figure 20.3 Flash Memory Configuration

The size of the user MAT is different from that of the user boot MAT. An address which the size of the 8-kbyte user boot MAT should not be accessed. If the attempt is made, da as undefined value.



Note: The RAM emulation can be performed in the eight blocks of 4 kbytes.

Figure 20.4 Block Division of User MAT

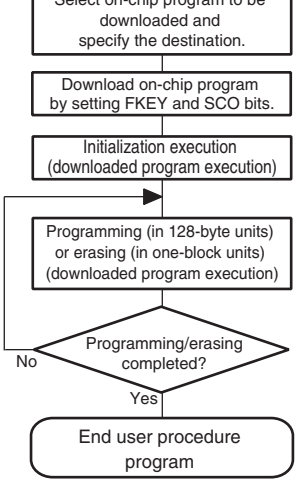


Figure 20.5 Overview of User Procedure Program

1. Selection of on-chip program to be downloaded

   This LSI has programming/erasing programs which can be downloaded to the on-ch
   The on-chip program to be downloaded is selected by setting the corresponding bits
   programming/erasing interface register. The address of the programming destination
   specified by the FTDAR.

2. Download of on-chip program

   The on-chip program is automatically downloaded by setting the SCO bit in the flash
   register (FKEY) and the flash control register (FCCS) of the programming/erasing in
   register.

   The flash memory is replaced to the embedded program storage area when downloa
   the flash memory cannot be read when programming/erasing, the procedure program

4. Programming/erasing execution

   To execute programming/erasing, it is necessary to enter user program mode by settin
   FLSHE bit in SYSCR2 to 1.

   The program data/programming destination address is specified in 128-byte units whe
   programming.

   The block to be erased is specified in erase-block units when erasing.

   These specifications are set by using the programming/erasing interface parameter an
   chip program is initiated. The on-chip program is executed by using the JSR or BSR
   instruction and performing the subroutine call of the specified address in the on-chip
   The execution result is returned to the programming/erasing interface parameter.

   The area to be programmed must be erased in advance when programming flash mem
   however, impossible to download at the same time the erasing program and the progr
   program. Therefore, execute the above procedures of 1 to 4 in the order of erasing firs
   programming next.

   All interrupts are prohibited during programming and erasing. Interrupts must be mas
   within the user system.

   Access in the flash memory space during programming/erasing is not guaranteed. Acc
   when the interrupt vector or the interrupt handler is in the flash memory, interrupt pro
   is not guaranteed. When NMI interrupt is inevitable during overprogramming/erasing
   such as in system error processing, set FVACR and FVADR to set the interrupt vecto
   interrupt processing routine in the on-chip RAM or the external space.

Table 20.3 shows the flash memory pin configuration.

Table 20.3 Pin Configuration

Pin Name	Abbreviation	Input/Output	Function
Reset	$\overline{\text{RES}}$	Input	Reset
Mode 2	MD2	Input	Sets operating mode of this LSI
Mode 1	MD1	Input	Sets operating mode of this LSI
Mode 0	MD0	Input	Sets operating mode of this LSI
Transmit data	TxD0	Output	Serial transmit data output (used in b mode)
Receive data	RxD0	Input	Serial receive data input (used in boo

Note: For the pin configuration in programmer mode, see section 20.9, Programmer Mo

- Flash erase code select register (FECS)
- Flash key code register (FKEY)
- Flash MAT select register (FMATS)
- Flash transfer destination address register (FTDAR)
- System control register 2 (SYSCR2)
- Flash pass and fail result (FPFR)
- Download pass and fail result (DPFR)
- Flash multipurpose address area (FMPAR)
- Flash multipurpose data destination area (FMPDR)
- Flash erase Block select (FEBS)
- Flash program and erase frequency control (FPEFEQ)
- Flash user branch address set parameter (FUBRA)
- RAM emulation register (RAMER)
- Flash vector address control register (FVACR)
- Flash vector address data register R (FVADRR)
- Flash vector address data register E (FVADRE)
- Flash vector address data register H (FVADRH)
- Flash vector address data register L (FVADRL)

There are several operating modes for accessing flash memory, for example, read mode/p mode.

There are two memory MATs: user MAT and user boot MAT. The dedicated registers/pa are allocated for each operating mode and MAT selection. The correspondence of operati and registers/parameters for use is shown in table 20.4.

Programming/ Erasing Interface Parameter	DPFR	○	—	—	—	—	—
	FPFR	—	○	○	○	—	—
	FPEFEQ	—	○	—	—	—	—
	FUBRA	—	○	—	—	—	—
	FMPAR	—	—	○	—	—	—
	FMPDR	—	—	○	—	—	—
	FEBS	—	—	—	○	—	—
RAM Emulation	RAMER	—	—	—	—	—	○

Notes: 1. The setting is required when programming or erasing user MAT in user boot m
2. The setting may be required according to the combination of initiation mode
target MAT.

20.3.1 Programming/Erasing Interface Register

The programming/erasing interface registers are as described below. They are all 8-bit r
that can be accessed in byte. Except for the FLER bit in FCCS, these registers are initial
power-on reset, in hardware standby mode, in software standby mode, or in watch mode
FLER bit is not initialized in software standby mode or in watch mode.

Flash Code Control and Status Register (FCCS): FCCS is configured by bits which r
monitor of error occurrence during programming or erasing flash memory and the down
on-chip program.

				the error protection state. This bit is initialized at tr to a power-on reset or hardware standby mode. When FLER is set to 1, high voltage is applied to t internal flash memory. To reduce the damage to fl memory, the reset must be released after the rese of 100 μs which is longer than normal. 0: Flash memory operates normally Programming/erasing protection for flash memo protection) is invalid. [Clearing condition] At a power-on reset or in hard standby mode 1: Indicates an error occurs during programming/e flash memory. Programming/erasing protection for flash memo protection) is valid. [Setting condition] See section 20.5.3, Error Protec
3 to 1	—	All 0	R	Reserved These bits are always read as 0. The write value s always be 0.

Four NOP instructions must be executed immedi
setting this bit to 1.

Since this bit is cleared to 0 when download is co
this bit cannot be read as 1.

0: Download of the on-chip programming/erasing
to the on-chip RAM is not executed

[Clear condition] When download is completed

1: Request that the on-chip programming/erasing
is downloaded to the on-chip RAM is occurred

[Set conditions] When all of the following conditio
satisfied and 1 is written to this bit

- H'A5 is written to FKEY
- During execution in the on-chip RAM
- Not in RAM emulation mode (RAMS in RAME

				[Clear condition] When transfer is completed
				1: On-chip programming program is selected

Flash Erase Code Select Register (FECS): FECS selects download of the on-chip erasi program.

Bit	Bit Name	Initial Value	R/W	Description
7 to 1	—	All 0	R	Reserved
				These bits are always read as 0. The write value s always be 0.
0	EPVB	0	R/W	Erase Pulse Verify Block
				Selects the erasing program.
				0: On-chip erasing program is not selected
				[Clear condition] When transfer is completed
				1: On-chip erasing program is selected

				the on-chip RAM cannot be executed.
3	K3	0	R/W	Only when H'5A is written, programming/erasing
2	K2	0	R/W	executed. Even if the on-chip programming/erasi
1	K1	0	R/W	program is executed, the flash memory cannot be
0	K0	0	R/W	programmed or erased when the value other than written to FKEY.

H'A5: Writing to the SCO bit is enabled (The SCO cannot be set by the value other than H'A5

H'5A: Programming/erasing is enabled (The valu than H'5A is in software protection state.)

H'00: Initial value

2	MS2	0	R/W	between User MAT and User Boot MAT. (The use
1	MS1	0/1*	R/W	MAT cannot be programmed in user program mod
0	MS0	0	R/W	boot MAT is selected by FMATS. The user boot M
				be programmed in boot mode or in programmer m

H'AA: The user boot MAT is selected (in user-MAT selection state when the value of these bits than H'AA)

Initial value when these bits are initiated in u mode.

H'00: Initial value when these bits are initiated in a except for user boot mode (in user-MAT sel state)

[Programmable condition] These bits are in the ex state in the on-chip RAM.

Note: Set to 1 when in user boot mode, otherwise set to 0.

Flash Transfer Destination Address Register (FTDAR): FTDAR is a register that spec address to download an on-chip program. This register must be specified before setting th bit in FCCS to 1.

				0: The value specified by bits TDA6 to TDA0 is w range.
				1: The value specified by is TDA6 to TDA0 is ove range (H'08 to H'FF) and the download is stop
6	TDA6	0	R/W	Transfer Destination Address
5	TDA5	0	R/W	Specifies the start address to download an on-ch
4	TDA4	0	R/W	program. H'00 to H'07 can be specified meaning start address in the on-chip RAM space can be s
3	TDA3	0	R/W	units of 4 kbytes.
2	TDA2	0	R/W	H'00: H'FF9000 is specified as a start address to
1	TDA1	0	R/W	an on-chip program.
0	TDA0	0	R/W	H'01: H'FFA000 is specified as a start address to an on-chip program.
				H'02: H'FFB000 is specified as a start address to an on-chip program.
				H'03: H'FFC000 is specified as a start address to an on-chip program.
				H'04: H'FFD000 is specified as a start address to an on-chip program.
				H'05: H'FFE000 is specified as a start address to an on-chip program.
				H'06: H'FF8000 is specified as a start address to an on-chip program.
				H'07: H'FF7000 is specified as a start address to an on-chip program.
				H'08 to H'FF: Setting prohibited. Specifying this v the TDRE bit to 1 and stops the do

				control register is not selected. In this case, the d of the flash memory control register is retained. 0: Flash control logic unit which controls H'FFFF H'FFFFAF is disabled. 1: Flash control logic unit which controls H'FFFF H'FFFFAF is enabled.
2	—	Undefined	—	Reserved The write value should always be 0.
1, 0	—	All 0	R/W	Reserved The write value should always be 0.

20.3.2 Programming/Erasing Interface Parameter

The programming/erasing interface parameter specifies the operating frequency, user bra destination address, storage place for program data, programming destination address, an block and exchanges the processing result for the downloaded on-chip program. This para uses the general registers of the CPU (ER0 and ER1) or the on-chip RAM area. The initia undefined at a power-on reset or in hardware standby mode.

When download, initialization, or on-chip program is executed, registers of the CPU exce R0L are stored. The return value of the processing result is written in R0L. Since the stac used for storing the registers except for R0L, the stack area must be saved at the processi (A maximum size of a stack area to be used is 128 bytes.)

The programming/erasing interface parameter is used in the following four items.

1. Download control
2. Initialization before programming or erasing

and fail result							
Flash pass and fail result	FPFR	—	○	○	○	R/W	Undefined
Flash programming/ erasing frequency control	FPEFEQ	—	○	—	—	R/W	Undefined
Flash user branch address set parameter	FUBRA	—	○	—	—	R/W	Undefined
Flash multi-purpose address area	FMPAR	—	—	○	—	R/W	Undefined
Flash multi-purpose data destination area	FMPDR	—	—	○	—	R/W	Undefined
Flash erase block select	FEBS	—	—	—	○	R/W	Undefined

Note: A single byte of the start address to download an on-chip program, which is speci
FTDAR.

(1) Download Control

The on-chip program is automatically downloaded by setting the SCO bit to 1. The on-c
area to be downloaded is the 2-kbyte area starting from the address specified by FTDAR
address map of the on-chip RAM, see figure 20.10.

Download control is set by the programming/erasing interface registers, and the DPFR p
indicates the return value.

7 to 3	—	—	—	Reserved Return 0
2	SS	—	R/W	Source Select Error Detect Only one type for the on-chip program which can b downloaded can be specified. When more than tw of the program are selected, the program is not se or the program is selected without mapping, error occurred. 0: Download program can be selected normally 1: Download error is occurred (multi-selection or p which is not mapped is selected)
1	FK	—	R/W	Flash Key Register Error Detect (FK) Returns the check result whether the value of FKE register is set to H'A5. 0: FKEY setting is normal (FKEY = H'A5) 1: Setting value of FKEY becomes error (FKEY = other than H'A5)
0	SF	—	R/W	Success/Fail Returns the result whether download is ended nor not. The determination result whether program tha downloaded to the on-chip RAM is read back and transferred to the on-chip RAM is returned. 0: Downloading on-chip program is ended normall error) 1: Downloading on-chip program is ended abnorm (error occurs)

This parameter sets the operating frequency of the CPU.

For the settable range of operating frequency in this LSI, see section 24.4.2, Clock Timi

Bit	Bit Name	Initial Value	R/W	Description
31 to 16	F31 to F16	—	R/W	Reserved These bit should be cleared to 0.
15 to 0	F15 to F0	—	R/W	Frequency Set Set the operating frequency of the CPU. The se value must be calculated as the following meth • The operating frequency which is shown in must be rounded in a number to three decir and be shown in a number of two decimal p • The value multiplied by 100 is converted to digit and is written to the FPEFEQ paramete register ER0). For example, when the operating frequency of t 25.000 MHz, the value is as follows. • The number to three decimal places of 25.0 rounded and the value is thus 25.00. • The formula that 25.00 × 100 = 2500 is conv the binary digit and B'0000,1001,1100,0100 is set to ER0.

program is not transferred. Be careful not to bra
area without execution codes, or runaway or des
of the on-chip program area or the stack area is
In case of runaway, the value of flash memory is
guaranteed.

During the processing in the user branch destina
not download or initialize the on-chip program or
programming/erasing program. Programming/era
the return from the user branch destination is no
guaranteed. In addition, do not modify the data p
to be programmed. Also during the processing in
branch destination, do not modify the
programming/erasing interface register or make
to RAM emulation mode. After the user branch
processing has completed, use the RTS instructi
return to programming/erasing program.

				downloaded programming/erasing-related prog 0: Setting of user branch address is normal 1: Setting of user branch address is abnormal
1	FQ	—	R/W	Frequency Error Detect Returns the check result whether the specified frequency of the CPU is in the range of the sup operating frequency. 0: Setting of operating frequency is normal 1: Setting of operating frequency is abnormal
0	SF	—	R/W	Success/Fail Indicates whether initialization is completed nor 0: Initialization is ended normally (no error) 1: Initialization is ended abnormally (error occu

of the CPU and in other than the flash memory space.

When data to be programmed does not satisfy 128 bytes, the 128-byte program data n prepared by filling with the dummy code H'FF.

The start address of the area in which the prepared program data is stored must be sto general register ER0. This parameter is called as FMPDR (flash multipurpose data de area parameter).

For details on the program processing procedure, see section 20.4.2, User Program Mode

(a) Flash multipurpose address area parameter (FMPAR: general register ER1 of CPU)

This parameter stores the start address of the programming destination on the user MAT.

When the address in the area other than flash memory space is set, an error occurs.

The start address of the programming destination must be at the 128-byte boundary. If thi boundary condition is not satisfied, an error occurs. The error occurrence is indicated by t bit in FPFR.

Bit	Bit Name	Initial Value	R/W	Description
31 to 0	MOA31 to MOA0	—	R/W	Store the start address of the programming desti on the user MAT. The consecutive 128-byte programming is executed starting from the speci address of the user MAT. Therefore, the specifie programming start address becomes a 128-byte boundary and MOA6 to MOA0 are always 0.

(c) Flash pass/fail parameter (FPFR: general register R0L of CPU)

This parameter indicates the return value of the program processing result.

Bit	Bit Name	Initial Value	R/W	Description
7	—	—	—	Reserved Return 0.
6	MD	—	R/W	Programming Mode Related Setting Error Dete Returns the check result that the error protectio not entered. For conditions to enter the error pr state, see section 20.5.3, Error Protection. 0: FLER setting is normal (FLER = 0) 1: Programming cannot be performed (FLER =

				mode. 0: Programming has ended normally 1: Programming has ended abnormally (program result is not guaranteed)
4	FK	—	R/W	Flash Key Register Error Detect Returns the check result of the value of FKEY be start of the programming processing. 0: FKEY setting is normal (FKEY = H'5A) 1: FKEY setting is error (FKEY = value other tha
3	—	—	—	Reserved Returns 0.
2	WD	—	R/W	Write Data Address Detect When the following address is specified as the s address of the storage destination of the progra an error occurs. • The address in the on-chip RAM where programming/erasing program is downloade • The address in the flash memory area 0: Setting of write data address is normal 1: Setting of write data address is abnormal

				1: Setting of programming destination address i abnormal
0	SF	—	R/W	Success/Fail Indicates whether the program processing is en normally or not. 0: Programming is ended normally (no error) 1: Programming is ended abnormally (error occ

(4) Erasure Execution

When flash memory is erased, the erase-block number on the user MAT must be passed erasing program which is downloaded. This is set to the FEBS parameter (general regist

One block is specified from the block number 0 to 15.

For details on the erasing processing procedure, see section 20.4.2, User Program Mode

4	EB4	—	R/W	EB15 block. An error occurs when the number o
3	EB3	—	R/W	0 to 15 is set.
2	EB2	—	R/W	
1	EB1	—	R/W	
0	EB0	—	R/W	

(b) Flash pass/fail parameter (FPFR: general register R0L of CPU)

This parameter returns value of the erasing processing result.

Bit	Bit Name	Initial Value	R/W	Description
7	—	—	—	Reserved Return 0.
6	MD	—	—	Erasure Mode Related Setting Error Detect Returns the check result of whether the error pro state is entered. For conditions to enter the error protection state, see section 20.5.3, Error Protec 0: FLER setting is normal (FLER = 0) 1: FLER = 1 and erasure cannot be performed

				0: Erasure has ended normally 1: Erasure has ended abnormally (erasure resu guaranteed)
4	FK	—	R/W	Flash Key Register Error Detect Returns the check result of FKEY value before the erasing processing. 0: FKEY setting is normal (FKEY = H'5A) 1: FKEY setting is error (FKEY = value other th
3	EB	—	R/W	Erase Block Select Error Detect Returns the check result whether the specified block number is in the block range of the user M 0: Setting of erase-block number is normal 1: Setting of erase-block number is abnormal
2, 1	—	—	—	Reserved Return 0.
0	SF	—	R/W	Success/Fail Indicates whether the erasing processing is enc normally or not. 0: Erasure is ended normally (no error) 1: Erasure is ended abnormally (error occurs)

Bit	Bit Name	Initial Value	R/W	Description
7 to 5	—	All 0	R	Reserved These bits are always read as 0. The write value always be 0.
4	—	0	R/W	Reserved The write value should always be 0.
3	RAMS	—	R/W	RAM Select Sets whether the user MAT is emulated or not. W RAMS = 1, all blocks of the user boot MAT are in programming/erasing protection state. 0: RAM emulation function is invalid All blocks of the user MAT are not in the programming/erasing protection state 1: RAM emulation function is valid All blocks of the user MAT are in the programming/erasing protection state
2	RAM2	0	R/W	User MAT Area Select
1	RAM1	0	R/W	These bits are used with bit 3 and select the use
0	RAM0	0	R/W	area to be overlapped with the on-chip RAM. (Se 20.6.)

H'006000 to H'006FFF	EB6 (4 kbytes)	1	1	1	0
H'007000 to H'007FFF	EB7 (4 kbytes)	1	1	1	1

Legend:

*: Don't care

20.3.4 Flash Vector Address Control Register (FVACR)

FVACR modifies the space from which the vector table data of the NMI interrupts is rea
Normally the vector table data is read from the address spaces from H'00001C to H'000
However, the vector table can be read from the internal I/O register (FVADRR to FVAD
the FVACR setting. FVACR is initialized to H'00 at a power-on reset or in hardware sta
mode.

All interrupts including NMI must be prohibited in the programming/erasing processing
downloading on-chip program. When the NMI interrupt is necessary such as in the syste
processing, FVACR and FVADRR to FVADRL must be set and the interrupt exception
processing routine must be set in the on-chip RAM space or in the external space.

Bit	Bit Name	Initial Value	R/W	Description
7	FVCHGE	0	R/W	Vector Switch Function Valid Selects whether the function for modifying the sp which the vector table data is read is valid or inva FVCHGE = 1, the vector table data can be read f internal I/O register (FVADRR to FVADRL). 0: Function for modifying the space from which th table data is read is invalid (Initial value) 1: Function for modifying the space from which th table data is read is valid

consists of four 8-bit registers: FVADRR, FVADRE, FVADRH, and FVADRL. This reg
initialized to H'00000000 at a power-on reset or in hardware standby mode.

- FVADRR

Bit	Bit Name	Initial Value	R/W	Description
31 to 24	—	All 0	R/W	Set the vector address.

- FVADRE

Bit	Bit Name	Initial Value	R/W	Description
23 to 16	—	All 0	R/W	Set the vector address.

- FVADRH

Bit	Bit Name	Initial Value	R/W	Description
15 to 8	—	All 0	R/W	Set the vector address.

- FVADRL

Bit	Bit Name	Initial Value	R/W	Description
7 to 0	—	All 0	R/W	Set the vector address.

20.4.1 Boot Mode

Boot mode executes programming/erasing user MAT and user boot MAT by means of t
command and program data transmitted from the host using the on-chip SCI. The tool fo
transmitting the control command and program data must be prepared in the host. The S
communication mode is set to asynchronous mode. When reset start is executed after thi
pin is set in boot mode, the boot program in the microcomputer is initiated. After the SC
is automatically adjusted, the communication with the host is executed by means of the
command method.

The system configuration diagram in boot mode is shown in figure 20.6. For details on t
setting in boot mode, see table 20.1. The NMI and other interrupts are ignored in boot m
However, the NMI and other interrupts should be disabled in the user system.

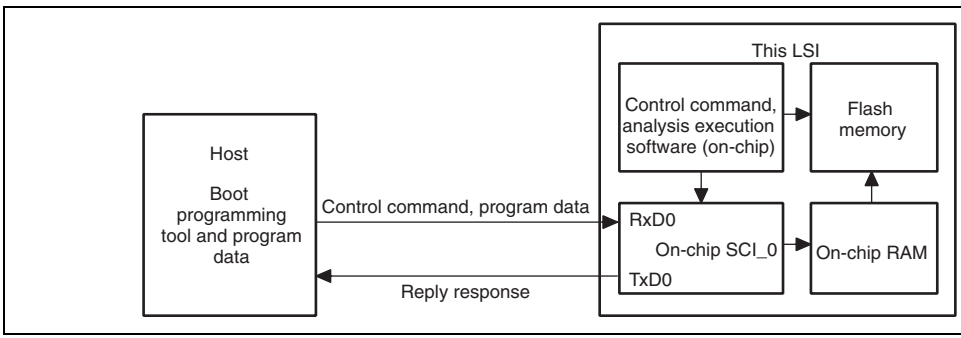


Figure 20.6 System Configuration in Boot Mode

set to 9,600 bps or 19,200 bps.

The system clock frequency, which can automatically adjust the transfer bit rate of the ho
the bit rate of this LSI, is shown in table 20.7. Boot mode must be initiated in the range o
system clock.

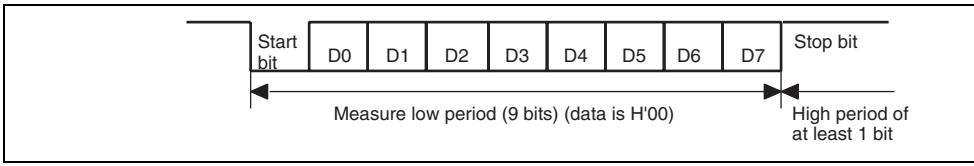


Figure 20.7 Automatic-Bit-Rate Adjustment Operation of SCI

Table 20.7 System Clock Frequency for Automatic-Bit-Rate Adjustment by This L

Bit Rate of Host	System Clock Frequency
9,600 bps	10 to 26 MHz
19,200 bps	16 to 26 MHz

(2) State Transition Diagram

The overview of the state transition diagram after boot mode is initiated is shown in figur

1. Bit rate adjustment
   After boot mode is initiated, the bit rate of the SCI interface is adjusted with that of th
2. Waiting for inquiry set command
   For inquiries about user-MAT size and configuration, MAT start address, and support
   the required information is transmitted to the host.
3. Automatic erasure of all user MAT and user boot MAT

Then the state for waiting erase-block data is returned to the state for waiting programming/erasing command. The erasure must be used when the specified bl programmed without a reset start after programming is executed in boot mode. W programming can be executed by only one operation, all blocks are erased before for waiting programming/erasing/other command is entered. The erasing operatio required.

— There are many commands other than programming/erasing. Examples are sum c blank check (erasure check), and memory read of the user MAT/user boot MAT acquisition of current status information.

Note that memory read of the user MAT/user boot MAT can only read the programmed all user MAT/user boot MAT has automatically been erased.

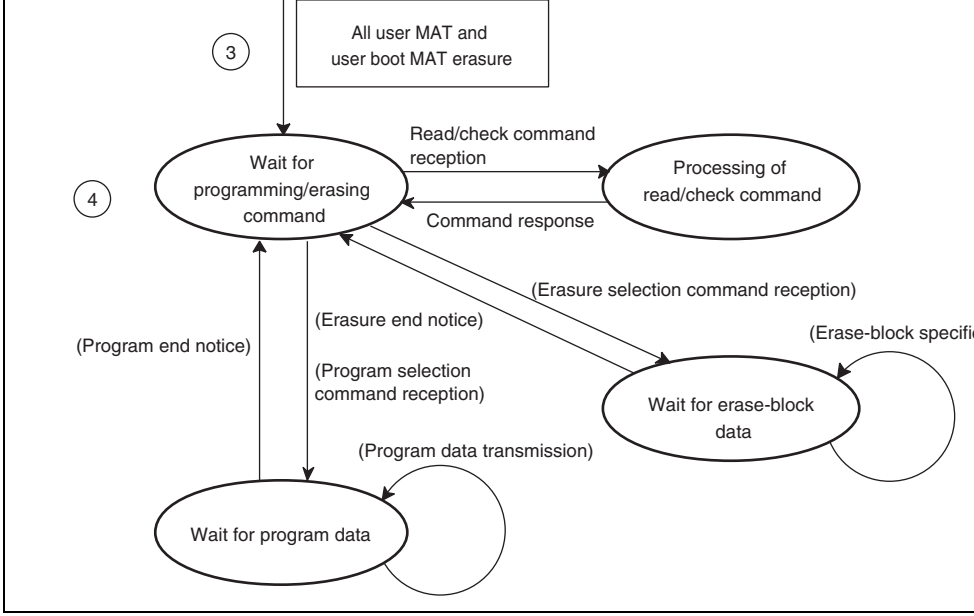
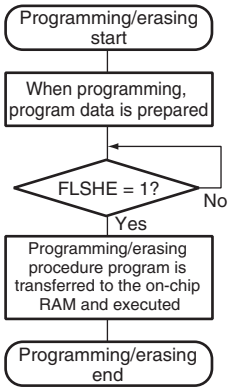


Figure 20.8 Overview of Boot Mode State Transition Diagram

reset input period of 100 µs, which is longer than usual.



1. RAM emulation mode must be canceled in advance. Download cannot be executed in emulation mode.
2. When the program data is made by means of emulation, change the download destination by the FTDAR register. Note that the download area and the emulation area will overlap if FTDAR is H'04.
3. Set FLSHE bit in SYSCR2 register to 1.
4. Programming/erasing is executed only in the on-chip RAM. However, if program data is in a consecutive area and can be accessed by the MOV.B instruction of the CPU like SRAM or ROM, the program data can be in an external space.
5. After programming/erasing is finished, the FLSHE bit is cleared to 0 and protected.

Figure 20.9 Programming/Erasing Overview Flow

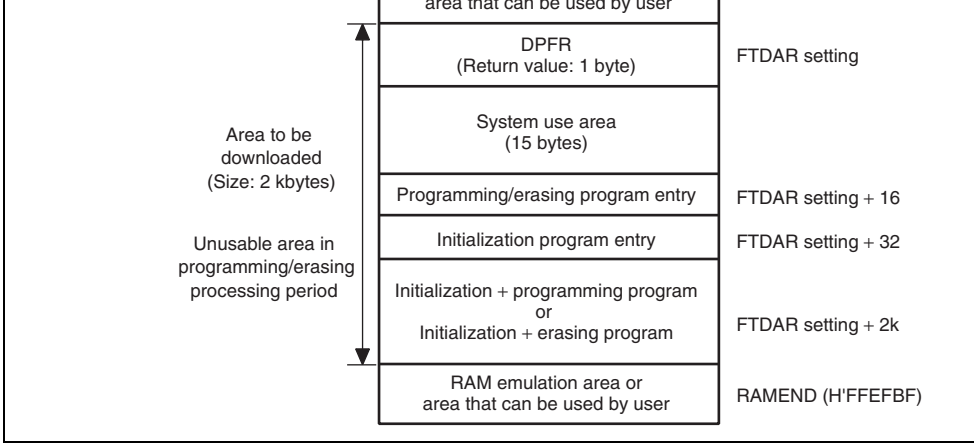


Figure 20.10 RAM Map when Programming/Erasing Is Executed

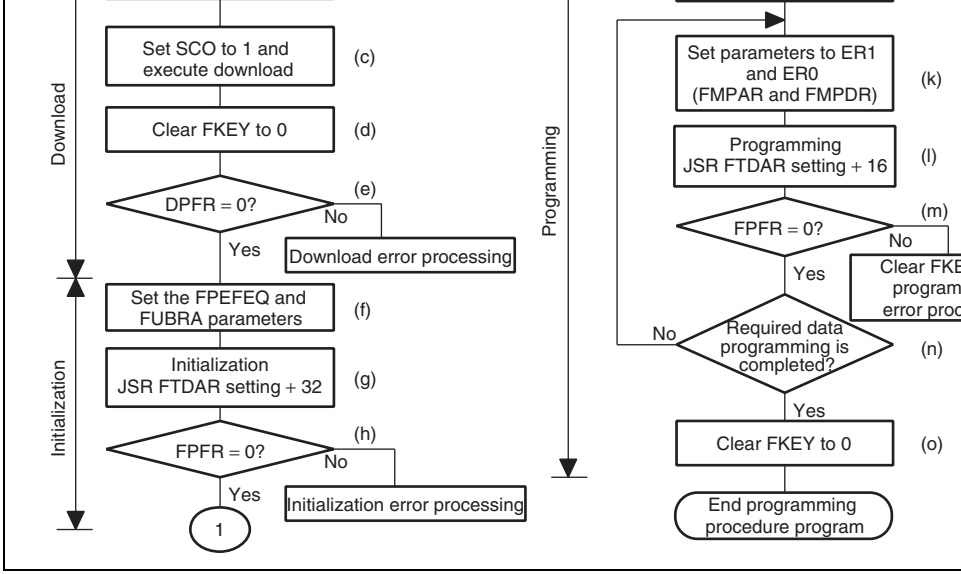


Figure 20.11 Programming Procedure

The procedure program must be executed in an area other than the flash memory to be programmed. Especially the part where the SCO bit in FCCS is set to 1 for downloading executed in the on-chip RAM.

The area that can be executed in the steps of the user procedure program (on-chip RAM MAT, and external space) is shown in section 20.4.4, Procedure Program and Storable A Programming Data.

The following description assumes the area to be programmed on the user MAT is erase program data is prepared in the consecutive area. When erasing is not executed, erasing executed before writing.

download is not performed and a download error is returned to the SS bit in DPFR. T
address of a download destination is specified by FTDAR.

(b) Program H'A5 in FKEY

If H'A5 is not written to FKEY for protection, 1 cannot be written to the SCO bit for c
request.

(c) 1 is written to the SCO bit of FCCS and then download is executed.

To write 1 to the SCO bit, the following conditions must be satisfied.

— RAM emulation mode is canceled.
— H'A5 is written to FKEY.
— The SCO bit writing is executed in the on-chip RAM.

When the SCO bit is set to 1, download is started automatically. When the program ex
processing returned to the user procedure program, the SCO is cleared to 0. Therefore
SCO bit cannot be confirmed to be 1 in the user procedure program.

The download result can be confirmed only by the return value of DPFR. Before the S
is set to 1, incorrect determination must be prevented by setting the one byte of the sta
address (to be used as DPFR) specified by FTDAR to a value other than the return va
(H'FF).

When download is executed, particular interrupt processing, which is accompanied by
switch as described below, is performed as an internal microcomputer processing. Fou
instructions are executed immediately after the instructions that set the SCO bit to 1.

— The user-MAT space is switched to the on-chip program storage area.
— After the selection condition of the download program and the FTDAR setting are
the transfer processing to the on-chip RAM specified by FTDAR is executed.
— FPCS, FECS, and the SCO bit in FCCS are cleared to 0.
— The return value is set to the DPFR parameter.

— Since a stack area of a maximum 128-byte is used, the area must be allocated bef
the SCO bit to 1.

— If a flash memory access by the DTC is requested during downloading, the opera
cannot be guaranteed. Therefore, an access request by the DTC must not be gene

(d) FKEY is cleared to H'00 for protection.

(e) The value of the DPFR parameter must be checked and the download result must be
confirmed.

— Check the value of the DPFR parameter (one byte of start address of the downloa
destination specified by FTDAR). If the value is H'00, download has been perfor
normally. If the value is not H'00, the source that caused download to fail can be
investigated by the description below.

— If the value of the DPFR parameter is the same as before downloading (e.g. H'FF
address setting of the download destination in FTDAR may be abnormal. In this
confirm the setting of the TDER bit (bit 7) in FTDAR.

— If the value of the DPFR parameter is different from before downloading, check
and the FK bit in the DPFR parameter to ensure that the download program selec
FKEY setting were normal, respectively.

(f) The operating frequency and user branch destination are set to the FPEFEQ and FUE
parameters for initialization.

— The current frequency of the CPU clock is set to the FPEFEQ parameter (general
ER0).

branch address setting parameter (FUBRA: General register ER1 of CPU).

(g) Initialization

When a programming program is downloaded, the initialization program is also down
the on-chip RAM. There is an entry point of the initialization program in the area fro
start address specified by FTDAR + 32 bytes of the on-chip RAM. The subroutine is
and initialization is executed by using the following steps.

```
MOV.L   #DLTOP+32,ER2          ; Set entry address to ER2
JSR     @ER2                   ; Call initialization routine
NOP
```

— The general registers other than R0L are held in the initialization program.

— R0L is a return value of the FPFR parameter.

— Since the stack area is used in the initialization program, a stack area of a maximu
bytes must be allocated in RAM.

— Interrupts can be accepted during the execution of the initialization program. The
storage area and stack area in the on-chip RAM and register values must not be de

(h) The return value in the initialization program, FPFR (general register R0L) is determi

(i) All interrupts and the use of a bus master other than the CPU are prohibited.

The specified voltage is applied for the specified time when programming or erasing.
interrupts occur or the bus mastership is moved to other than the CPU during this tim
voltage for more than the specified time will be applied and flash memory may be da
Therefore, interrupts and movement of bus mastership other than the CPU are prohibi

To prohibit the interrupt, bit 7 (I) in the condition code register (CCR) of the CPU sh
set to B'1 in interrupt control mode 0. Then interrupts other than NMI are held and ar
executed.

(k) The parameter which is required for programming is set.

The start address of the programming destination of the user MAT (FMPAR) is set t register ER1. The start address of the program data area (FMPDR) is set to general r ER0.

— Example of the FMPAR setting

FMPAR specifies the programming destination address. When an address other t the user MAT area is specified, even if the programming program is executed, programming is not executed and an error is returned to the return value paramet Since the unit is 128 bytes, the lower eight bits (A7 to A0) must be H'00 or H'80 boundary of 128 bytes.

— Example of the FMPDR setting

When the storage destination of the program data is flash memory, even if the pr execution routine is executed, programming is not executed and an error is return FPFR parameter. In this case, the program data must be transferred to the on-chi and then programming must be executed.

(l) Programming

There is an entry point of the programming program in the area from the start addres by FTDAR + 16 bytes of the on-chip RAM. The subroutine is called and programmi executed by using the following steps.

```
MOV.L   #DLTOP+16,ER2          ; Set entry address to ER2
JSR     @ER2                   ; Call programming routine
NOP
```

— The general registers other than R0L are held in the programming program.

— R0L is a return value of the FPFR parameter.

(o) After programming finishes, clear FKEY and specify software protection.
If this LSI is restarted by a power-on reset immediately after user MAT programming finished, secure a reset period (period of $\overline{RES}$ = 0) of 100 µs or longer, which is longe usual.

(3) Erasing Procedure in User Program Mode

The procedures for download, initialization, and erasing are shown in figure 20.12.

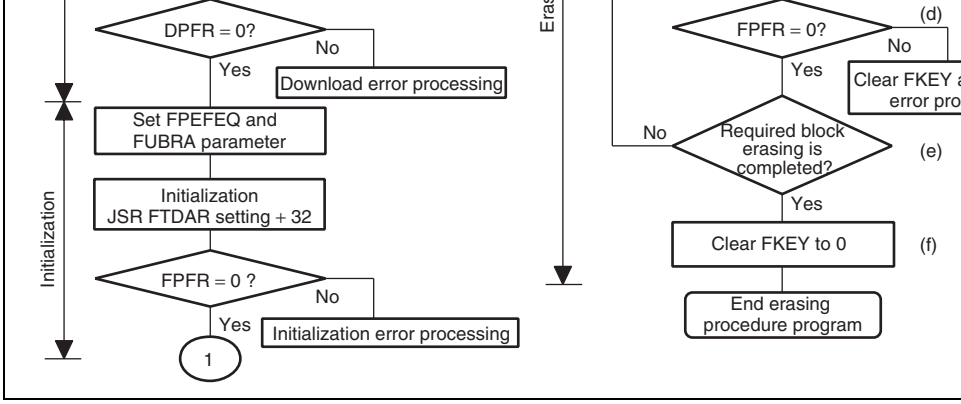


Figure 20.12 Erasing Procedure

The procedure program must be executed in an area other than the user MAT to be erase
Especially the part where the SCO bit in FCCS is set to 1 for downloading must be exec
on-chip RAM.

The area that can be executed in the steps of the user procedure program (on-chip RAM
MAT, and external space) is shown in section 20.4.4, Procedure Program and Storable A
Programming Data.

For the downloaded on-chip program area, see the RAM map for programming/erasing
20.10.

A single divided block is erased by one erasing processing. For block divisions, see figu
To erase two or more blocks, update the erase block number and perform the erasing pro
for each block.

The procedures after setting parameters for erasing programs are as follows:

(b) Set the FEBS parameter necessary for erasure

Set the erase block number of the user MAT in the flash erase block select parameter (general register ER0). If a value other than an erase block number of the user MAT i block is erased even though the erasing program is executed, and an error is returned return value parameter FPFR.

(c) Erasure

Similar to as in programming, there is an entry point of the erasing program in the are the start address of a download destination specified by FTDAR + 16 bytes of on-chip The subroutine is called and erasing is executed by using the following steps.

```
MOV.L   #DLTOP+16,ER2          ; Set entry address to ER2
JSR     @ER2                   ; Call erasing routine
NOP
```

- The general registers other than R0L are held in the erasing program.
- R0L is a return value of the FPFR parameter.
- Since the stack area is used in the erasing program, a stack area of maximum 128 must be allocated in RAM

(d) The return value in the erasing program, FPFR (general register R0L) is determined.

(e) Determine whether erasure of the necessary blocks has completed.

If more than one block is to be erased, update the FEBS parameter and repeat steps (b Blocks that have already been erased can be erased again.

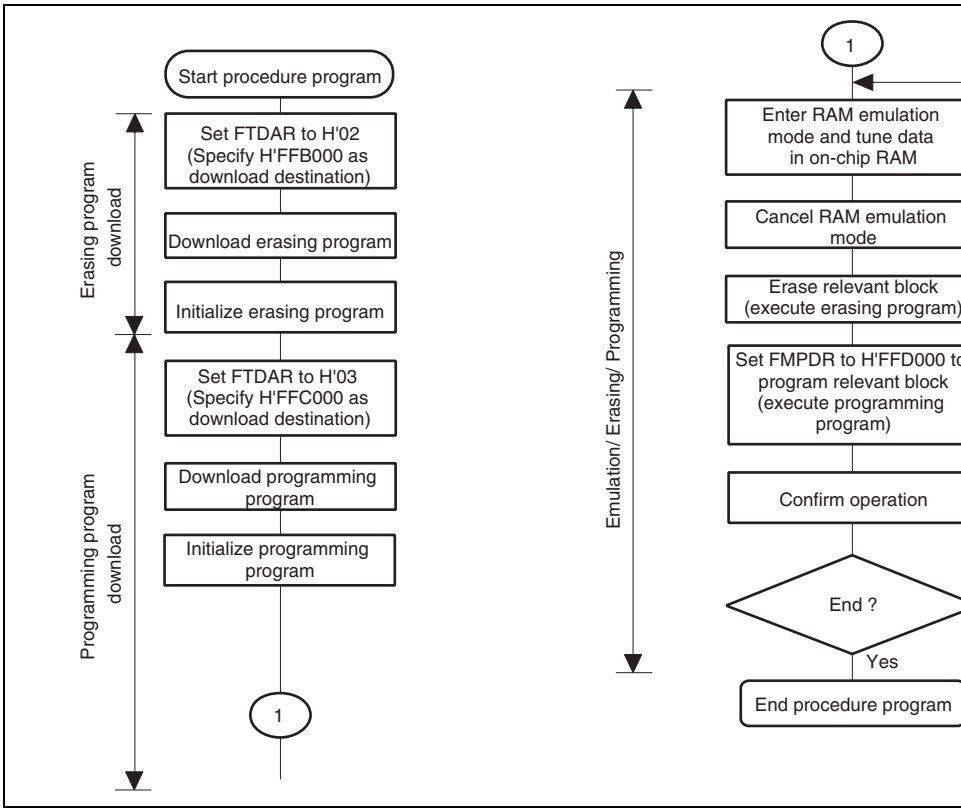


Figure 20.13 Sample Procedure of Repeating RAM Emulation, Erasing, and Prog (Overview)

In the above example, the erasing program and programming program are downloaded t excluding the 4 kbytes (H'FFD000 to H'FFDFFF) in the on-chip RAM. Download and

example) and (download start address for programming program) + 32 bytes (H'F
this example).

20.4.3 User Boot Mode

This LSI has user boot mode which is initiated with different mode pin settings than thos
program mode or boot mode. User boot mode is a user-arbitrary boot mode, unlike boot m
uses the on-chip SCI.

Only the user MAT can be programmed/erased in user boot mode. Programming/erasing
user boot MAT is only enabled in boot mode or programmer mode.

(1) User Boot Mode Initiation

For the mode pin settings to start up user boot mode, see table 20.1.

When the reset start is executed in user boot mode, the built-in check routine runs. The us
and user boot MAT states are checked by this check routine.

While the check routine is running, NMI and all other interrupts cannot be accepted.

Next, processing starts from the execution start address of the reset vector in the user boo
At this point, H'AA is set to FMATS because the execution MAT is the user boot MAT.

(2) User MAT Programming in User Boot Mode

For programming the user MAT in user boot mode, additional processing made by setting
are required: switching from user-boot-MAT selection state to user-MAT selection state,
switching back to user-boot-MAT selection state after programming completes.

Figure 20.14 shows the procedure for programming the user MAT in user boot mode.

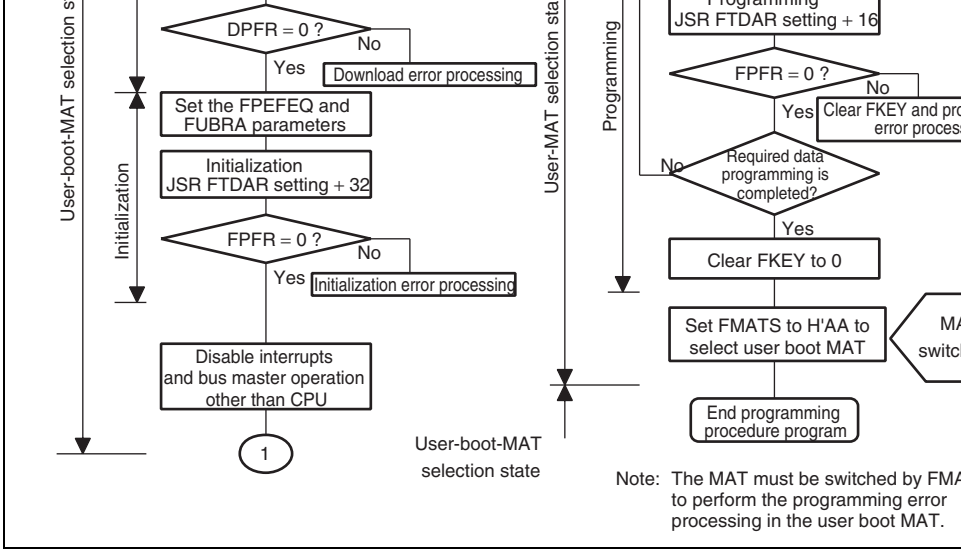


Figure 20.14 Procedure for Programming User MAT in User Boot Mode

The difference between the programming procedures in user program mode and user bo whether the MAT is switched or not as shown in figure 20.14.

In user boot mode, the user boot MAT can be seen in the flash memory space with the u hidden in the background. The user MAT and user boot MAT are switched only while t MAT is being programmed. Because the user boot MAT is hidden while the user MAT programmed, the procedure program must be located in an area other than flash memory programming completes, switch the MATs again to return to the first state.

MAT switching is enabled by writing a specific value to FMATS. However note that wh MATs are being switched, the LSI is in an unstable state, e.g. access to a MAT is not all MAT switching is completed, and if an interrupt occurs, from which MAT the interrupt

switching back to user-boot-MAT selection state after erasing completes.

Figure 20.15 shows the procedure for erasing the user MAT in user boot mode.

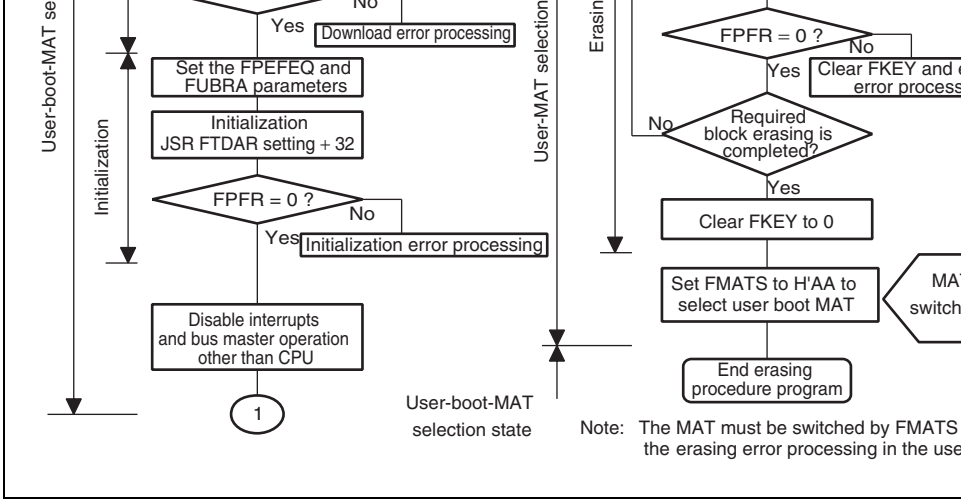


Figure 20.15 Procedure for Erasing User MAT in User Boot Mode

The difference between the erasing procedures in user program mode and user boot mod on whether the MAT is switched or not as shown in figure 20.15.

MAT switching is enabled by writing a specific value to FMATS. However note that wh MATs are being switched, the LSI is in an unstable state, e.g. access to a MAT is not all MAT switching is completed, and if an interrupt occurs, from which MAT the interrupt read is undetermined. Perform MAT switching in accordance with the description in sec Switching between User MAT and User Boot MAT.

Except for MAT switching, the erasing procedure is the same as that in user program mo

The area that can be executed in the steps of the user procedure program (on-chip RAM MAT) is shown in section 20.4.4, Procedure Program and Storable Area for Programmi

2. The on-chip programming/erasing program will use 128 bytes as a stack. So, make su this area is secured.
3. Download by setting the SCO bit to 1 will lead to switching of the MAT. If, therefore operation is used, it should be executed from the on-chip RAM.
4. The flash memory is accessible until the start of programming or erasing, that is, until result of downloading has been determined. When in single-chip mode in which the e address space is not accessible, the required procedure programs, NMI handling vecto handler, and user branch program should be transferred to the on-chip RAM before programming/erasing of the flash memory starts.
5. The flash memory is not accessible during programming/erasing operations, therefore operation program is downloaded to the on-chip RAM to be executed. The NMI-hand vector and programs such as that which activate the operation program, and NMI han should thus be stored in on-chip memory other than flash memory or the external add space.
6. After programming/erasing, the flash memory should be inhibited until FKEY is clea

   The reset state ($\overline{RES} = 0$) must be in place for more than 100 µs when the LSI mode i to reset on completion of a programming/erasing operation.

   Transitions to the reset state, and hardware standby mode are inhibited during programming/erasing. When the reset signal is accidentally input to the chip, a longer the reset state than usual (100 µs) is needed before the reset signal is released.
7. Switching of the MATs by FMATS should be needed when programming/erasing of t MAT is operated in user-boot mode. The program which switches the MATs should b executed from the on-chip RAM. See section 20.7, Switching between User MAT and Boot MAT. Please make sure you know which MAT is selected when switching betw them.
8. When the data storable area indicated by programming parameter FMPDR is within t memory area, an error will occur even when the data stored is normal. Therefore, the

Programming	Table 20.9 (1)	Table 20.9 (3)
Erasing	Table 20.9 (2)	Table 20.9 (4)

Note: * Programming/Erasing is possible to user MATs.

Table 20.9 (1) Useable Area for Programming in User Program Mode

	Storable /Executable Area				Selected I
Item	**On-Chip RAM**	**Target Flash Memory**	**External Space (Expanded Mode)**	**User MAT**	**Emb Prog Stor**
Storage Area for Program Data	○	×*	○	—	
Operation for Selection of On-chip Program to be Downloaded	○	○	○	○	
Operation for Writing H'A5 to FKEY	○	○	○	○	
Execution of Writing SCO = 1 to FCCS (Download)	○	×	×		
Operation for FKEY Clear	○	○	○	○	

Initialization				
Determination of Initialization Result	○	○	○	○
Operation for Initialization Error	○	○	○	○
NMI Handling Routine	○	×	○	○
Operation for Inhibit of Interrupt	○	○	○	○
Operation for Writing H'5A to FKEY	○	○	○	○
Operation for Settings of Program Parameter	○	×	○	○
Execution of Programming	○	×	×	○
Determination of Program Result	○	×	○	○
Operation for Program Error	○	×	○	○
Operation for FKEY Clear	○	×	○	○

Note: Transferring the data to the on-chip RAM enables this area to be used.

Execution of Writing SCO = 1 to FCCS (Download)	○	×	×	
Operation for FKEY Clear	○	○	○	○
Determination of Download Result	○	○	○	○
Operation for Download Error	○	○	○	○
Operation for Settings of Initial Parameter	○	○	○	○
Execution of Initialization	○	×	×	○
Determination of Initialization Result	○	○	○	○
Operation for Initialization Error	○	○	○	○
NMI Handling Routine	○	×	○	○
Operation for Inhibit of Interrupt	○	○	○	○
Operation for Writing H'5A to FKEY	○	○	○	○
Operation for Settings of Erasure Parameter	○	×	○	○

Table 20.9 (3) Useable Area for Programming in User Boot Mode

	Storable/Executable Area			Selected MAT	
Item	**On-Chip RAM**	**User Boot MAT**	**User MAT**	**User Boot MAT**	**Embedd Program Storage**
Storage Area for Program Data	○	×*[1]	—	—	–
Operation for Selection of On-chip Program to be Downloaded	○	○		○	
Operation for Writing H'A5 to FKEY	○	○		○	
Execution of Writing SCO = 1 to FCCS (Download)	○	×			○
Operation for FKEY Clear	○	○		○	
Determination of Download Result	○	○		○	
Operation for Download Error	○	○		○	
Operation for Settings of Initial Parameter	○	○		○	

Operation for Interrupt Inhibit	○	○		○
Switching MATs by FMATS	○	×	○	
Operation for Writing H'5A to FKEY	○	×	○	
Operation for Settings of Program Parameter	○	×	○	
Execution of Programming	○	×	○	
Determination of Program Result	○	×	○	
Operation for Program Error	○	×*2	○	
Operation for FKEY Clear	○	×	○	
Switching MATs by FMATS	○	×		○

Notes: 1. Transferring the data to the on-chip RAM enables this area to be used.
2. Switching FMATS by a program in the on-chip RAM enables this area to be u

Execution of Writing SCO = 1 to FCCS (Download)	○	×		
Operation for FKEY Clear	○	○		○
Determination of Download Result	○	○		○
Operation for Download Error	○	○		○
Operation for Settings of Initial Parameter	○	○		○
Execution of Initialization	○	×		○
Determination of Initialization Result	○	○		○
Operation for Initialization Error	○	○		○
NMI Handling Routine	○	×		○
Operation for Interrupt Inhibit	○	○		○
Switching MATs by FMATS	○	×	○	
Operation for Writing H'5A to FKEY	○	×	○	

Erasure Error				
Operation for FKEY Clear	○	×	○	
Switching MATs by FMATS	○	×		○

Note: Switching FMATS by a program in the on-chip RAM enables this area to be used

Table 20.10 Hardware Protection

Item	Description	Function to Be Prot	
		Download	Program
Reset/standby protection	• The program/erase interface registers are initialized in the power-on reset state (including a power-on reset by the WDT) and standby mode and the program/erase-protected state is entered. • The reset state will not be entered by a reset using the $\overline{\text{RES}}$ pin unless the $\overline{\text{RES}}$ pin is held low until oscillation has stabilized after power is initially supplied. In the case of a reset during operation, hold the $\overline{\text{RES}}$ pin low for the $\overline{\text{RES}}$ pulse width that is specified in the section on AC characteristics section. If a reset is input during programming or erasure, data values in the flash memory are not guaranteed. In this case, after holding the $\overline{\text{RES}}$ pin low for 100 μs or longer, execute erasure and then execute program again.	○	○

SCO bit		entered by clearing the SCO bit in FCCS which disables the downloading of the programming/erasing programs.		
Protection by the FKEY register	•	Downloading and programming/erasing are disabled unless the required key code is written in FKEY. Different key codes are used for downloading and for programming/erasing.	○	○
Emulation protection	•	The program/erase-protected state is entered by setting the RAMS bit in the RAM emulation register (RAMER).	○	○

20.5.3 Error Protection

Error protection is a mechanism for aborting programming or erasure when an error occ
form of the microcomputer entering runaway during programming/erasing of the flash m
operations that are not according to the established procedures for programming/erasing
programming or erasure in such cases prevents damage to the flash memory due to exce
programming or erasing.

If the microcomputer malfunctions during programming/erasing of the flash memory, th
bit in the FCCS register is set to 1 and the error-protection state is entered, and this abor
programming or erasure.

reset should only be released after providing a reset input period of 100 µs, which is long usual. Since high voltages are applied during programming/erasing of the flash memory, voltage may remain after the error-protection state has been entered. For this reason, it is necessary to reduce the risk of damage to the flash memory by extending the reset period the charge is released.

The state-transition diagram in figure 20.16 shows transitions to and from the error-protec state.

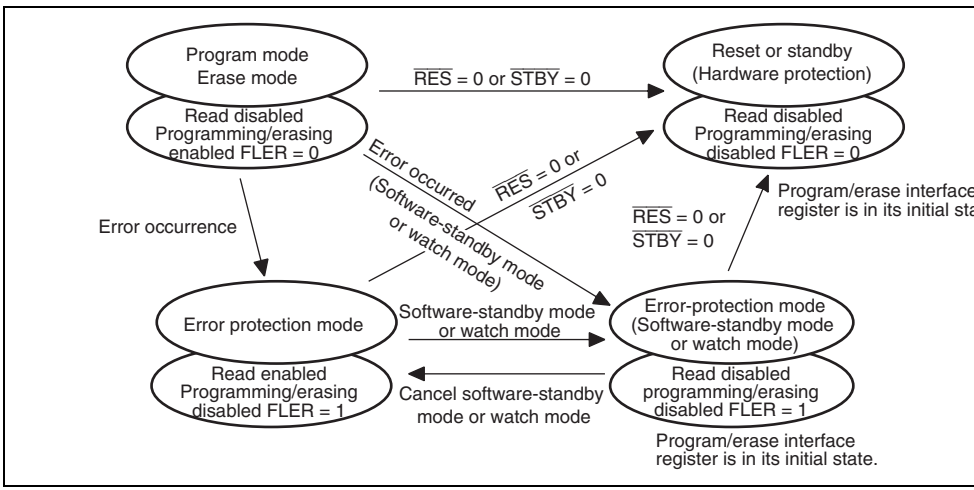


Figure 20.16 Transitions to Error-Protection State

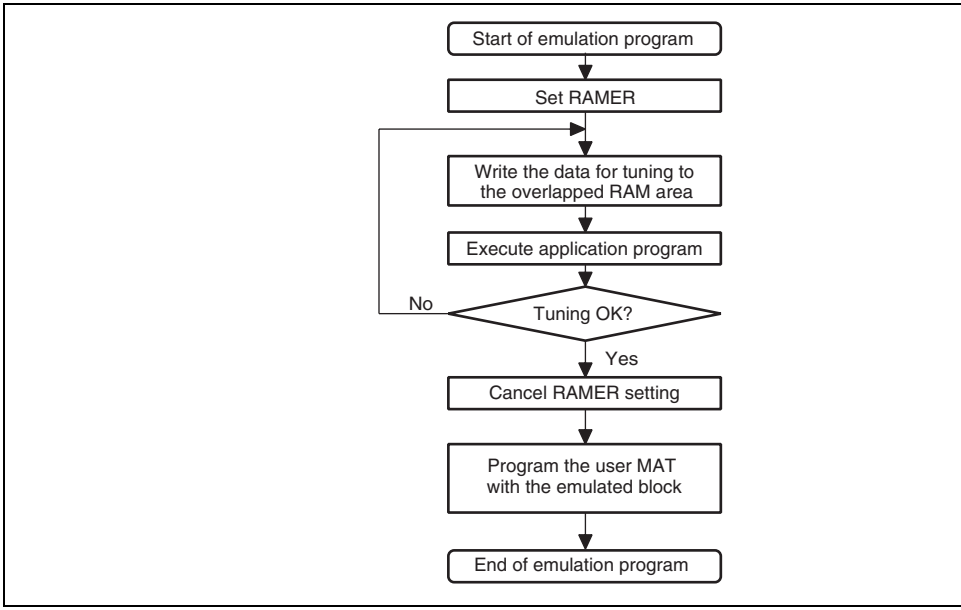


Figure 20.17 Emulation of Flash Memory in RAM

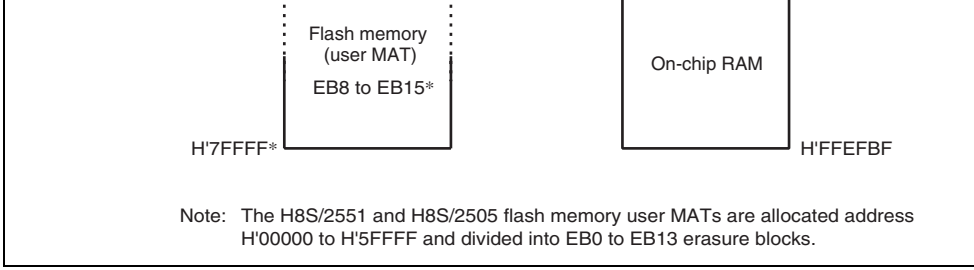


Figure 20.18 Example of a RAM-Overlap Operation

Figure 20.18 shows an example of an overlap on block area EB0 of the flash memory.

Emulation is possible for a single area selected from among the eight areas, from EB0 to
user MAT bank 0. The area is selected by the setting of the RAM2 to RAM0 bits in the R

1. To overlap a part of the RAM on area EB0, to allow real-time programming of the da
area, set the RAMS bit in RAMER to 1, and each of the RAM2 to RAM0 bits to 0.
2. Real-time programming is carried out using the overlapped area of RAM.

In programming or erasing the user MAT, it is necessary to run a program that implemen
of procedural steps, including the downloading of an on-chip program. In this process, th
RAM area and the area where the on-chip program is to be downloaded overlap. Therefo
data that is to be programmed must be saved beforehand in an area that is not used by the

Figure 20.19 shows an example of programming of the data, after emulation has been co
to the EB0 area in the user MAT.

Note: The H8S/2551 and H8S/2505 flash memory user MATs are allocated address H'00000 to H'5FFFF and divided into EB0 to EB13 erasure blocks.

Figure 20.19 Programming of the Data after Tuning

1. After the data to be programmed has fixed values, clear the RAMS bit to cancel the RAM.
2. Move the fixed programmed data in the overlap area in the RAM to the area to whic programming/erasing program created by the user is transferred and outside the area on-chip programs are downloaded.
3. Transfer the user-created programming/erasing-procedure program to the RAM.
4. Run the programming/erasing-procedure program on the RAM and download the on programming/erasing program.
5. When the EB0 area of the user MAT has not been erased, the programming program downloaded after erasing. Set the parameters FMPAR and FMPDR in the data to be programmed so that the tuned data that has been saved is designated, and execute programming.

Note: Setting the RAMS bit to 1 puts all the blocks in the flash MAT into a program/e protected state regardless of the values of the RAM2 to RAM0 bits (emulation p In this state, downloading of the on-chip programs is also disabled, so clear the before actual programming or erasure.

access to the flash memory during MAT switching).

3. If an interrupt has occurred during switching, there is no guarantee of which memory being accessed. Always mask the maskable interrupts before switching between MAT addition, configure the system so that NMI interrupts do not occur during MAT switc
4. After the MATs have been switched, take care because the interrupt vector table will been switched. If interrupt processing is to be the same before and after MAT switchi transfer the interrupt-processing routines to the on-chip RAM, and use the settings of and FVADR to place the interrupt-vector table in the on-chip RAM.
5. Memory sizes of the user MAT and user boot MAT are different. When accessing the boot MAT, do not access addresses above the top of its 8-kbyte memory space. If acc beyond the 8-kbyte space, the values read are undefined.

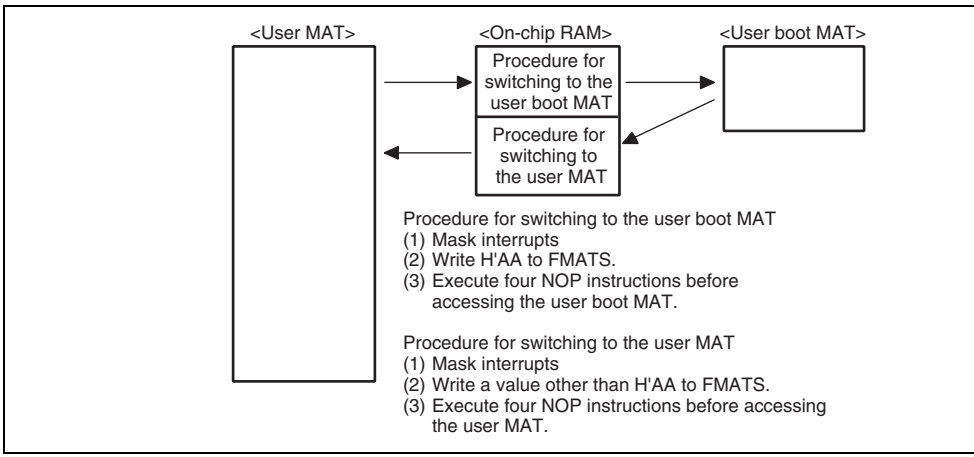


Figure 20.20 Switching between the User MAT and User Boot MAT

sure that these registers are not accidentally written to, otherwise an on-chip program downloaded and damage RAM or a MAT switchover may occur and the CPU may g control. Do not use DTC to program FLASH-related registers.

3. Compatibility with programming/erasing program of conventional F-ZTAT H8S microcomputer

   A programming/erasing program for flash memory used in the conventional F-ZTAT microcomputer which does not support download of the on-chip program by an SCC request cannot run in this LSI. Be sure to download the on-chip program to execute programming/erasing of flash memory in this LSI.

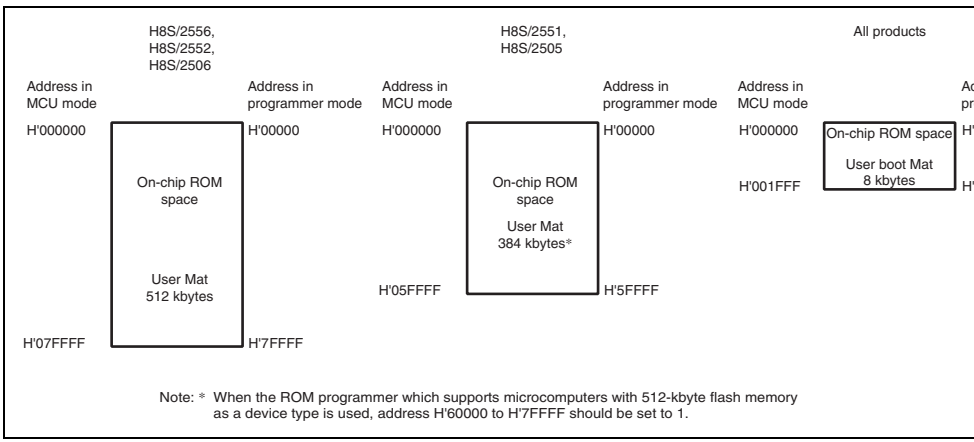
4. Monitoring runaway by WDT

   Unlike the conventional F-ZTAT H8S microcomputer, no countermeasures are avail runaway by WDT during programming/erasing by the downloaded on-chip program countermeasures (e.g. use of the user branch routine or the periodic timer interrupts) while taking the programming/erasing time into consideration as required.

MHz input-clock signal.

20.9.1 Pin Arrangement of Socket Adapter

Figure 20.21 and figure 20.22 show on-chip ROM memory map and socket adapter corre
map respectively. Attach the socket adapter to the LSI in the way shown in figure 20.22.
result, conversion to 40 pins is allowed.

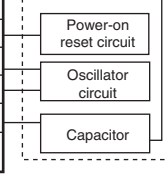


Note: * When the ROM programmer which supports microcomputers with 512-kbyte flash memory as a device type is used, address H'60000 to H'7FFFF should be set to 1.

Figure 20.21 On-Chip Flash Memory Map

24	K2	A10
25	L3	A11
26	L1	A12
27	L2	A13
28	L4	A14
29	M1	A15
30	M2	A16
31	M3	A17
32	N1	A18
4	C2	D0
5	D3	D1
6	C1	D2
7	D2	D3
8	E4	D4
9	D1	D5
10	E3	D6
11	E2	D7
3	B1	$\overline{CE}$
1	A1	$\overline{OE}$
2	C3	$\overline{WE}$
14, 72, 73, 77, 80, 84, 91, 96, 99, 118, 143	C9, C10, D9, D10, E14, F3, F14, G3, G4, H3, H4, H12, K12, M12, M15, N13, N15, P13, P14, R13, R14, R15, A2	VCC
12, 54, 67, 86, 87, 90, 92, 94, 97, 98, 110, 112, 117, 144	A10, A11, A12, A13, A14, B2, B10, C12, C13, E1, E12, E15, F1, F2, F12, F13, G1, G2, G12, G13, G14, H13, H14, H15, J12, J14, K14, K15, P7, P8, R6, R7, R11, R12	VSS
100	E13	$\overline{RES}$
93	G15	XTAL
95	F15	EXTAL
88	J13	VCL
Other than above		NC (OPEN)

32	A10
33	A11
34	A12
35	A13
36	A14
37	A15
38	A16
39	A17
10	A18
19	I/O0
18	I/O1
17	I/O2
16	I/O3
15	I/O4
14	I/O5
13	I/O6
12	I/O7
2	$\overline{CE}$
20	$\overline{OE}$
3	$\overline{WE}$
4	FWE
1, 40	V_{CC}
11, 30	V_{SS}
5, 6, 7	NC
8	A20
9	A19



Legend:
FWE: Flash write
I/O 7 to I/O0: Data I/O
A18 to A0: Address inp
$\overline{CE}$: Chip enable
$\overline{OE}$: Output enal
$\overline{WE}$: Write enabl

Note: * Available only in the H8S/2552 Group and the H8S/2506 Group.

Figure 20.22 Pin arrangement of Socket Adapter

and user boot MAT. Status polling is used to confirm the completion of automatic er

- Status-read mode: Status polling is used with automatic programming/automatic eras
  Normal completion can be confirmed by reading the signal on the I/O 6. In status-rea
  error information is output when an error has occurred.

Table 20.12 Setting Procedure of each Operation Mode of Programmer Mode

	Pin Name				
Mode	**$\overline{CE}$**	**$\overline{OE}$**	**WE**	**I/O 7 to 0**	**A 1**
Read	L	L	H	Data output	Ain
Output disable	L	H	H	Hi-Z	X
Command write	L	H	L	Data output	Ain
Chip disable	H	X	X	Hi-Z	X

Notes: Chip disable mode is not a standby state; internally, operating state.
* Ain indicates that address may be input in auto-program mode.

Auto-erase mode	2	User MAT	write	X	H'20	write	X
		User boot MAT	write	X	H'25		
Status-read mode	2	Common to both MAT	write	X	H'71	write	X

Notes: 1. In automatic programming mode, 129 cycles of command programming are because of simultaneous 128-byte programming.

2. In memory-read mode, the number of cycles varies according to the number address writing cycles (n).

20.9.3 Memory-Read Mode

1. On completion of an automatic program, automatic erase, or status read, the LSI en command waiting state. To read the contents of memory after these operations, issu command to change the mode to memory-read mode before reading from the memo

2. In memory-read mode, command programming can be performed as in the case of waiting state.

3. Continuous read can be performed after the transition to memory-read mode is mad

4. Transition to the memory-read mode is made after power has been supplied.
For the AC characteristics in memory-read mode, see section 20.11, AC Characteris Timing in Programmer Mode.

the third cycle or later.

5. Do not write commands while programming is in progress.
6. One time automatic programming should be performed for each 128-byte block of ad Additional programming of the block to the address where already programmed is no possible.
7. To confirm the end of automatic programming, check the signal on I/O6 pin. Confirm status-read mode is also possible. (Status polling of the I/O7 pin is used to check the status of automatic programming.)
8. Information on the pins I/O6 and I/O7 is retained until the next command is written. as no command is written, the information can be read by enabling the $\overline{CE}$ and $\overline{OE}$.

For details on the AC characteristics in auto-program mode, see section 20.11, AC Chara and Timing in Programmer Mode.

20.9.5 Auto-Erase Mode

1. In auto-erase mode, only erasing the entire memory is supported.
2. Command writing should not be preformed during automatic erasing.
3. To confirm the end of automatic erasing, check the signal on the I/O6 pin. Confirmat status read mode is also possible. (Status polling of the I/O7 pin is used to check the status of automatic erasure.)
4. Information on the pins I/O6 and I/O7 is retained until the next command is written. as other command is written, the information can be read by enabling the $\overline{CE}$ and $\overline{OE}$

For details on AC characteristics in auto-erase mode, see section 20.11, AC Characteristi Timing in Programmer Mode.

Table 20.14 Return Codes in Status-Read Mode

Pin Name	I/O7	I/O6	I/O5	I/O4	I/O3	I/O2	I/O1
Attribute	Normal end indicator	Command error	Program-ming error	Erase error	—	—	Programming or erasure count exceed
Initial value	0	0	0	0	0	0	0
Descrip-tion	Normal end: 0 Abnormal end: 1	Command error: 1 Other: 0	Program-ming error: 1 Other: 0	Erase error: 1 Other: 0	—	—	Count exceeded: 1 Other: 0

Note: I/O2 and I/O3 are undefined.

20.9.7 Status Polling

1. The I/O7 status-polling output is a flag that indicates the operating status in auto-pr auto-erase mode.
2. The I/O6 status-polling output is a flag that indicates normal/abnormal end in auto-p auto-erase mode.

Table 20.15 True Value Table of Status Polling Output

Pin Name	In Progress	Abnormal End	—	Normal
I/O7	0	1	0	1
I/O6	0	0	1	1
I/O0 to I/O5	0	0	0	0

programming mode with a programmer, it is recommended performing automatic programming after automatic erase.

3. Do not remove the chip from the PROM programmer nor input a reset signal during programming/erasing. As a high voltage is applied to the flash memory during programming/erasing, doing so may damage flash memory permanently. If a reset is accidentally, the reset must be released after a reset period of 100 μs, which is longer usual.
4. The initial state of a Renesas Technology product at shipment is the erased state. For product whose history of erasing is undefined, automatic erasure for checking the ini (erased state) and compensating is recommended.
5. In this LSI, production identification mode such as multipurpose EPROM is not supp device name can not be automatically set to PROM writer.
6. For the PROM programmer suitable for programmer mode in this LSI and its progra version, refer to the instruction manual of the socket adapter.

mode enables starting of the boot program and entry to the bit-rate-adjustment state. program receives the command from the host to adjust the bit rate. After adjusting th the program enters the inquiry/selection state.

2. Inquiry/Selection State

   In this state, the boot program responds to inquiry commands from the host. The dev clock mode, and bit rate are selected. After selection of these settings, the program is enter the programming/erasing state by the command for a transition to the programming/erasing state. The program transfers the libraries required for erasure t RAM and erases the user MATs and user boot MATs before the transition.

3. Programming/erasing state

   Programming and erasure by the boot program take place in this state. The boot prog made to transfer the programming/erasing programs to the RAM by commands from Sum checks and blank checks are executed by sending these commands from the hos

These boot program states are shown in figure 20.23.

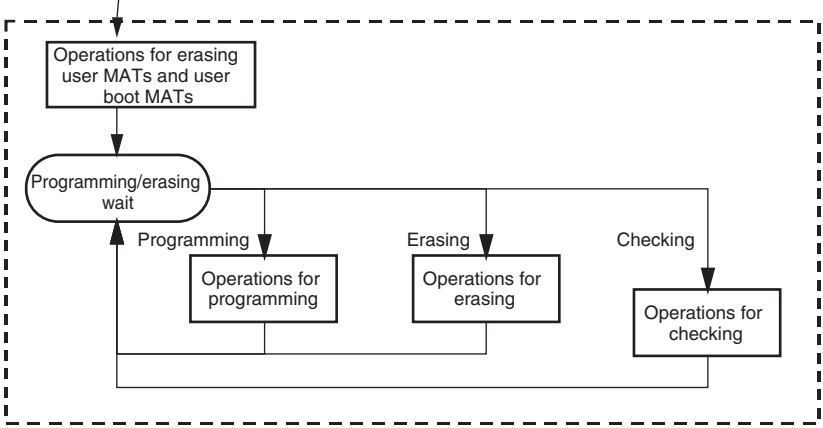


Figure 20.23 Boot Program States

(2) Bit-Rate-Adjustment State

The bit rate is calculated by measuring the period of transfer of a low-level byte (H'00) fr
host. The bit rate can be changed by the command for a new bit rate selection. After the b
has been adjusted, the boot program enters the inquiry and selection state. The bit-rate-ad
sequence is shown in figure 20.24.

(3) Communications Protocol

After adjustment of the bit rate, the protocol for communications between the host and t
program is as shown below.

1. One-byte commands and one-byte responses

   These commands and responses are comprised of a single byte. These are consists of
   inquiries and the ACK for successful completion.
2. n-byte commands or n-byte responses

   These commands and responses are comprised of n bytes of data. These are selection
   responses to inquiries.

   The amount of programming data is not included under this heading because it is det
   in another command.
3. Error response

   The error response is a response to inquiries that consists of an error response and an
   code. The response comes in two bytes.
4. Programming of 128 bytes

   The size is not specified in commands. The size of n is indicated in response to the
   programming unit inquiry.
5. Memory read response

   This response consists of four bytes of data.

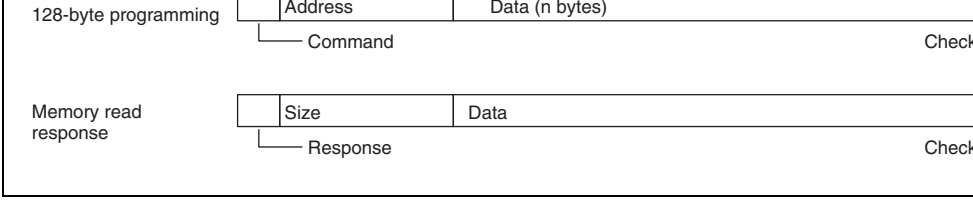


Figure 20.25 Communication Protocol Format

- Command (one byte): Commands including inquiries, selection, programming, erasin checking
- Response (one byte): Response to an inquiry
- Size (one byte): The amount of data for transmission excluding the command, amoun and checksum
- Checksum (one byte): The checksum is calculated so that the total of all values from t command byte to the SUM byte becomes H'00.
- Data (n bytes): Detailed data of a command or response
- Error response (one byte): Error response to a command
- Error code (one byte): Type of the error
- Address (four bytes): Address for programming
- Data (n bytes): Data to be programmed (the size is indicated in the response to the programming unit inquiry.)
- Size (four bytes): Four-byte response to a memory read

(4) Inquiry and Selection States

The boot program returns information from the flash memory in response to the host's in commands and sets the device code, clock mode, and bit rate in response to the host's sel command.

		multiplied clock types, the number of multi ratios, and the values of each multiple
H'23	Operating Clock Frequency Inquiry	Inquiry regarding the maximum and minim values of the main clock and peripheral cl
H'24	User Boot MAT Information Inquiry	Inquiry regarding the number of user boot and the start and last addresses of each M
H'25	User MAT Information Inquiry	Inquiry regarding the a number of user MA the start and last addresses of each MAT
H'26	Block for Erasing Information Inquiry	Inquiry regarding the number of blocks an and last addresses of each block
H'27	Programming Unit Inquiry	Inquiry regarding the unit of programming
H'3F	New Bit Rate Selection	Selection of new bit rate
H'40	Transition to Programming/Erasing State	Erasing of user MAT and user boot MAT, to programming/erasing state
H'4F	Boot Program Status Inquiry	Inquiry into the operated status of the boo

The selection commands, which are device selection (H'10), clock mode selection (H'11 bit rate selection (H'3F), should be sent from the host in that order. These commands wi be needed. When two or more selection commands are sent at once, the last command w valid.

All of these commands, except for the boot program status inquiry command (H'4F), wi until the boot program receives the programming/erasing transition (H'40). The host can the needed commands out of the commands and inquiries listed above. The boot progra inquiry command (H'4F) is valid after the boot program has received the programming/e transition command (H'40).

...			
SUM			

- Response, H'30, (one byte): Response to the supported device inquiry
- Size (one byte): Number of bytes to be transmitted, excluding the command, size, and checksum, that is, the amount of data contributes by the number of devices, characters codes and product names
- Number of devices (one byte): The number of device types supported by the boot pro
- Number of characters (one byte): The number of characters in the device codes and bo program's name
- Device code (four bytes): ASCII code of the supporting product
- Product name (n bytes): Type name of the boot program in ASCII-coded characters
- SUM (one byte): Checksum

  The checksum is calculated so that the total number of all values from the command b the SUM byte becomes H'00.

(b) Device Selection

The boot program will set the supported device to the specified device code. The program return the selected device code in response to the inquiry after this setting has been made.

Command	H'10	Size	Device code	SUM

- Command, H'10, (one byte): Device selection
- Size (one byte): Amount of device-code data
  This is fixed at 4
- Device code (four bytes): Device code (ASCII code) returned in response to the suppo device inquiry

H'21: Device code error, that is, the device code does not match

(c) Clock Mode Inquiry

The boot program will return the supported clock modes in response to the clock mode i

Command	H'21

- Command, H'21, (one byte): Inquiry regarding clock mode

Response	H'31	Size	Number of modes	Mode	···	SUM

- Response, H'31, (one byte): Response to the clock-mode inquiry
- Size (one byte): Amount of data that represents the number of modes and modes
- Number of clock modes (one byte): The number of supported clock modes
  H'00 indicates no clock mode or the device allows to read the clock mode.
- Mode (one byte): Values of the supported clock modes (i.e. H'01 means clock mode
- SUM (one byte): Checksum

(d) Clock Mode Selection

The boot program will set the specified clock mode. The program will return the selecte
mode information after this setting has been made.

The clock-mode selection command should be sent after the device-selection commands

Command	H'11	Size	Mode	SUM

- Command, H'11, (one byte): Selection of clock mode
- Size (one byte): Amount of data that represents the modes
- Mode (one byte): A clock mode returned in reply to the supported clock mode inquir
- SUM (one byte): Checksum

be selected using these respective values.

(e) Multiplication Ratio Inquiry

The boot program will return the supported multiplication and division ratios.

Command	H'22

- Command, H'22, (one byte): Inquiry regarding multiplication ratio

Response	H'32	Size	Number of types				
	Number of multiplication ratios	Multiplica-tion ratio	···				
	···						
	SUM						

- Response, H'32, (one byte): Response to the multiplication ratio inquiry
- Size (one byte): The amount of data that represents the number of clock sources and multiplication ratios and the multiplication ratios
- Number of types (one byte): The number of supported multiplied clock types
  (e.g. when there are two multiplied clock types, which are the main and peripheral clo number of types will be H'02.)
- Number of multiplication ratios (one byte): The number of multiplication ratios for ea (e.g. the number of multiplication ratios to which the main clock can be set and the pe clock can be set.)
- Multiplication ratio (one byte)

  Multiplication ratio: The value of the multiplication ratio (e.g. when the clock-frequen multiplier is four, the value of multiplication ratio will be H'04.)

Command	H'23

- Command, H'23, (one byte): Inquiry regarding operating clock frequencies

Response	H'33	Size	Number of operating clock frequencies
	Minimum value of operating clock frequency		Maximum value of operating clock frequency
	...		
	SUM		

- Response, H'33, (one byte): Response to operating clock frequency inquiry
- Size (one byte): The number of bytes that represents the minimum values, maximum and the number of frequencies.
- Number of operating clock frequencies (one byte): The number of supported operati frequency types
  (e.g. when there are two operating clock frequency types, which are the main and pe clocks, the number of types will be H'02.)
- Minimum value of operating clock frequency (two bytes): The minimum value of th multiplied or divided clock frequency.

  The minimum and maximum values represent the values in MHz, valid to the hundr of MHz, and multiplied by 100. (e.g. when the value is 20.00 MHz, it will be D'2000 H'07D0.)
- Maximum value (two bytes): Maximum value of the multiplied or divided clock fre

  There are as many pairs of minimum and maximum values as there are operating clo frequencies.
- SUM (one byte): Checksum

- Response, H'34, (one byte): Response to user boot MAT information inquiry
- Size (one byte): The number of bytes that represents the number of areas, area-start ad and area-last address
- Number of Areas (one byte): The number of consecutive user boot MAT areas
  When user boot MAT areas are consecutive, the number of areas returned is H'01.
- Area-start address (four bytes): Start address of the area
- Area-last address (four bytes): Last address of the area
  There are as many groups of data representing the start and last addresses as there are
- SUM (one byte): Checksum

(h) User MAT Information Inquiry

The boot program will return the number of user MATs and their addresses.

Command	H'25

- Command, H'25, (one byte): Inquiry regarding user MAT information

Response	H'35	Size	Number of areas	
	Start address area			Last address area
	...			
	SUM			

- Response, H'35, (one byte): Response to the user MAT information inquiry
- Size (one byte): The number of bytes that represents the number of areas, area-start ad and area-last address
- Number of areas (one byte): The number of consecutive user MAT areas
  When the user MAT areas are consecutive, the number of areas is H'01.
- Area-start address (four bytes): Start address of the area

Response	H'36	Size	Number of blocks	
	Block start address			Block last address
	...			
	SUM			

- Response, H'36, (one byte): Response to the number of erased blocks and addresses
- Size (two bytes): The number of bytes that represents the number of blocks, block-st addresses, and block-last addresses.
- Number of blocks (one byte): The number of erased blocks
- Block start address (four bytes): Start address of a block
- Block last Address (four bytes): Last address of a block
  There are as many groups of data representing the start and last addresses as there ar
- SUM (one byte): Checksum

(j) Programming Unit Inquiry

The boot program will return the programming unit used to program data.

Command	H'27			

- Command, H'27, (one byte): Inquiry regarding programming unit

Response	H'37	Size	Programming unit	SUM

- Response, H'37, (one byte): Response to programming unit inquiry
- Size (one byte): The number of bytes that indicate the programming unit, which is fi
- Programming unit (two bytes): A unit for programming
  This is the unit for reception of programming.
- SUM (one byte): Checksum

- Size (one byte): The number of bytes that represents the bit rate, input frequency, nu multiplication ratios, and multiplication ratio
- Bit rate (two bytes): New bit rate
  One hundredth of the value (e.g. when the value is 19200 bps, the bit rate is H'00C0, D'192.)
- Input frequency (two bytes): Frequency of the clock input to the boot program
  This is valid to the hundredths place and represents the value in MHz multiplied by 1 when the value is 20.00 MHz, the input frequency is H'07D0 (= D'2000).)
- Number of multiplication ratios (one byte): The number of multiplication ratios to wh device can be set. Normally, 2 is set because multiplication ratios for the main-operati peripheral frequencies are set. (In this LSI, set H'01.)
- Multiplication ratio 1 (one byte): The value of multiplication or division ratios for the operating frequency
  Multiplication ratio (one byte): The value of the multiplication ratio (e.g. when the clo frequency is multiplied by four, the multiplication ratio will be H'04. In this LSI, set H
  Division ratio: The inverse of the division ratio, as a negative number (e.g. when the c frequency is divided by two, the value of division ratio will be H'FE. H'FE = D'-2. In set H'01.)
- Multiplication ratio 2 (one byte): The value of multiplication or division ratios for the peripheral frequency
  Multiplication ratio (one byte): The value of the multiplication ratio (e.g. when the clo frequency is multiplied by four, the multiplication ratio will be H'04. In this LSI, this not need to be set.)
  (Division ratio: The inverse of the division ratio, as a negative number (E.g. when the divided by two, the value of division ratio will be H'FE. H'FE = D'-2 In this LSI, set H
- SUM (one byte): Checksum

H'25: Error in input frequency
This input frequency is not within the specified range.
H'26: Multiplication-ratio error
The ratio does not match an available ratio.
H'27: Operating frequency error
The frequency is not within the specified range.

(5) Received Data Check

The methods for checking of received data are listed below.

1. Input frequency

   The received value of the input frequency is checked to ensure that it is within the ra
   minimum to maximum frequencies which matches the clock modes of the specified
   When the value is out of this range, an input-frequency error is generated.
2. Multiplication ratio

   The received value of the multiplication ratio or division ratio is checked to ensure th
   matches the clock modes of the specified device. If the value does not match the mul
   ratio or division ratio, it is the multiplication ratio error.
3. Operating frequency error

   Operating frequency is calculated from the received value of the input frequency and
   multiplication or division ratio. The input frequency is input to the LSI and the LSI i
   at the operating frequency. The expression is given below.

   Operating frequency = Input frequency × Multiplication ratio, or
   Operating frequency = Input frequency ÷ Division ratio

   The calculated operating frequency should be checked to ensure that it is within the
   minimum to maximum frequencies which are available with the clock modes of the
   device. When it is out of this range, an operating frequency error is generated.

response. The host will send an ACK with the new bit rate for confirmation and the boot will response with that rate.

Confirmation	H'06

- Confirmation, H'06, (one byte): Confirmation of a new bit rate

Response	H'06

- Response, H'06, (one byte): Response to confirmation of a new bit rate

The sequence of new bit-rate selection is shown in figure 20.26.

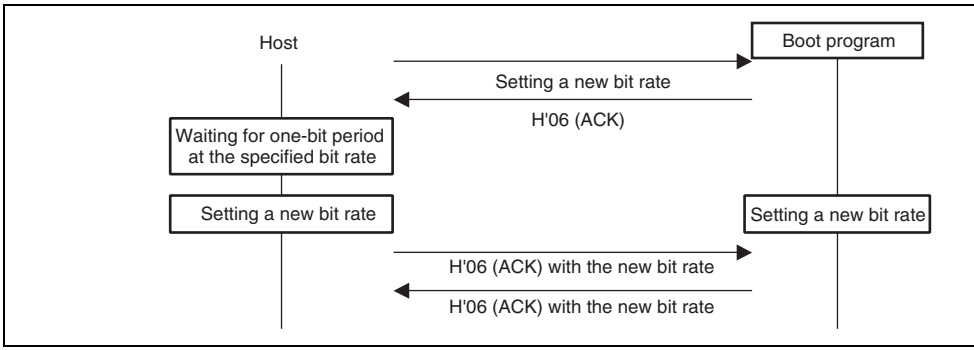


Figure 20.26 New Bit-Rate Selection Sequence

(6) Transition to Programming/Erasing State

The boot program will transfer the erasing program, and erase the user MATs and user bo in that order. On completion of this erasure, ACK will be returned and will enter the programming/erasing state.

The boot program will send ACK when the user MAT and user boot MAT have beer
by the transferred erasing program.

Error Response	H'C0	H'51

- Error response, H'C0, (one byte): Error response for user boot MAT blank check
- Error code, H'51, (one byte): Erasing error
  An error occurred and erasure was not completed.

(7) Command Error

A command error will occur when a command is undefined, the order of commands is ir
or a command is unacceptable. Issuing a clock-mode selection command before a device
or an inquiry command after the transition to programming/erasing state command, are

Error Response	H'80	H'xx

- Error response, H'80, (one byte): Command error
- Command, H'xx, (one byte): Received command

(8) Command Order

The order for commands in the inquiry selection state is shown below.

1. A supported device inquiry (H'20) should be made to inquire about the supported de
2. The device should be selected from among those described by the returned informati
   with a device-selection (H'10) command.
3. A clock-mode inquiry (H'21) should be made to inquire about the supported clock m
4. The clock mode should be selected from among those described by the returned info
   and set.

programming/erasing state command (H'40). The boot program will then enter the programming/erasing state.

(9) Programming/Erasing State

A programming selection command makes the boot program select the programming met 128-byte programming command makes it program the memory with data, and an erasing selection command and block erasing command make it erase the block. The programming/erasing commands are listed below.

Table 20.17 Programming/Erasing Command

Command	Command Name	Description
H'42	User boot MAT programming selection	Transfers the user boot MAT programm program
H'43	User MAT programming selection	Transfers the user MAT programming p
H'50	128-byte programming	Programs 128 bytes of data
H'48	Erasing selection	Transfers the erasing program
H'58	Block erasing	Erases a block of data
H'52	Memory read	Reads the contents of memory
H'4A	User boot MAT sum check	Checks the checksum of the user boot
H'4B	User MAT sum check	Checks the checksum of the user MAT
H'4C	User boot MAT blank check	Checks whether the contents of the use MAT are blank
H'4D	User MAT blank check	Checks whether the contents of the use are blank
H'4F	Boot program status inquiry	Inquires into the boot program's status

After issuing the programming selection command, the host should send the 128-byte programming command. The 128-byte programming command that follows the selectio command represents the data programmed according to the method specified by the sele command. When more than 128-byte data is programmed, 128-byte commands should n be executed. Sending a 128-byte programming command with H'FFFFFFFF as the addr stop the programming. On completion of programming, the boot program will wait for s programming or erasing.

Where the sequence of programming operations that is executed includes programming another method or of another MAT, the procedure must be repeated from the programm selection command.

The sequence for programming-selection and 128-byte programming commands is shov figure 20.27.

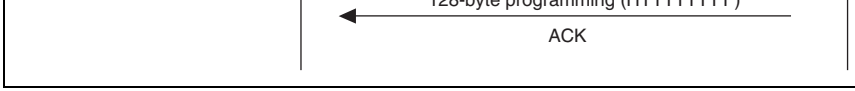


Figure 20.27 Programming Sequence

(a) User boot MAT programming selection

The boot program will transfer a programming program. The data is programmed to the u
MATs by the transferred programming program.

Command	H'42

- Command, H'42, (one byte): User boot-program programming selection

Response	H'06

- Response, H'06, (one byte): Response to user boot-program programming selection
  When the programming program has been transferred, the boot program will return A

Error Response	H'C2	ERROR

- Error response : H'C2 (1 byte): Error response to user boot MAT programming selecti
- ERROR : (1 byte): Error code
  H'54: Selection processing error (transfer error occurs and processing is not complete

(b) User-program programming selection

The boot program will transfer a program for programming. The data is programmed to t
MATs by the transferred program for programming.

Command	H'43

- Command, H'43, (one byte): User-program programming selection

The boot program will use the programming program transferred by the programming s
program the user boot MATs or user MATs in response to 128-byte programming.

Command	H'50	Address						
	Data	···						
	···							
	SUM							

- Command, H'50, (one byte): 128-byte programming
- Programming Address (four bytes): Start address for programming
  Multiple of the size specified in response to the programming unit inquiry
  (i.e. H'00, H'01, H'00, H'00 : H'00010000)
- Programming Data (128 bytes): Data to be programmed
  The size is specified in the response to the programming unit inquiry.
- SUM (one byte): Checksum

Response	H'06

- Response, H'06, (one byte): Response to 128-byte programming
  On completion of programming, the boot program will return ACK.

Error Response	H'D0	ERROR

- Error response, H'D0, (one byte): Error response for 128-byte programming
- ERROR: (one byte): Error code

  H'11: Checksum Error

  H'28: Address error
  Address is not in the specified MAT.

  H'53: Programming error
  A programming error has occurred and programming cannot be continued.

- Programming Address (four bytes): End code is H'FF, H'FF, H'FF, H'FF.
- SUM (one byte): Checksum

Response	H'06

- Response, H'06, (one byte): Response to 128-byte programming
  On completion of programming, the boot program will return ACK.

Error Response	H'D0	ERROR

- Error Response, H'D0, (one byte): Error response for 128-byte programming
- ERROR: (one byte): Error code
  H'11: Sum check error
  H'53: Programming error
  An error has occurred in programming and programming cannot be continued

(11) Erasure

Erasure is performed with the erasure selection and block erasure command.

Firstly, erasure is selected by the erasure selection command and the boot program then e
specified block. The command should be repeatedly executed if two or more blocks are t
erased. Sending a block-erasure command from the host with the block number H'FF will
erasure operating. On completion of erasing, the boot program will wait for selection of
programming or erasing.

The sequences of the issuing of erasure selection commands and the erasure of data are sl
figure 20.28.

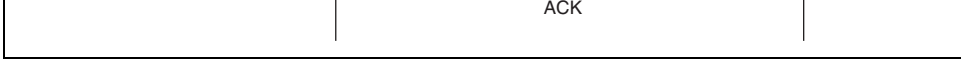


Figure 20.28 Erasure Sequence

(a) Erasure Selection

The boot program will transfer the erasure program. User MAT data is erased by the tra
erasure program.

Command	H'48

- Command, H'48, (one byte): Erasure selection

Response	H'06

- Response, H'06, (one byte): Response for erasure selection
  After the erasure program has been transferred, the boot program will return ACK.

Error Response	H'C8	ERROR

- Error Response, H'C8, (one byte): Error response to erasure selection
- ERROR: (one byte): Error code
  H'54: Selection processing error (transfer error occurs and processing is not compl

(b) Block Erasure

The boot program will erase the contents of the specified block.

Command	H'58	Size	Block number	SUM

- Command, H'58, (one byte): Erasure
- Size (one byte): The number of bytes that represents the erasure block number
  This is fixed to 1.
- Block number (one byte): Number of the block to be erased

Block number is incorrect.

H'51: Erasure error
An error has occurred during erasure.

On receiving block number H'FF, the boot program will stop erasure and wait for a select command.

Command	H'58	Size	Block number	SUM

- Command, H'58, (one byte): Erasure
- Size, (one byte): The number of bytes that represents the block number
  This is fixed to 1.
- Block number (one byte): H'FF
  Stop code for erasure
- SUM (one byte): Checksum

Response	H'06

- Response, H'06, (one byte): Response to end of erasure (ACK)
  When erasure is to be performed after the block number H'FF has been sent, the proce should be executed from the erasure selection command.

(12) Memory read

The boot program will return the data in the specified address.

Command	H'52	Size	Area	Read address	
	Read size				SUM

- Command: H'52 (1 byte): Memory read
- Size (1 byte): Amount of data that represents the area, read address, and read size (fix

- Response: H'52 (1 byte): Response to memory read
- Read size (4 bytes): Size of data to be read
- Data (n bytes): Data for the read size from the read address
- SUM (1 byte): Checksum

Error Response	H'D2	ERROR

- Error response: H'D2 (1 byte): Error response to memory read
- ERROR: (1 byte): Error code

  H'11: Sum check error

  H'2A: Address error

  The read address is not in the MAT.

  H'2B: Size error

  The read size exceeds the MAT.

(13) User-Boot Program Sum Check

The boot program will return the byte-by-byte total of the contents of the bytes of the us program, as a four-byte value.

Command	H'4A

- Command, H'4A, (one byte): Sum check for user-boot program

Response	H'5A	Size	Checksum of user boot program	SUM

- Response, H'5A, (one byte): Response to the sum check of user-boot program
- Size (one byte): The number of bytes that represents the checksum
  This is fixed to 4.
- Checksum of user boot program (four bytes): Checksum of user boot MATs
  The total of the data is obtained in byte units.

- Response, H'5B, (one byte): Response to the sum check of the user program
- Size (one byte): The number of bytes that represents the checksum
  This is fixed to 4.
- Checksum of user boot program (four bytes): Checksum of user MATs
  The total of the data is obtained in byte units.
- SUM (one byte): Sum check for data being transmitted

(15) User Boot MAT Blank Check

The boot program will check whether or not all user boot MATs are blank and return the

Command	H'4C

- Command, H'4C, (one byte): Blank check for user boot MAT

Response	H'06

- Response, H'06, (one byte): Response to the blank check of user boot MAT
  If all user MATs are blank (H'FF), the boot program will return ACK.

Error Response	H'CC	H'52

- Error Response, H'CC, (one byte): Response to blank check for user boot MAT
- Error Code, H'52, (one byte): Erasure has not been completed.

(16) User MAT Blank Check

The boot program will check whether or not all user MATs are blank and return the resul

Command	H'4D

- Command, H'4D, (one byte): Blank check for user MATs

be made in the inquiry/selection state or the programming/erasing state.

Command	H'4F

- Command, H'4F, (one byte): Inquiry regarding boot program's state

Response	H'5F	Size	Status	ERROR	SUM

- Response, H'5F, (one byte): Response to boot program state inquiry
- Size (one byte): The number of bytes. This is fixed to 2.
- Status (one byte): State of the boot program
- ERROR (one byte): Error state

  ERROR = 0 indicates normal operation.
  ERROR other than 0 indicates abnormal.

- SUM (one byte): Checksum
  This command can be accepted during programming/erasing operation, however, res
  time will be longer.

H'5F	Erasure Block Specification Wait (Erasure is completed)

Table 20.19 Error Code

Code	Description
H'00	No Error
H'11	Sum Check Error
H'12	Program Size Error
H'21	Device Code Mismatch Error
H'22	Clock Mode Mismatch Error
H'24	Bit Rate Selection Error
H'25	Input Frequency Error
H'26	Multiplication Ratio Error
H'27	Operating Frequency Error
H'29	Block Number Error
H'2A	Address Error
H'2B	Data Length Error
H'51	Erasure Error
H'52	Erasure Incomplete Error
H'53	Programming Error
H'54	Selection Processing Error
H'80	Command Error
H'FF	Bit-Rate-Adjustment Confirmation Error

Data setup time	t_{ds}	50	—	ns
Programming pulse width	t_{wep}	70	—	ns
$\overline{WE}$ rising time	t_r		30	ns
$\overline{WE}$ falling time	t_f		30	ns

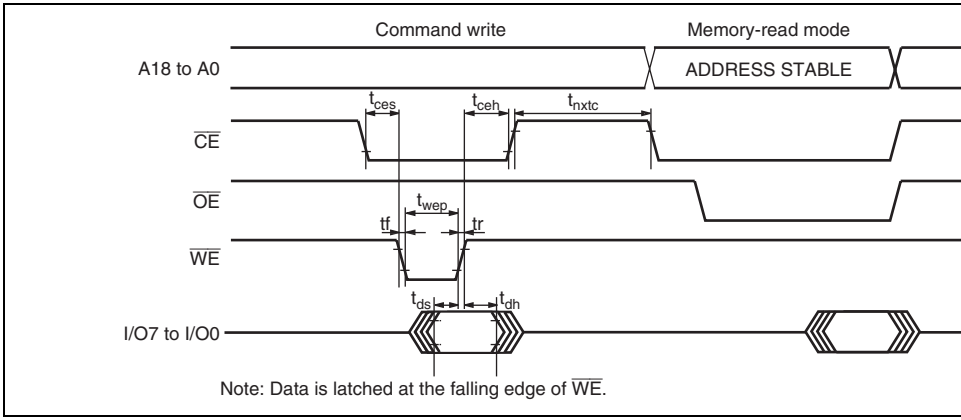


Figure 20.29 Memory Read Timing after Command Programming

Programming pulse width	t_{wep}	70	—	ns
$\overline{WE}$ rising time	t_r	—	30	ns
$\overline{WE}$ falling time	t_f	—	30	ns

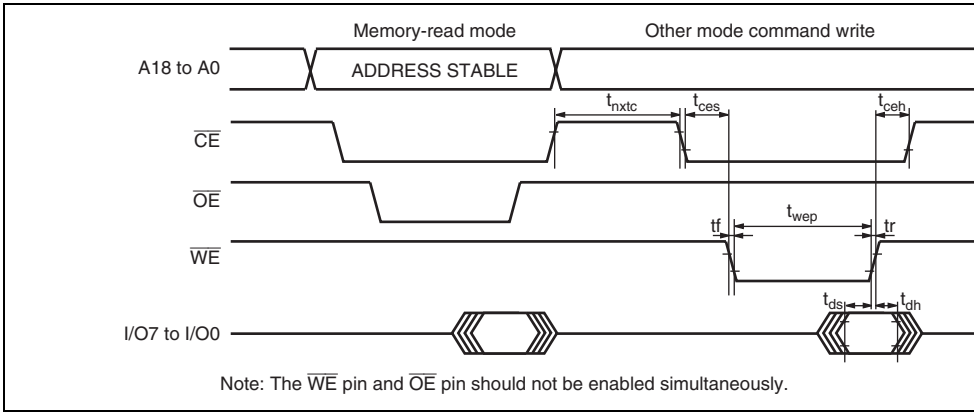


Figure 20.30 Waveform of Transition from Memory-Read Mode to Other M

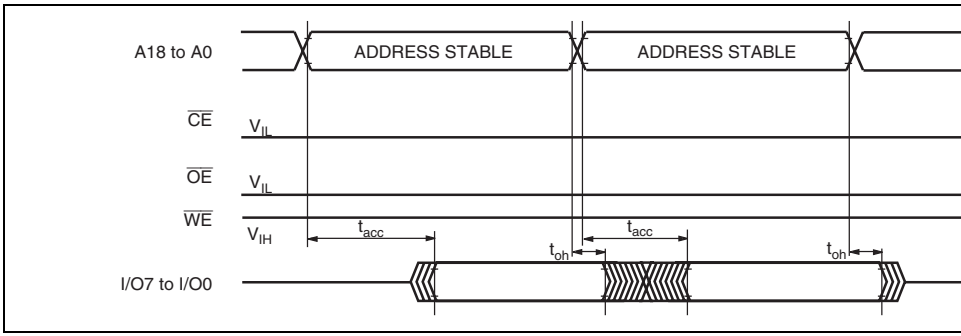


Figure 20.31 Waveform of $\overline{CE}$, $\overline{OE}$ Enable State Read

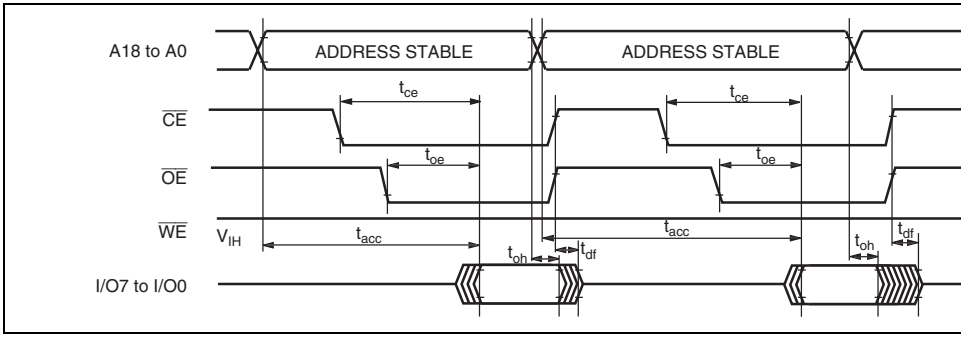


Figure 20.32 Waveform of $\overline{CE}$, $\overline{OE}$ Clock System Read

Programming pulse width	t_{wep}	70	—	ns
Status polling start time	t_{wsts}	1	—	ms
Status polling access time	t_{spa}	—	150	ns
Address setup time	t_{as}	0	—	ns
Address hold time	t_{ah}	60	—	ns
Memory programming time	t_{write}	1	3000	ms
$\overline{WE}$ rising time	t_r	—	30	ns
$\overline{WE}$ falling time	t_f	—	30	ns

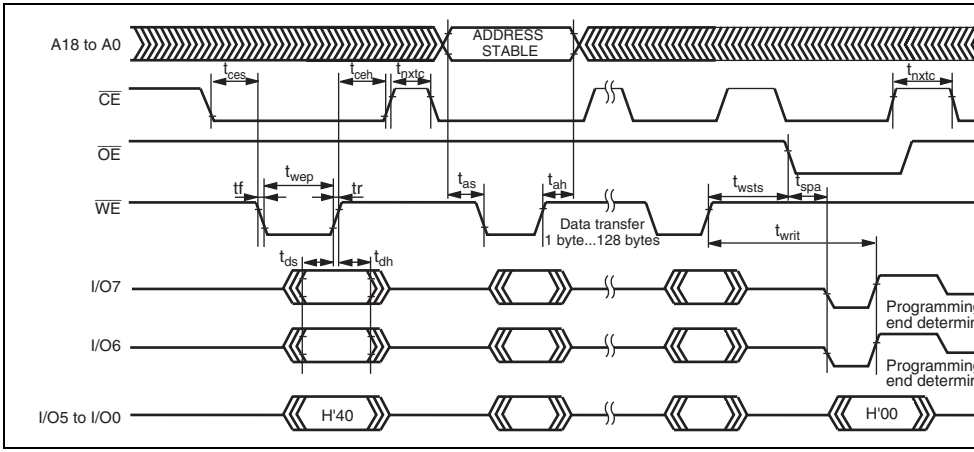


Figure 20.33 Waveform of Automatic Programming Mode

Programming pulse width	t_{wep}	70	—	ns
Status polling start time	T_{ests}	1	—	ms
Status polling access time	t_{spa}	—	150	ns
Memory erasing time	t_{erase}	100	40000	ms
$\overline{WE}$ rising time	t_r	—	30	ns
$\overline{WE}$ falling time	t_f	—	30	ns

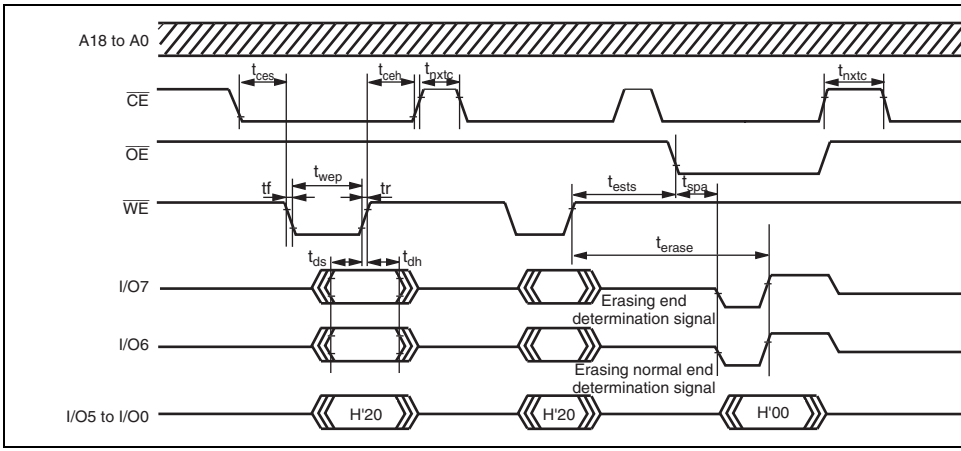


Figure 20.34 Waveform in Auto-Erase Mode

Programming pulse width	t_{wep}	70	—	ns
$\overline{OE}$ output delay time	t_{oe}	—	150	ns
Disable delay time	t_{df}	—	100	ns
$\overline{CE}$ output delay time	t_{ce}	—	150	ns
$\overline{WE}$ rising time	t_r	—	30	ns
$\overline{WE}$ falling time	t_f	—	30	ns

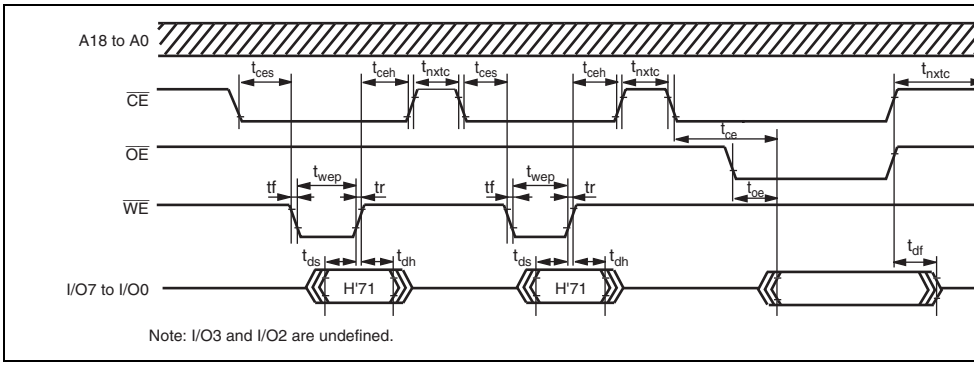


Figure 20.35 Waveform in Status-Read Mode

Figure 20.36 Oscillation Stabilized Time, Programmer Mode Setup Time and Falling Sequence

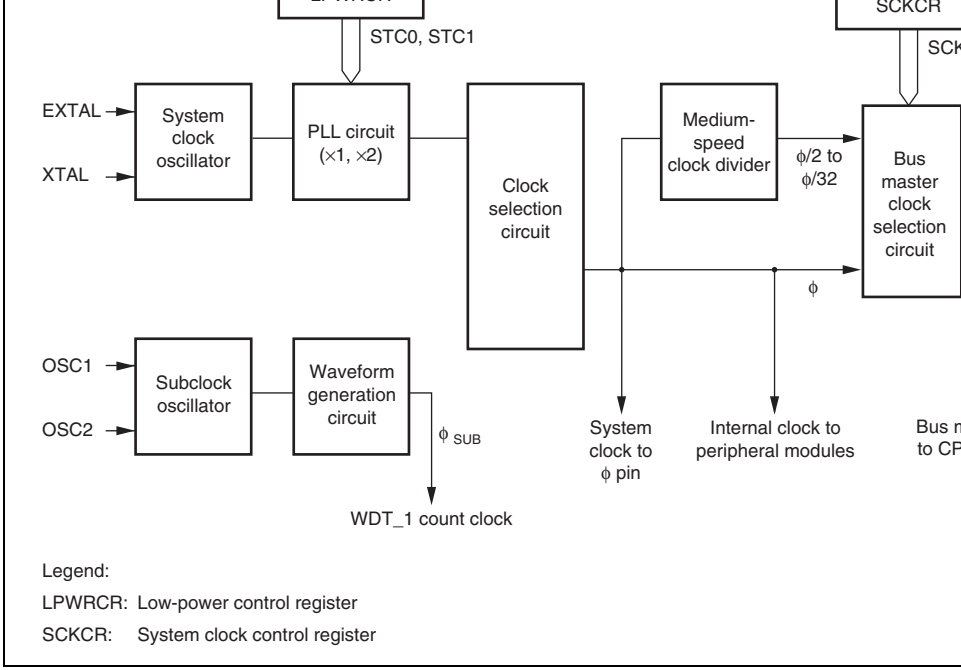


Figure 21.1 Block Diagram of Clock Pulse Generator

Frequency changes are performed by software by settings in the low-power control regis
(LPWRCR) and system clock control register (SCKCR).

of the PLL circuit, and medium-speed mode control.

Bit	Bit Name	Initial Value	R/W	Description
7	PSTOP	0	R/W	φ Clock Output Prohibited Controls φ output. • High-speed, medium-speed, and sleep modes 0: φ output 1: Fixed to high • Software standby and watch modes 0: Fixed to high 1: Fixed to high • Hardware standby mode 0: High impedance 1: High impedance
6	—	0	R/W	Reserved This is a readable/writable bit, but the write value s always be 0.
5, 4	—	All 0	—	Reserved These bits are always read as 0 and cannot be mo

1	SCK1	0	R/W	Select the bus master clock.
0	SCK0	0	R/W	000: High-speed mode 001: Medium-speed clock is ϕ/2 010: Medium-speed clock is ϕ/4 011: Medium-speed clock is ϕ/8 100: Medium-speed clock is ϕ/16 101: Medium-speed clock is ϕ/32 11X: Setting prohibited

Legend:
X: Don't care

21.1.2 Low-Power Control Register (LPWRCR)

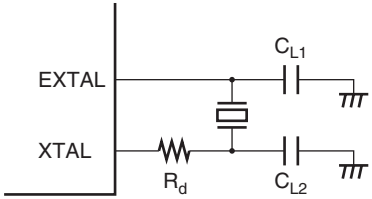
LPWRCR controls power-down mode, selects sampling frequency for eliminating noise
a subclock oscillator, and specifies multiplication ratio.

Bit	Bit Name	Initial Value	R/W	Description
7	DTON	0	R/W	Direct Transfer On Flag 0: When the SLEEP instruction is executed in hig mode or medium-speed mode, operation shifts mode, software standby mode, or watch mode 1: Setting prohibited

				0: Sampling using ϕ/32 1: Setting prohibited
4	SUBSTP	0	R/W	Subclock Oscillator Control Enables/disables a subclock oscillator. This bit sho set to 1 when a subclock is not used. 0: Enables subclock oscillator 1: Disables subclock oscillator
3	RFCUT	0	R/W	On-Chip Feedback Resistor Control Selects whether or not on-chip feedback resistor c system clock generator is used when an external c input. Do not access when the crystal resonator is After setting this bit in the external clock input stat software standby mode or watch mode. When soft standby mode or watch mode is entered, this bit s whether or not on-chip feedback resistor is used. 0: On-chip feedback resistor of the system clock g used 1: On-chip feedback resistor of the system clock g not used
2	—	0	R/W	Reserved This is a readable/writable bit, but the write value s always be 0.

11: Setting prohibited

Note: * When watch mode is entered, high-speed mode must be set.



Note: C_{L1} and C_{L2} are reference values including the floating capacitance of the board.

Figure 21.2 Connection of Crystal Resonator (Example)

Table 21.1 Damping Resistance Value

Frequency (MHz)	8	10	12	16	20	25
R_d (Ω)	200	100	0	0	0	0

Figure 21.3 shows the equivalent circuit of the crystal resonator. Use a crystal resonator t
the characteristics shown in table 21.2.

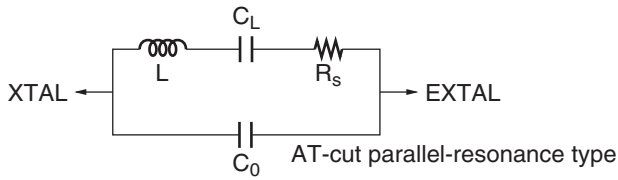


Figure 21.3 Crystal Resonator Equivalent Circuit

to the EXTAL pin, wait for longer than clock oscillator settling time to keep enough tim
settling before power is supplied or standby mode is canceled.

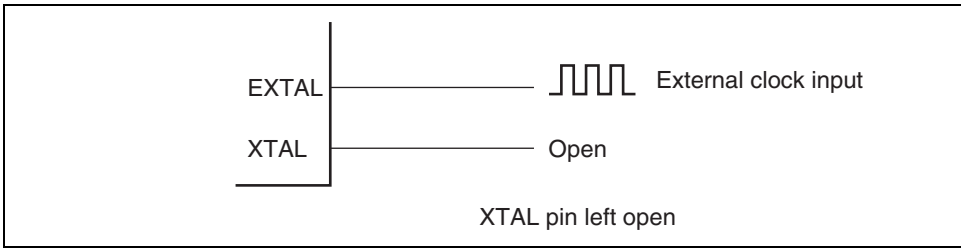


Figure 21.4 External Clock Input (Examples)

	multiplication					
External clock input low pulse width	t_{EXL}	0.4	0.6	0.4	0.6	t_{EXcyc}
External clock input high pulse width	t_{EXH}	0.4	0.6	0.4	0.6	t_{EXcyc}
External clock rise time	t_{EXr}	—	5	—	5	ns
External clock fall time	t_{EXf}	—	5	—	5	ns

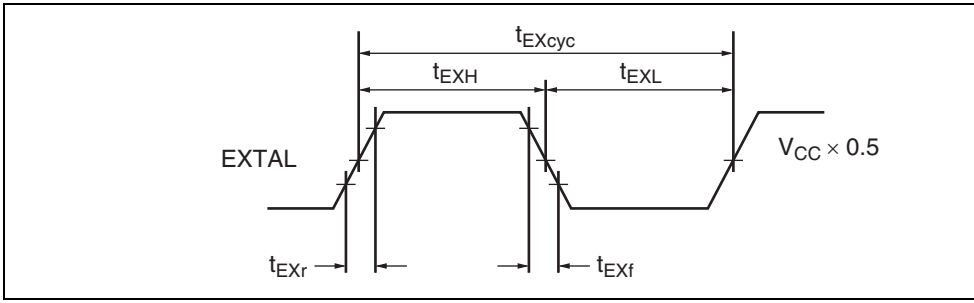


Figure 21.5 External Clock Input Timing

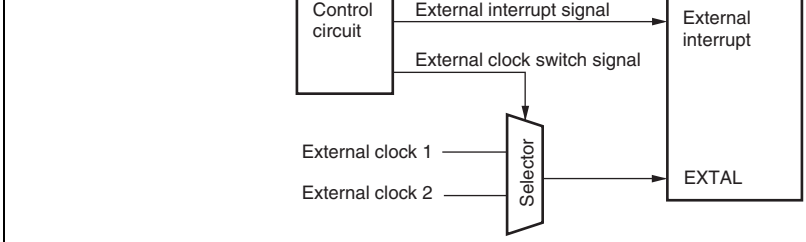


Figure 21.6 External Clock Switching Circuit (Examples)

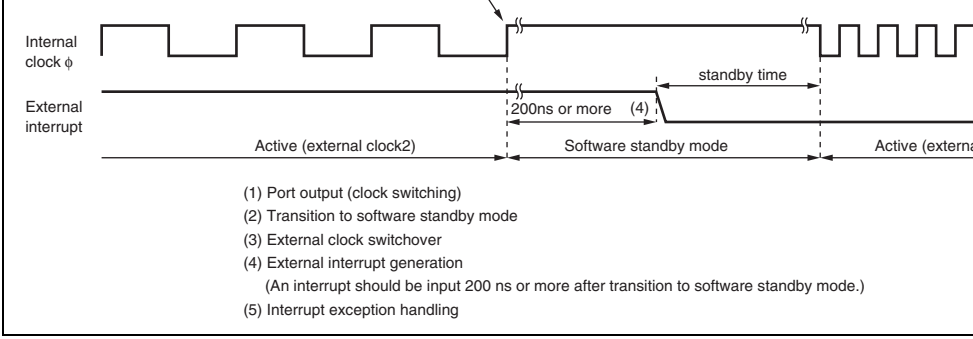


Figure 21.7 External Clock Switching Timing (Examples)

21.3 PLL Circuit

The PLL circuit multiplies the frequency from the clock pulse generator by one or two. T multiplication ratio is set by the STC1 and STC0 bits in LPWRCR. At the setting, the pha rising edge of an internal clock is controlled to match that of the rising edge of the EXTA

When changing the multiplication ratio of the PLL circuit, the operation differs according setting of the STCS bit in SCKCR.

When the STCS bit is 0, the changed multiplication ratio is valid after software standby m watch mode is entered. The transition time is set by the STS2 to STS0 bits in the standby register (SBYCR). For details on SBYCR, see section 22.1.1, Standby Control Register (S

1. In the initial state, the multiplication ratio of the PLL circuit is 1.
2. The transition time is set by the STS2 to STS0 bits.
3. The multiplication ratio is set by the STC1 and STC0 bits, and software standby mode watch mode is entered.

The medium-speed clock divider divides the system clock to generate φ/2, φ/4, φ/8, φ/16
φ/32.

21.5 Bus Master Clock Selection Circuit

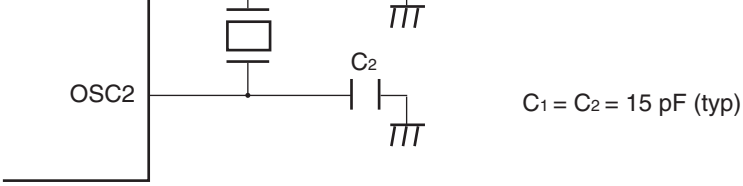
The bus master clock selection circuit selects the clock supplied to the bus master by set
SCK2 to SCK0 bits in SCKCR. The bus master clock can be selected from system clock
medium-speed clocks (φ/2, φ/4, φ/8, φ/16, and φ/32).

21.6 System Clock with IEBus

When using the IEBus*[1], the system clock should be set with one of 12 MHz, 12.58 MH
18 MHz, 18.87 MHz, 24 MHz, or 25.16 MHz.

When the IEBus*[1] is not used, any system clock frequency between 8 MHz and 26 MHz
used.

Notes: 1. The IEBus is supported only by the H8S/2552 Group.
2. System clock frequency up to 20 MHz is supported by the H8S/2556 Group.



Note: C1 and C2 are reference values including the floating capacitance of the boad.

Figure 21.8 Connection Example of 32.768-kHz Crystal Resonator

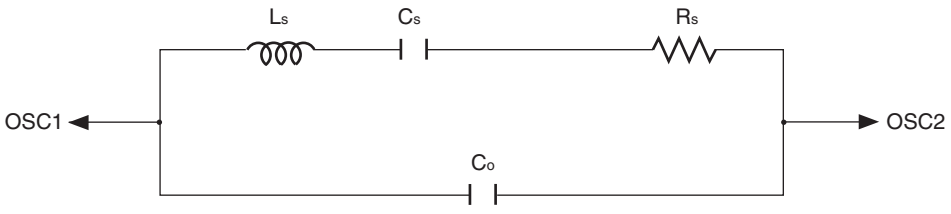


Figure 21.9 Equivalent Circuit for 32.768-kHz Crystal Resonator

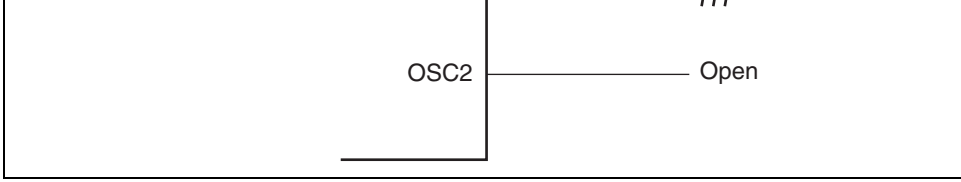


Figure 21.10 Pin Handling when Subclock Is Not Used

21.8 Subclock Waveform Generation Circuit

To eliminate noise from the subclock input from the OSC1 pin, the subclock is sampled dividing clock ϕ. The sampling frequency is set using the NESEL bit in LPWRCR. For section 21.1.2, Low-Power Control Register (LPWRCR).

No sampling is performed in watch mode.

21.9.2 Note on Board Design

When designing the board, place the crystal resonator and its load capacitors as close as p
to the XTAL, EXTAL, OSC1, and OSC2 pins. Make wires as short as possible. Other sig
should be routed away from the oscillator circuit, as shown in figure 21.11. This is to pre
induction from interfering with correct oscillation.

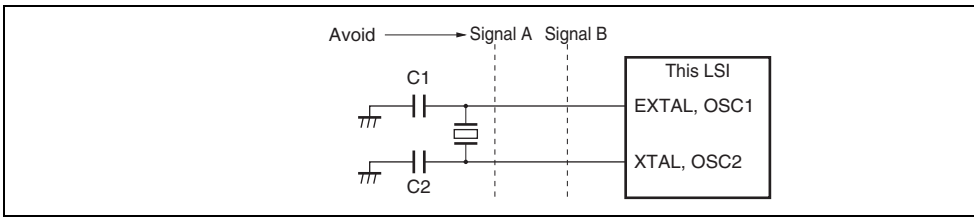


Figure 21.11 Note on Board Design of Oscillator Circuit

Figure 21.12 shows the recommended connection circuit between the power supply pins a
pin. The CB which is a capacitor for stabilization should be inserted near the pin between
power supply pins (V_{CC}, V_{CL}, $P1V_{CC}$, and $P2V_{CC}$) and Vss pin. Two CBs should be placed
$P1V_{CC}$ line. Other signal lines should not be crossed.

CB = 0.47 µF (recommended value)
Note: The CB is a laminated ceramic.

Figure 21.12 Recommended Connection Circuit between Power Supply Pins and

(2) Medium-speed mode
(3) Sleep mode
(4) Watch mode
(5) Module stop mode
(6) Software standby mode
(7) Hardware standby mode

Of these, (2) to (7) are power-down modes. Sleep mode is a CPU state, medium-speed m
CPU and bus master states, and module stop mode is an on-chip peripheral function (inc
masters other than the CPU) state.

After a reset, the LSI is in high-speed mode.

Table 22.1 shows the internal state of the LSI in the respective modes. Table 22.2 shows
conditions for shifting between power-down modes.

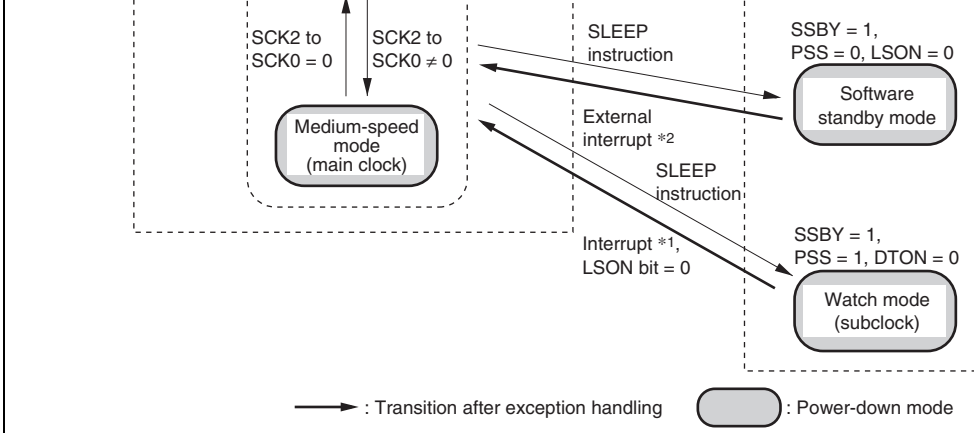
Figure 22.1 is a mode transition diagram.

Table 22.1 LSI Internal States in Each Mode

Function	High-Speed	Medium-Speed	Sleep	Module Stop	Watch	Software Standby
System clock pulse generator	Functioning	Functioning	Functioning	Functioning	Halted	Halted
Subclock pulse generator	Functioning /halted	Functioning /halted	Functioning /halted	Functioning /halted	Function-ing	Functioning /halted

	IRQn							
Peripheral functions	PBC	Functioning	Medium-speed operation	Functioning	Functioning/ halted (retained)	Halted (retained)	Halted (retained)	H (r
	DTC	Functioning	Medium-speed operation	Functioning	Functioning/ halted (retained)	Halted (retained)	Halted (retained)	H (r
	WDT_1	Functioning	Functioning	Functioning	Functioning	Subclock operation	Halted (retained)	H (r
	WDT_0	Functioning	Functioning	Functioning	Functioning	Halted (retained)	Halted (retained)	H (r
	TMR	Functioning	Functioning	Functioning	Functioning/ halted (retained)	Halted (retained)	Halted (retained)	H (r
	TPU	Functioning	Functioning	Functioning	Functioning/ halted (retained)	Halted (retained)	Halted (retained)	H (r
	SCI							
	I^2C2				Halted*3 (reset/ retained)	Halted*3 (reset/ retained)	Halted*3 (reset/ retained)	
	D/A				Functioning/ halted (retained) *4	Halted (retained) *4	Halted (retained) *4	

1. Supported only by the H8S/2552 Group.
2. Supported only by the H8S/2556 Group.
3. BC2 to BC0 are halted (reset) and other registers are halted (retained).
4. "Halted (retained)" means that internal register values are retained. For analo
   the given D/A absolute accuracy is not satisfied because the internal state is
   suspended."



Notes: When a transition is made between modes by means of an interrupt, the transition cannot be ma
on interrupt source generation alone. Ensure that interrupt handling is performed after accepting
interrupt request.
From any state except hardware standby mode, a transition to the reset state occurs when $\overline{\text{RES}}$
driven low. From any state except hardware standby mode and power-on reset state, a transitio
the manual reset state occurs when $\overline{\text{MRES}}$ is driven low.
From any state, a transition to hardware standby mode occurs when $\overline{\text{STBY}}$ is driven low.
Always select high-speed mode before making a transition to watch mode.
1. NMI, IRQ0 to IRQ7, WDT1 interrupt
2. NMI, IRQ0 to IRQ7

Figure 22.1 Mode Transition Diagram

1	0	1	X	—	—
1	1	0	0	Watch	High-speed
1	1	1	X	—	—
1	1	0	1	—	—

Legend:
X: Don't care
—: Do not set.

22.1 Register Descriptions

The following registers relate to the power-down modes. For details on the system clock
register (SCKCR), refer to section 21.1.1, System Clock Control Register (SCKCR). Fo
on the low power control register (LPWRCR), refer to section 21.1.2, Low-Power Contr
Register (LPWRCR). For details on the timer control/status register (TCSR_1), refer to
11.3.5, Timer Control/Status Register (TCSR).

- Standby control register (SBYCR)
- Module stop control register A (MSTPCRA)
- Module stop control register B (MSTPCRB)
- Module stop control register C (MSTPCRC)
- Low power control register (LPWRCR)
- System clock control register (SCKCR)
- Timer control/status register (TCSR_1)

				1: Shifts to software standby mode or watch mode the SLEEP instruction is executed in high-speed or medium-speed mode. Note that the value of the SSBY bit does not chan when software standby mode is canceled and mak normal operation mode transition by executing an interrupt. To clear this bit, 0 should be written to.
6	STS2	0	R/W	Standby Timer Select 2 to 0
5	STS1	0	R/W	These bits select the MCU wait time for clock settl
4	STS0	0	R/W	cancel software standby mode or watch mode by an external interrupt. With a crystal resonator (see tables 22.3 and 24.6 a wait time of 8 ms (oscillation settling time) or mo depending on the operating frequency. 000: Standby time = 8192 states 001: Standby time = 16384 states 010: Standby time = 32768 states 011: Standby time = 65536 states 100: Standby time = 131072 states 101: Standby time = 262144 states 110: Reserved 111: Reserved

Note: * The $\overline{CS1}$ and $\overline{CS2}$ signals are not supported by the H8S/2556 Group.

22.1.2 Module Stop Control Registers A to C (MSTPCRA to MSTPCRC)

MSTPCR performs module stop mode control. When bits in MSTPCR are set to 1, mod
mode is set. When cleared to 0, module stop mode is cleared.

- MSTPCRA

Bit	Bit Name	Initial Value	R/W	Target Module
7	MSTPA7*[1]	0	R/W	
6	MSTPA6	0	R/W	Data transfer controller (DTC)
5	MSTPA5	1	R/W	16-bit timer pulse unit (TPU)
4	MSTPA4	1	R/W	8-bit timer (TMR_0, TMR_1)
3	MSTPA3*[1]	1	R/W	
2	MSTPA2*[1]	1	R/W	
1	MSTPA1	1	R/W	A/D converter
0	MSTPA0	1	R/W	8-bit timer (TMR_2, TMR_3)

1	MSTPB1	1	R/W	
0	MSTPB0*[1]	1	R/W	

- MSTPCRC

Bit	Bit Name	Initial Value	R/W	Target Module
7	MSTPC7	1	R/W	Serial communication interface 3 (SCI_3)
6	MSTPC6	1	R/W	Serial communication interface 4 (SCI_4)
5	MSTPC5	1	R/W	D/A converter
4	MSTPC4	1	R/W	PC break controller (PBC)
3	MSTPC3	1	R/W	IEBus™ controller (IEB)*[2]
2	MSTPC2	1	R/W	Controller area network (HCAN)*[3]
1	MSTPC1*[1]	1	R/W	
0	MSTPC0*[1]	1	R/W	

Notes: 1. The MSTPA7 bit can be read from or written to. This bit is initialized to 0. The v
value should always be 0. The MSTPA3, MSTPA2, MSTPB2 to MSTPB0, MST
and MSTPC0 bits can be read from or written to. These bits are initialized to 1.
write value should always be 1.
2. Supported only by the H8S/2552 Group.
3. Supported only by the H8S/2556 Group.

memory is accessed in 4 states, and internal I/O registers in 8 states.

Medium-speed mode is cleared by clearing all of bits SCK2 to SCK0 to 0. A transition i
high-speed mode and medium-speed mode is cleared at the end of the current bus cycle.

If a SLEEP instruction is executed when the SSBY bit in SBYCR and the LSON bit in I
are cleared to 0, a transition is made to sleep mode. When sleep mode is cleared by an ir
medium-speed mode is restored.

When the SLEEP instruction is executed with the SSBY bit = 1, LSON bit = 0, and PSS
TCSR_1 (WDT_1) = 0, operation shifts to the software standby mode. When software s
mode is cleared by an external interrupt, medium-speed mode is restored.

When the $\overline{\text{RES}}$ pin is set low and medium-speed mode is cancelled, operation shifts to th
state. The same applies in the case of a reset caused by overflow of the watchdog timer.

When the $\overline{\text{STBY}}$ pin is driven low, a transition is made to hardware standby mode.

Figure 22.2 shows the timing for transition to and clearance of medium-speed mode.

Figure 22.2 Medium-Speed Mode Transition and Clearance Timing

22.3 Sleep Mode

22.3.1 Transition to Sleep Mode

When the SLEEP instruction is executed while the SSBY bit in SBYCR = 0 and the LSO
LPWRCR = 0, the CPU enters sleep mode. In sleep mode, CPU operation stops but the c
the CPU's internal registers are retained. Other peripheral modules do not stop.

22.3.2 Clearing Sleep Mode

Sleep mode is cleared by any interrupt, or signals at the $\overline{\text{RES}}$, $\overline{\text{MRES}}$, or $\overline{\text{STBY}}$ pin.

- Clearing with an interrupt
  When an interrupt occurs, sleep mode is cleared and interrupt exception processing st
  Sleep mode is not cleared if the interrupt is disabled, or interrupts other than NMI are
  by the CPU.
- Clearing with the $\overline{\text{RES}}$ pin or $\overline{\text{MRES}}$ pin
  Setting the $\overline{\text{RES}}$ pin or the $\overline{\text{MRES}}$ pin level low selects the reset state. After the stipul
  input duration, driving the $\overline{\text{RES}}$ pin or $\overline{\text{MRES}}$ pin high starts the CPU performing rese
  exception processing.
- Clearing with the $\overline{\text{STBY}}$ Pin
  When the $\overline{\text{STBY}}$ pin is driven low, a transition is made to hardware standby mode.

consumption is significantly reduced.

Notes: 1. Supported only by the H8S/2556 Group.
2. Supported only by the H8S/2552 Group.

22.4.2 Clearing Software Standby Mode

Software standby mode is cleared by an external interrupt (NMI pin, or pins $\overline{\text{IRQ0}}$ to $\overline{\text{IR}}$ means of the $\overline{\text{RES}}$ pin, $\overline{\text{MRES}}$ pin, or $\overline{\text{STBY}}$ pin.

- Clearing with an interrupt

  When an NMI or IRQ0 to IRQ7 interrupt request signal is input, clock oscillation sta after the elapse of the time set in bits STS2 to STS0 in SBYCR, stable clocks are sup the entire chip, software standby mode is cleared, and interrupt exception handling is

  When clearing software standby mode with an IRQ0 to IRQ7 interrupt, set the corres enable bit/pin function switching bit to 1 and ensure that no interrupt with a higher p than interrupts IRQ0 to IRQ7 is generated. Software standby mode cannot be cleared interrupt has been masked on the CPU side or has been designated as a DTC activati
- Clearing with the $\overline{\text{RES}}$ pin or $\overline{\text{MRES}}$ pin

  When the $\overline{\text{RES}}$ pin or $\overline{\text{MRES}}$ pin is driven low, clock oscillation is started. At the sam clock oscillation starts, clocks are supplied to the entire chip. Note that the $\overline{\text{RES}}$ pin a pin must be held low until clock oscillation settles. When the $\overline{\text{RES}}$ pin or $\overline{\text{MRES}}$ pin the CPU begins reset exception handling.
- Clearing with the $\overline{\text{STBY}}$ pin

  When the $\overline{\text{STBY}}$ pin is driven low, a transition is made to hardware standby mode.

	1	0	32768 states	1.3	1.6	2.0	2.5	3.3	4.1
		1	65536 states	2.6	3.3	4.1	5.0	6.6	8.2
1	0	0	131072 states	5.2	6.6	8.2	10.1	13.1	16.4
		1	262144 states	10.5	13.1	16.4	20.2	26.2	32.8
	1	X	Reserved	—	—	—	—	—	—

Shading: Recommended time setting

Legend:

X: Don't care

Note: * The H8S/2556 Group supports the operating frequencies up to 20 MHz.

22.4.4 Software Standby Mode Application Example

Figure 22.3 shows an example in which a transition is made to software standby mode at a falling edge on the NMI pin, and software standby mode is cleared at the rising edge on t pin.

In this example, an NMI interrupt is accepted with the NMIEG bit in SYSCR cleared to 0 edge specification), then the NMIEG bit is set to 1 (rising edge specification), the SSBY b to 1, and a SLEEP instruction is executed, causing a transition to software standby mode.

Software standby mode is then cleared at the rising edge on the NMI pin.

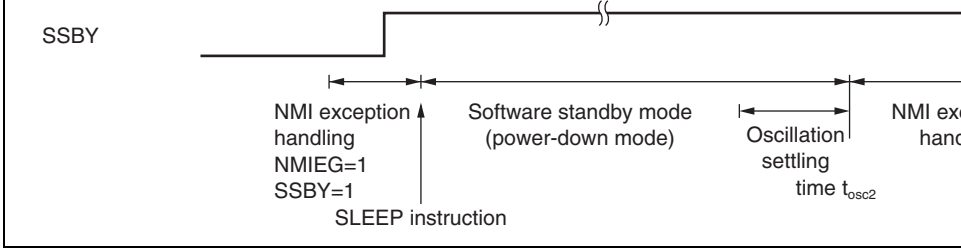


Figure 22.3 Software Standby Mode Application Example

Do not change the state of the mode pins (MD2 to MD0) during hardware standby mode.

22.5.2 Clearing Hardware Standby Mode

Hardware standby mode is cleared by means of the $\overline{\text{STBY}}$ pin and the $\overline{\text{RES}}$ pin. When the pin is driven high while the $\overline{\text{RES}}$ pin is low, the reset state is set and clock oscillation is st Ensure that the $\overline{\text{RES}}$ pin is held low until the clock oscillator settles (at least t_{osc1} ms). Whe $\overline{\text{RES}}$ pin is subsequently driven high, a transition is made to the program execution state v reset exception handling state.

22.5.3 Hardware Standby Mode Timing

Figure 22.4 shows an example of hardware standby mode timing.

When the $\overline{\text{STBY}}$ pin is driven low after the $\overline{\text{RES}}$ pin has been driven low, a transition is m hardware standby mode. Hardware standby mode is cleared by driving the $\overline{\text{STBY}}$ pin high waiting for the oscillation settling time, then changing the $\overline{\text{RES}}$ pin from low to high.

Figure 22.4 Hardware Standby Mode Timing

22.6 Module Stop Mode

Module stop mode can be set for individual on-chip peripheral modules.

When the corresponding MSTP bit in MSTPCR is set to 1, module operation stops at th
the bus cycle and a transition is made to module stop mode. The CPU continues operati
independently.

When the corresponding MSTP bit is cleared to 0, module stop mode is cleared and the
starts operating at the end of the bus cycle. In module stop mode, the internal states of m
other than the A/D converter are retained.

After reset clearance, all modules other than the DTC are in module stop mode.

When an on-chip peripheral module is in module stop mode, read/write access to its reg
disabled.

Since the operations of the bus controller and I/O port are stopped when sleep mode is e
the all-module stop state (MSTPCR=H'FFFFFFFF), power consumption can further be r

functions) and I/O ports are retained. To make a transition to watch mode, bits SCK2 to S
SCKCR must be set to 0.

Notes: 1. Supported only by the H8S/2556 Group.
2. Supported only by the H8S/2552 Group.

22.7.2 Clearing Watch Mode

Watch mode is cleared by any interrupt (WOVI1 interrupt, NMI pin, or $\overline{\text{IRQ0}}$ to $\overline{\text{IRQ7}}$), o
at the $\overline{\text{RES}}$, $\overline{\text{MRES}}$, or $\overline{\text{STBY}}$ pin.

- Clearing with an interrupt

  When an interrupt occurs, watch mode is cleared and a transition is made to high-spee
  or medium-speed mode. When a transition is made to high-speed mode, a stable clock
  supplied to all LSI circuits and interrupt exception processing starts after the time set
  STS2 to STS0 bits in SBYCR has elapsed. In the case of IRQ0 to IRQ7 interrupts, no
  transition is made from watch mode if the corresponding enable bit/pin function switc
  has been cleared to 0, and, in the case of interrupts from the internal peripheral modul
  interrupt enable register has been set to disable the reception of that interrupt, or is ma
  the CPU.

  See section 22.4.3, Oscillation Settling Time after Clearing Software Standby Mode, f
  to set the oscillation settling time when making a transition from watch mode to high-
  mode.
- Clearing with the $\overline{\text{RES}}$ pin or $\overline{\text{MRES}}$ pin

  For clearing watch mode by the $\overline{\text{RES}}$ pin or $\overline{\text{MRES}}$ pin, see section 22.4.2, Clearing S
  Standby Mode.
- Clearing with the $\overline{\text{STBY}}$ pin

  When the $\overline{\text{STBY}}$ pin is driven low, a transition is made to hardware standby mode.

PSTOP	—	0	1
Hardware standby mode	High impedance	High impedance	High impeda
Software standby mode, watch mode	High impedance	Fixed to high	Fixed to high
Sleep mode	High impedance	ϕ output	Fixed to high
High-speed mode, medium-speed mode	High impedance	ϕ output	Fixed to high

22.9.3 DTC Module Stop

Depending on the operating status of the DTC, the MSTPA6 bit may not be set to 1. Setti
DTC module stop mode should be carried out only when the respective module is not act

For details, refer to section 8, Data Transfer Controller (DTC).

22.9.4 On-Chip Peripheral Module Interrupt

- Module stop mode

  Relevant interrupt operations cannot be performed in module stop mode. Consequentl
  module stop mode is entered when an interrupt has been requested, it will not be poss
  clear the CPU interrupt source or the DTC activation source.

  Interrupts should therefore be disabled before entering module stop mode.
- Watch mode

  On-chip peripheral modules (DTC, TPU, and IIC2) that stop operation in watch mode
  clear interrupt sources of the CPU after they make a transition to watch mode while a
  interrupt is being requested.

  Interrupts should therefore be disabled before executing the SLEEP instruction and en
  watch mode.

22.9.5 Writing to MSTPCR

MSTPCR should only be written to by the CPU.

2. Register Bits

- Bit configurations of the registers are described in the same order as the Register Ad
(by functional module, in ascending order of addresses).
- Reserved bits are indicated by — in the bit name.
- No entry in the bit-name column indicates that the whole register is allocated as a co
for holding data.

3. Register States in Each Operating Mode

- Register states are described in the same order as the Register Addresses (by functio
module, in ascending order of addresses).
- The register states described are for the basic operating modes. If there is a specific r
on-chip module, refer to the section on that on-chip module.

DTC source address register	SAR	24	H'EFBF	DTC	16/32*2
DTC mode register B	MRB	8		DTC	16/32*2
DTC destination address register	DAR	24		DTC	16/32*2
DTC transfer count register A	CRA	16		DTC	16/32*2
DTC transfer count register B	CRB	16		DTC	16/32*2
IEBus control register	IECTR	8	H'F800	IEB	8
IEBus command register	IECMR	8	H'F801	IEB	8
IEBus master control register	IEMCR	8	H'F802	IEB	8
IEBus master unit address register 1	IEAR1	8	H'F803	IEB	8
IEBus master unit address register 2	IEAR2	8	H'F804	IEB	8
IEBus slave address setting register 1	IESA1	8	H'F805	IEB	8
IEBus slave address setting register 2	IESA2	8	H'F806	IEB	8
IEBus transmit frame length register	IETBFL	8	H'F807	IEB	8
IEBus transmit buffer register	IETBR	8	H'F808	IEB	8
IEBus reception master address register 1	IEMA1	8	H'F809	IEB	8
IEBus reception master address register 2	IEMA2	8	H'F80A	IEB	8
IEBus receive control field register	IERCTL	8	H'F80B	IEB	8
IEBus receive frame length register	IERBFL	8	H'F80C	IEB	8
IEBus receive buffer register	IERBR	8	H'F80D	IEB	8
IEBus lock address register 1	IELA1	8	H'F80E	IEB	8
IEBus lock address register 2	IELA2	8	H'F80F	IEB	8

IEBus receive interrupt enable register	IEIER	8	H'F815	IEB	8
IEBus receive error flag register	IEREF	8	H'F816	IEB	8
Port H data direction register	PHDDR	8	H'FA10	PORT	8
Port J data direction register	PJDDR	8	H'FA11	PORT	8
Port H data register	PHDR	8	H'FA12	PORT	8
Port J data register	PJDR	8	H'FA13	PORT	8
Port H register	PORTH	8	H'FA14	PORT	8
Port J register	PORTJ	8	H'FA15	PORT	8
IIC bus control register 1_0	ICCR1_0	8	H'FA20	IIC2_0	8
IIC bus control register 2_0	ICCR2_0	8	H'FA21	IIC2_0	8
IIC bus mode register_0	ICMR_0	8	H'FA22	IIC2_0	8
IIC bus interrupt enable register_0	ICIER_0	8	H'FA23	IIC2_0	8
IIC bus status register_0	ICSR_0	8	H'FA24	IIC2_0	8
Slave address register_0	SAR_0	8	H'FA25	IIC2_0	8
IIC bus transmit data register_0	ICDRT_0	8	H'FA26	IIC2_0	8
IIC bus receive data register_0	ICDRR_0	8	H'FA27	IIC2_0	8
IIC bus control register 1_1	ICCR1_1	8	H'FA28	IIC2_1	8
IIC bus control register 2_1	ICCR2_1	8	H'FA29	IIC2_1	8
IIC bus mode register_1	ICMR_1	8	H'FA2A	IIC2_1	8
IIC bus interrupt enable register_1	ICIER_1	8	H'FA2B	IIC2_1	8
IIC bus status register_1	ICSR_1	8	H'FA2C	IIC2_1	8
Slave address register_1	SAR_1	8	H'FA2D	IIC2_1	8
IIC bus transmit data register_1	ICDRT_1	8	H'FA2E	IIC2_1	8

Transmit wait cancel register	TXCR	16	H'FB08	HCAN	16
Transmit acknowledge register	TXACK	16	H'FB0A	HCAN	16
Abort acknowledge register	ABACK	16	H'FB0C	HCAN	16
Receive complete register	RXPR	16	H'FB0E	HCAN	16
Remote request register	RFPR	16	H'FB10	HCAN	16
Interrupt register	IRR	16	H'FB12	HCAN	16
Mailbox interrupt mask register	MBIMR	16	H'FB14	HCAN	16
Interrupt mask register	IMR	16	H'FB16	HCAN	16
Receive error counter	REC	8	H'FB18	HCAN	16
Transmit error counter	TEC	8	H'FB19	HCAN	16
Unread message status register	UMSR	16	H'FB1A	HCAN	16
Local acceptance filter mask L	LAFML	16	H'FB1C	HCAN	16
Local acceptance filter mask H	LAFMH	16	H'FB1E	HCAN	16
Message control 0[1]	MC0[1]	8	H'FB20	HCAN	16
Message control 0[2]	MC0[2]	8	H'FB21	HCAN	16
Message control 0[3]	MC0[3]	8	H'FB22	HCAN	16
Message control 0[4]	MC0[4]	8	H'FB23	HCAN	16
Message control 0[5]	MC0[5]	8	H'FB24	HCAN	16
Message control 0[6]	MC0[6]	8	H'FB25	HCAN	16
Message control 0[7]	MC0[7]	8	H'FB26	HCAN	16
Message control 0[8]	MC0[8]	8	H'FB27	HCAN	16
Message control 1[1]	MC1[1]	8	H'FB28	HCAN	16
Message control 1[2]	MC1[2]	8	H'FB29	HCAN	16

Message control 2[1]	MC2[1]	8	H'FB30	HCAN	16
Message control 2[2]	MC2[2]	8	H'FB31	HCAN	16
Message control 2[3]	MC2[3]	8	H'FB32	HCAN	16
Message control 2[4]	MC2[4]	8	H'FB33	HCAN	16
Message control 2[5]	MC2[5]	8	H'FB34	HCAN	16
Message control 2[6]	MC2[6]	8	H'FB35	HCAN	16
Message control 2[8]	MC2[8]	8	H'FB37	HCAN	16
Message control 3[1]	MC3[1]	8	H'FB38	HCAN	16
Message control 3[2]	MC3[2]	8	H'FB39	HCAN	16
Message control 3[3]	MC3[3]	8	H'FB3A	HCAN	16
Message control 3[4]	MC3[4]	8	H'FB3B	HCAN	16
Message control 3[5]	MC3[5]	8	H'FB3C	HCAN	16
Message control 3[6]	MC3[6]	8	H'FB3D	HCAN	16
Message control 3[7]	MC3[7]	8	H'FB3E	HCAN	16
Message control 3[8]	MC3[8]	8	H'FB3F	HCAN	16
Message control 4[1]	MC4[1]	8	H'FB40	HCAN	16
Message control 4[2]	MC4[2]	8	H'FB41	HCAN	16
Message control 4[3]	MC4[3]	8	H'FB42	HCAN	16
Message control 4[4]	MC4[4]	8	H'FB43	HCAN	16
Message control 4[5]	MC4[5]	8	H'FB44	HCAN	16
Message control 4[6]	MC4[6]	8	H'FB45	HCAN	16
Message control 4[7]	MC4[7]	8	H'FB46	HCAN	16
Message control 4[8]	MC4[8]	8	H'FB47	HCAN	16

Message control 5[7]	MC5[7]	8	H'FB4E	HCAN	16
Message control 5[8]	MC5[8]	8	H'FB4F	HCAN	16
Message control 6[1]	MC6[1]	8	H'FB50	HCAN	16
Message control 6[2]	MC6[2]	8	H'FB51	HCAN	16
Message control 6[3]	MC6[3]	8	H'FB52	HCAN	16
Message control 6[4]	MC6[4]	8	H'FB53	HCAN	16
Message control 6[5]	MC6[5]	8	H'FB54	HCAN	16
Message control 6[6]	MC6[6]	8	H'FB55	HCAN	16
Message control 6[7]	MC6[7]	8	H'FB56	HCAN	16
Message control 6[8]	MC6[8]	8	H'FB57	HCAN	16
Message control 7[1]	MC7[1]	8	H'FB58	HCAN	16
Message control 7[2]	MC7[2]	8	H'FB59	HCAN	16
Message control 7[3]	MC7[3]	8	H'FB5A	HCAN	16
Message control 7[4]	MC7[4]	8	H'FB5B	HCAN	16
Message control 7[5]	MC7[5]	8	H'FB5C	HCAN	16
Message control 7[6]	MC7[6]	8	H'FB5D	HCAN	16
Message control 7[7]	MC7[7]	8	H'FB5E	HCAN	16
Message control 7[8]	MC7[8]	8	H'FB5F	HCAN	16
Message control 8[1]	MC8[1]	8	H'FB60	HCAN	16
Message control 8[2]	MC8[2]	8	H'FB61	HCAN	16
Message control 8[3]	MC8[3]	8	H'FB62	HCAN	16
Message control 8[4]	MC8[4]	8	H'FB63	HCAN	16
Message control 8[5]	MC8[5]	8	H'FB64	HCAN	16

Message control 9[4]	MC9[4]	8	H'FB6B	HCAN	16
Message control 9[5]	MC9[5]	8	H'FB6C	HCAN	16
Message control 9[6]	MC9[6]	8	H'FB6D	HCAN	16
Message control 9[7]	MC9[7]	8	H'FB6E	HCAN	16
Message control 9[8]	MC9[8]	8	H'FB6F	HCAN	16
Message control 10[1]	MC10[1]	8	H'FB70	HCAN	16
Message control 10[2]	MC10[2]	8	H'FB71	HCAN	16
Message control 10[3]	MC10[3]	8	H'FB72	HCAN	16
Message control 10[4]	MC10[4]	8	H'FB73	HCAN	16
Message control 10[5]	MC10[5]	8	H'FB74	HCAN	16
Message control 10[6]	MC10[6]	8	H'FB75	HCAN	16
Message control 10[7]	MC10[7]	8	H'FB76	HCAN	16
Message control 10[8]	MC10[8]	8	H'FB77	HCAN	16
Message control 11[1]	MC11[1]	8	H'FB78	HCAN	16
Message control 11[2]	MC11[2]	8	H'FB79	HCAN	16
Message control 11[3]	MC11[3]	8	H'FB7A	HCAN	16
Message control 11[4]	MC11[4]	8	H'FB7B	HCAN	16
Message control 11[5]	MC11[5]	8	H'FB7C	HCAN	16
Message control 11[6]	MC11[6]	8	H'FB7D	HCAN	16
Message control 11[7]	MC11[7]	8	H'FB7E	HCAN	16
Message control 11[8]	MC11[8]	8	H'FB7F	HCAN	16
Message control 12[1]	MC12[1]	8	H'FB80	HCAN	16
Message control 12[2]	MC12[2]	8	H'FB81	HCAN	16

Message control 13[1]	MC13[1]	8	H'FB88	HCAN	16
Message control 13[2]	MC13[2]	8	H'FB89	HCAN	16
Message control 13[3]	MC13[3]	8	H'FB8A	HCAN	16
Message control 13[4]	MC13[4]	8	H'FB8B	HCAN	16
Message control 13[5]	MC13[5]	8	H'FB8C	HCAN	16
Message control 13[6]	MC13[6]	8	H'FB8D	HCAN	16
Message control 13[7]	MC13[7]	8	H'FB8E	HCAN	16
Message control 13[8]	MC13[8]	8	H'FB8F	HCAN	16
Message control 14[1]	MC14[1]	8	H'FB90	HCAN	16
Message control 14[2]	MC14[2]	8	H'FB91	HCAN	16
Message control 14[3]	MC14[3]	8	H'FB92	HCAN	16
Message control 14[4]	MC14[4]	8	H'FB93	HCAN	16
Message control 14[5]	MC14[5]	8	H'FB94	HCAN	16
Message control 14[6]	MC14[6]	8	H'FB95	HCAN	16
Message control 14[7]	MC14[7]	8	H'FB96	HCAN	16
Message control 14[8]	MC14[8]	8	H'FB97	HCAN	16
Message control 15[1]	MC15[1]	8	H'FB98	HCAN	16
Message control 15[2]	MC15[2]	8	H'FB99	HCAN	16
Message control 15[3]	MC15[3]	8	H'FB9A	HCAN	16
Message control 15[4]	MC15[4]	8	H'FB9B	HCAN	16
Message control 15[5]	MC15[5]	8	H'FB9C	HCAN	16
Message control 15[6]	MC15[6]	8	H'FB9D	HCAN	16
Message control 15[7]	MC15[7]	8	H'FB9E	HCAN	16

Message data 0[6]	MD0[6]	8	H'FBB5	HCAN	16
Message data 0[7]	MD0[7]	8	H'FBB6	HCAN	16
Message data 0[8]	MD0[8]	8	H'FBB7	HCAN	16
Message data 1[1]	MD1[1]	8	H'FBB8	HCAN	16
Message data 1[2]	MD1[2]	8	H'FBB9	HCAN	16
Message data 1[3]	MD1[3]	8	H'FBBA	HCAN	16
Message data 1[4]	MD1[4]	8	H'FBBB	HCAN	16
Message data 1[5]	MD1[5]	8	H'FBBC	HCAN	16
Message data 1[6]	MD1[6]	8	H'FBBD	HCAN	16
Message data 1[7]	MD1[7]	8	H'FBBE	HCAN	16
Message data 1[8]	MD1[8]	8	H'FBBF	HCAN	16
Message data 2[1]	MD2[1]	8	H'FBC0	HCAN	16
Message data 2[2]	MD2[2]	8	H'FBC1	HCAN	16
Message data 2[3]	MD2[3]	8	H'FBC2	HCAN	16
Message data 2[4]	MD2[4]	8	H'FBC3	HCAN	16
Message data 2[5]	MD2[5]	8	H'FBC4	HCAN	16
Message data 2[6]	MD2[6]	8	H'FBC5	HCAN	16
Message data 2[7]	MD2[7]	8	H'FBC6	HCAN	16
Message data 2[8]	MD2[8]	8	H'FBC7	HCAN	16
Message data 3[1]	MD3[1]	8	H'FBC8	HCAN	16
Message data 3[2]	MD3[2]	8	H'FBC9	HCAN	16
Message data 3[3]	MD3[3]	8	H'FBCA	HCAN	16
Message data 3[4]	MD3[4]	8	H'FBCB	HCAN	16

Message data 4[3]	MD4[3]	8	H'FBD2	HCAN	16
Message data 4[4]	MD4[4]	8	H'FBD3	HCAN	16
Message data 4[5]	MD4[5]	8	H'FBD4	HCAN	16
Message data 4[6]	MD4[6]	8	H'FBD5	HCAN	16
Message data 4[7]	MD4[7]	8	H'FBD6	HCAN	16
Message data 4[8]	MD4[8]	8	H'FBD7	HCAN	16
Message data 5[1]	MD5[1]	8	H'FBD8	HCAN	16
Message data 5[2]	MD5[2]	8	H'FBD9	HCAN	16
Message data 5[3]	MD5[3]	8	H'FBDA	HCAN	16
Message data 5[4]	MD5[4]	8	H'FBDB	HCAN	16
Message data 5[5]	MD5[5]	8	H'FBDC	HCAN	16
Message data 5[6]	MD5[6]	8	H'FBDD	HCAN	16
Message data 5[7]	MD5[7]	8	H'FBDE	HCAN	16
Message data 5[8]	MD5[8]	8	H'FBDF	HCAN	16
Message data 6[1]	MD6[1]	8	H'FBE0	HCAN	16
Message data 6[2]	MD6[2]	8	H'FBE1	HCAN	16
Message data 6[3]	MD6[3]	8	H'FBE2	HCAN	16
Message data 6[4]	MD6[4]	8	H'FBE3	HCAN	16
Message data 6[5]	MD6[5]	8	H'FBE4	HCAN	16
Message data 6[6]	MD6[6]	8	H'FBE5	HCAN	16
Message data 6[7]	MD6[7]	8	H'FBE6	HCAN	16
Message data 6[8]	MD6[8]	8	H'FBE7	HCAN	16
Message data 7[1]	MD7[1]	8	H'FBE8	HCAN	16

Message data 7[8]	MD7[8]	8	H'FBEF	HCAN	16
Message data 8[1]	MD8[1]	8	H'FBF0	HCAN	16
Message data 8[2]	MD8[2]	8	H'FBF1	HCAN	16
Message data 8[3]	MD8[3]	8	H'FBF2	HCAN	16
Message data 8[4]	MD8[4]	8	H'FBF3	HCAN	16
Message data 8[5]	MD8[5]	8	H'FBF4	HCAN	16
Message data 8[6]	MD8[6]	8	H'FBF5	HCAN	16
Message data 8[7]	MD8[7]	8	H'FBF6	HCAN	16
Message data 8[8]	MD8[8]	8	H'FBF7	HCAN	16
Message data 9[1]	MD9[1]	8	H'FBF8	HCAN	16
Message data 9[2]	MD9[2]	8	H'FBF9	HCAN	16
Message data 9[3]	MD9[3]	8	H'FBFA	HCAN	16
Message data 9[4]	MD9[4]	8	H'FBFB	HCAN	16
Message data 9[5]	MD9[5]	8	H'FBFC	HCAN	16
Message data 9[6]	MD9[6]	8	H'FBFD	HCAN	16
Message data 9[7]	MD9[7]	8	H'FBFE	HCAN	16
Message data 9[8]	MD9[8]	8	H'FBFF	HCAN	16
Message data 10[1]	MD10[1]	8	H'FC00	HCAN	16
Message data 10[2]	MD10[2]	8	H'FC01	HCAN	16
Message data 10[3]	MD10[3]	8	H'FC02	HCAN	16
Message data 10[4]	MD10[4]	8	H'FC03	HCAN	16
Message data 10[5]	MD10[5]	8	H'FC04	HCAN	16
Message data 10[6]	MD10[6]	8	H'FC05	HCAN	16

Message data 11[5]	MD11[5]	8	H'FC0C	HCAN	16
Message data 11[6]	MD11[6]	8	H'FC0D	HCAN	16
Message data 11[7]	MD11[7]	8	H'FC0E	HCAN	16
Message data 11[8]	MD11[8]	8	H'FC0F	HCAN	16
Message data 12[1]	MD12[1]	8	H'FC10	HCAN	16
Message data 12[2]	MD12[2]	8	H'FC11	HCAN	16
Message data 12[3]	MD12[3]	8	H'FC12	HCAN	16
Message data 12[4]	MD12[4]	8	H'FC13	HCAN	16
Message data 12[5]	MD12[5]	8	H'FC14	HCAN	16
Message data 12[6]	MD12[6]	8	H'FC15	HCAN	16
Message data 12[7]	MD12[7]	8	H'FC16	HCAN	16
Message data 12[8]	MD12[8]	8	H'FC17	HCAN	16
Message data 13[1]	MD13[1]	8	H'FC18	HCAN	16
Message data 13[2]	MD13[2]	8	H'FC19	HCAN	16
Message data 13[3]	MD13[3]	8	H'FC1A	HCAN	16
Message data 13[4]	MD13[4]	8	H'FC1B	HCAN	16
Message data 13[5]	MD13[5]	8	H'FC1C	HCAN	16
Message data 13[6]	MD13[6]	8	H'FC1D	HCAN	16
Message data 13[7]	MD13[7]	8	H'FC1E	HCAN	16
Message data 13[8]	MD13[8]	8	H'FC1F	HCAN	16
Message data 14[1]	MD14[1]	8	H'FC20	HCAN	16
Message data 14[2]	MD14[2]	8	H'FC21	HCAN	16
Message data 14[3]	MD14[3]	8	H'FC22	HCAN	16

Message data 15[2]	MD15[2]	8	H'FC29	HCAN	16
Message data 15[3]	MD15[3]	8	H'FC2A	HCAN	16
Message data 15[4]	MD15[4]	8	H'FC2B	HCAN	16
Message data 15[5]	MD15[5]	8	H'FC2C	HCAN	16
Message data 15[6]	MD15[6]	8	H'FC2D	HCAN	16
Message data 15[7]	MD15[7]	8	H'FC2E	HCAN	16
Message data 15[8]	MD15[8]	8	H'FC2F	HCAN	16
D/A data register 0	DADR0	8	H'FDAC	D/A converter	8
D/A data register 1	DADR1	8	H'FDAD	D/A converter	8
D/A control register	DACR	8	H'FDAE	D/A converter	8
Timer control register_2	TCR_2	8	H'FDC0	TMR_2	8/16
Timer control register_3	TCR_3	8	H'FDC1	TMR_3	8/16
Timer control/status register_2	TCSR_2	8	H'FDC2	TMR_2	8/16
Timer control/status register_3	TCSR_3	8	H'FDC3	TMR_3	8/16
Time constant register A_2	TCORA_2	8	H'FDC4	TMR_2	8/16
Time constant register A_3	TCORA_3	8	H'FDC5	TMR_3	8/16
Time constant register B_2	TCORB_2	8	H'FDC6	TMR_2	8/16
Time constant register B_3	TCORB_3	8	H'FDC7	TMR_3	8/16
Timer counter_2	TCNT_2	8	H'FDC8	TMR_2	8/16
Timer counter_3	TCNT_3	8	H'FDC9	TMR_3	8/16
Serial mode register_3	SMR_3	8	H'FDD0	SCI_3	8

Serial mode register_4	SMR_4	8	H'FDD8	SCI_4	8
Bit rate register_4	BRR_4	8	H'FDD9	SCI_4	8
Serial control register_4	SCR_4	8	H'FDDA	SCI_4	8
Transmit data register_4	TDR_4	8	H'FDDB	SCI_4	8
Serial status register_4	SSR_4	8	H'FDDC	SCI_4	8
Receive data register_4	RDR_4	8	H'FDDD	SCI_4	8
Smart card mode register_4	SCMR_4	8	H'FDDE	SCI_4	8
IC power control register	ICPCR	8	H'FDE1	PORT	8
System control register 2	SYSCR2	8	H'FDE2	FLASH	8
Standby control register	SBYCR	8	H'FDE4	SYSTEM	8
System control register	SYSCR	8	H'FDE5	SYSTEM	8
System clock control register	SCKCR	8	H'FDE6	SYSTEM	8
Mode control register	MDCR	8	H'FDE7	SYSTEM	8
Module stop control register A	MSTPCRA	8	H'FDE8	SYSTEM	8
Module stop control register B	MSTPCRB	8	H'FDE9	SYSTEM	8
Module stop control register C	MSTPCRC	8	H'FDEA	SYSTEM	8
Pin function control register	PFCR	8	H'FDEB	BSC	8
Low power control register	LPWRCR	8	H'FDEC	SYSTEM	8
Break address register A	BARA	32	H'FE00	PBC	8/16
Break address register B	BARB	32	H'FE04	PBC	8/16
Break control register A	BCRA	8	H'FE08	PBC	8/16
Break control register B	BCRB	8	H'FE09	PBC	8/16
IRQ sense control register H	ISCRH	8	H'FE12	INT	8

DTC enable register D	DTCERD	8	H'FE19	DTC	8
DTC enable register E	DTCERE	8	H'FE1A	DTC	8
DTC enable register F	DTCERF	8	H'FE1B	DTC	8
DTC enable register G	DTCERG	8	H'FE1C	DTC	8
DTC enable register I	DTCERI	8	H'FE1E	DTC	8
DTC vector register	DTVECR	8	H'FE1F	DTC	8
Port 1 data direction register	P1DDR	8	H'FE30	PORT	8
Port 2 data direction register	P2DDR	8	H'FE31	PORT	8
Port 3 data direction register	P3DDR	8	H'FE32	PORT	8
Port 5 data direction register	P5DDR	8	H'FE34	PORT	8
Port 7 data direction register	P7DDR	8	H'FE36	PORT	8
Port A data direction register	PADDR	8	H'FE39	PORT	8
Port B data direction register	PBDDR	8	H'FE3A	PORT	8
Port C data direction register	PCDDR	8	H'FE3B	PORT	8
Port D data direction register	PDDDR	8	H'FE3C	PORT	8
Port E data direction register	PEDDR	8	H'FE3D	PORT	8
Port F data direction register	PFDDR	8	H'FE3E	PORT	8
Port G data direction register	PGDDR	8	H'FE3F	PORT	8
Port A pull-up MOS control register	PAPCR	8	H'FE40	PORT	8
Port B pull-up MOS control register	PBPCR	8	H'FE41	PORT	8
Port C pull-up MOS control register	PCPCR	8	H'FE42	PORT	8
Port D pull-up MOS control register	PDPCR	8	H'FE43	PORT	8
Port E pull-up MOS control register	PEPCR	8	H'FE44	PORT	8

Timer interrupt enable register _3	TIER_3	8	H'FE84	TPU_3	8/16
Timer status register_3	TSR_3	8	H'FE85	TPU_3	8/16
Timer counter_3	TCNT_3	16	H'FE86	TPU_3	16
Timer general register A_3	TGRA_3	16	H'FE88	TPU_3	16
Timer general register B_3	TGRB_3	16	H'FE8A	TPU_3	16
Timer general register C_3	TGRC_3	16	H'FE8C	TPU_3	16
Timer general register D_3	TGRD_3	16	H'FE8E	TPU_3	16
Timer control register_4	TCR_4	8	H'FE90	TPU_4	8/16
Timer mode register_4	TMDR_4	8	H'FE91	TPU_4	8/16
Timer I/O control register_4	TIOR_4	8	H'FE92	TPU_4	8/16
Timer interrupt enable register_4	TIER_4	8	H'FE94	TPU_4	8/16
Timer status register_4	TSR_4	8	H'FE95	TPU_4	8/16
Timer counter_4	TCNT_4	16	H'FE96	TPU_4	16
Timer general register A_4	TGRA_4	16	H'FE98	TPU_4	16
Timer general register B_4	TGRB_4	16	H'FE9A	TPU_4	16
Timer control register_5	TCR_5	8	H'FEA0	TPU_5	8/16
Timer mode register_5	TMDR_5	8	H'FEA1	TPU_5	8/16
Timer I/O control register_5	TIOR_5	8	H'FEA2	TPU_5	8/16
Timer interrupt enable register_5	TIER_5	8	H'FEA4	TPU_5	8/16
Timer status register_5	TSR_5	8	H'FEA5	TPU_5	8/16
Timer counter_5	TCNT_5	16	H'FEA6	TPU_5	16
Timer general register A_5	TGRA_5	16	H'FEA8	TPU_5	16
Timer general register B_5	TGRB_5	16	H'FEAA	TPU_5	16

Interrupt priority register E	IPRE	8	H'FEC4	INT	8
Interrupt priority register F	IPRF	8	H'FEC5	INT	8
Interrupt priority register G	IPRG	8	H'FEC6	INT	8
Interrupt priority register H	IPRH	8	H'FEC7	INT	8
Interrupt priority register I	IPRI	8	H'FEC8	INT	8
Interrupt priority register J	IPRJ	8	H'FEC9	INT	8
Interrupt priority register K	IPRK	8	H'FECA	INT	8
Interrupt priority register L	IPRL	8	H'FECB	INT	8
Interrupt priority register M	IPRM	8	H'FECC	INT	8
Interrupt priority register O	IPRO	8	H'FECE	INT	8
Bus width control register	ABWCR	8	H'FED0	BSC	8
Access state control register	ASTCR	8	H'FED1	BSC	8
Wait control register H	WCRH	8	H'FED2	BSC	8
Wait control register L	WCRL	8	H'FED3	BSC	8
Bus control register H	BCRH	8	H'FED4	BSC	8
Bus control register L	BCRL	8	H'FED5	BSC	8
RAM emulation register	RAMER	8	H'FEDB	FLASH	8
Port 1 data register	P1DR	8	H'FF00	PORT	8
Port 2 data register	P2DR	8	H'FF01	PORT	8
Port 3 data register	P3DR	8	H'FF02	PORT	8
Port 5 data register	P5DR	8	H'FF04	PORT	8
Port 7 data register	P7DR	8	H'FF06	PORT	8
Port A data register	PADR	8	H'FF09	PORT	8

Timer control register_0	TCR_0	8	H'FF10	TPU_0	8/16
Timer mode register_0	TMDR_0	8	H'FF11	TPU_0	8/16
Timer I/O control register H_0	TIORH_0	8	H'FF12	TPU_0	8/16
Timer I/O control register L_0	TIORL_0	8	H'FF13	TPU_0	8/16
Timer interrupt enable register_0	TIER_0	8	H'FF14	TPU_0	8/16
Timer status register_0	TSR_0	8	H'FF15	TPU_0	8/16
Timer counter_0	TCNT_0	16	H'FF16	TPU_0	16
Timer general register A_0	TGRA_0	16	H'FF18	TPU_0	16
Timer general register B_0	TGRB_0	16	H'FF1A	TPU_0	16
Timer general register C_0	TGRC_0	16	H'FF1C	TPU_0	16
Timer general register D_0	TGRD_0	16	H'FF1E	TPU_0	16
Timer control register_1	TCR_1	8	H'FF20	TPU_1	8/16
Timer mode register_1	TMDR_1	8	H'FF21	TPU_1	8/16
Timer I/O control register_1	TIOR_1	8	H'FF22	TPU_1	8/16
Timer interrupt enable register_1	TIER_1	8	H'FF24	TPU_1	8/16
Timer status register_1	TSR_1	8	H'FF25	TPU_1	8/16
Timer counter_1	TCNT_1	16	H'FF26	TPU_1	16
Timer general register A_1	TGRA_1	16	H'FF28	TPU_1	16
Timer general register B_1	TGRB_1	16	H'FF2A	TPU_1	16
Timer control register_2	TCR_2	8	H'FF30	TPU_2	8/16
Timer mode register_2	TMDR_2	8	H'FF31	TPU_2	8/16
Timer I/O control register_2	TIOR_2	8	H'FF32	TPU_2	8/16
Timer interrupt enable register_2	TIER_2	8	H'FF34	TPU_2	8/16

Timer control/status register_0	TCSR_0	8	H'FF6A	TMR_0	8/16
Timer control/status register_1	TCSR_1	8	H'FF6B	TMR_1	8/16
Timer constant register A_0	TCORA_0	8	H'FF6C	TMR_0	8/16
Timer constant register A_1	TCORA_1	8	H'FF6D	TMR_1	8/16
Timer constant register B_0	TCORB_0	8	H'FF6E	TMR_0	8/16
Timer constant register B_1	TCORB_1	8	H'FF6F	TMR_1	8/16
Timer counter_0	TCNT_0	8	H'FF70	TMR_0	8/16
Timer counter_1	TCNT_1	8	H'FF71	TMR_1	8/16
Timer control/status register_0	TCSR_0	8	H'FF74	WDT_0	16
Timer counter_0	TCNT_0	8	H'FF74 (write)	WDT_0	16
Timer counter_0	TCNT_0	8	H'FF75 (read)	WDT_0	16
Reset control/status register	RSTCSR	8	H'FF76 (write)	WDT_0	16
Reset control/status register	RSTCSR	8	H'FF77 (read)	WDT_0	16
Serial mode register_0	SMR_0	8	H'FF78	SCI_0	8
Bit rate register_0	BRR_0	8	H'FF79	SCI_0	8
Serial control register_0	SCR_0	8	H'FF7A	SCI_0	8
Transmit data register_0	TDR_0	8	H'FF7B	SCI_0	8
Serial status register_0	SSR_0	8	H'FF7C	SCI_0	8
Receive data register_0	RDR_0	8	H'FF7D	SCI_0	8
Smart card mode register_0	SCMR_0	8	H'FF7E	SCI_0	8

Smart card mode register_1	SCMR_1	8	H'FF86	SCI_1	8
Serial mode register_2	SMR_2	8	H'FF88	SCI_2	8
Bit rate register_2	BRR_2	8	H'FF89	SCI_2	8
Serial control register_2	SCR_2	8	H'FF8A	SCI_2	8
Transmit data register_2	TDR_2	8	H'FF8B	SCI_2	8
Serial status register_2	SSR_2	8	H'FF8C	SCI_2	8
Receive data register_2	RDR_2	8	H'FF8D	SCI_2	8
Smart card mode register_2	SCMR_2	8	H'FF8E	SCI_2	8
A/D data register AH	ADDRAH	8	H'FF90	A/D	8
A/D data register AL	ADDRAL	8	H'FF91	A/D	8
A/D data register BH	ADDRBH	8	H'FF92	A/D	8
A/D data register BL	ADDRBL	8	H'FF93	A/D	8
A/D data register CH	ADDRCH	8	H'FF94	A/D	8
A/D data register CL	ADDRCL	8	H'FF95	A/D	8
A/D data register DH	ADDRDH	8	H'FF96	A/D	8
A/D data register DL	ADDRDL	8	H'FF97	A/D	8
A/D control/status register	ADCSR	8	H'FF98	A/D	8
A/D control register	ADCR	8	H'FF99	A/D	8
Timer control/status register_1	TCSR_1	8	H'FFA2	WDT_1	16
Timer counter_1	TCNT_1	8	H'FFA2 (write)	WDT_1	16
Timer counter_1	TCNT_1	8	H'FFA3 (read)	WDT_1	16
Flash code control status register	FCCS	8	H'FFA4	FLASH	8

Flash vector address code control register	FVACR	8	H'FFAB	FLASH	8
Flash vector address data register R	FVADRR	8	H'FFAC	FLASH	8
Flash vector address data register E	FVADRE	8	H'FFAD	FLASH	8
Flash vector address data register H	FVADRH	8	H'FFAE	FLASH	8
Flash vector address data register L	FVADRL	8	H'FFAF	FLASH	8
Port 1 register	PORT1	8	H'FFB0	PORT	8
Port 2 register	PORT2	8	H'FFB1	PORT	8
Port 3 register	PORT3	8	H'FFB2	PORT	8
Port 4 register	PORT4	8	H'FFB3	PORT	8
Port 5 register	PORT5	8	H'FFB4	PORT	8
Port 7 register	PORT7	8	H'FFB6	PORT	8
Port 9 register	PORT9	8	H'FFB8	PORT	8
Port A register	PORTA	8	H'FFB9	PORT	8
Port B register	PORTB	8	H'FFBA	PORT	8
Port C register	PORTC	8	H'FFBB	PORT	8
Port D register	PORTD	8	H'FFBC	PORT	8
Port E register	PORTE	8	H'FFBD	PORT	8
Port F register	PORTF	8	H'FFBE	PORT	8
Port G register	PORTG	8	H'FFBF	PORT	8

Notes: 1. Lower 16 bits of the address.

2. Allocated on the on-chip RAM. 32-bit bus when DTC accesses as register inf and 16-bit in other cases.

	Bit7	Bit6	Bit5	Bit4	Bit3	Bit2	Bit1	Bit0	
MRB	CHNE	DISEL	—	—	—	—	—	—	
DAR	Bit23	Bit22	Bit21	Bit20	Bit19	Bit18	Bit17	Bit16	
	Bit15	Bit14	Bit13	Bit12	Bit11	Bit10	Bit9	Bit8	
	Bit7	Bit6	Bit5	Bit4	Bit3	Bit2	Bit1	Bit0	
CRA	Bit15	Bit14	Bit13	Bit12	Bit11	Bit10	Bit9	Bit8	
	Bit7	Bit6	Bit5	Bit4	Bit3	Bit2	Bit1	Bit0	
CRB	Bit15	Bit14	Bit13	Bit12	Bit11	Bit10	Bit9	Bit8	
	Bit7	Bit6	Bit5	Bit4	Bit3	Bit2	Bit1	Bit0	
IECTR	IEE	IOL	DEE	CKS1	RE	LUEE	CKS0	—	IE
IECMR	—	—	—	—	—	CMD2	CMD1	CMD0	
IEMCR	SS	RN2	RN1	RN0	CTL3	CTL2	CTL1	CTL0	
IEAR1	IAR3	IAR2	IAR1	IAR0	IMD1	IMD0	—	STE	
IEAR2	IAR11	IAR10	IAR9	IAR8	IAR7	IAR6	IAR5	IAR4	
IESA1	ISA3	ISA2	ISA1	ISA0	—	—	—	—	
IESA2	ISA11	ISA10	ISA9	ISA8	ISA7	ISA6	ISA5	ISA4	
IETBFL	TBFL7	TBFL6	TBFL5	TBFL4	TBFL3	TBFL2	TBFL1	TBFL0	
IETBR	TBR7	TBR6	TBR5	TBR4	TBR3	TBR2	TBR1	TBR0	
IEMA1	IMA3	IMA2	IMA1	IMA0	—	—	—	—	
IEMA2	IMA11	IMA10	IMA9	IMA8	IMA7	IMA6	IMA5	IMA4	
IERCTL	—	—	—	—	RCTL3	RCTL2	RCTL1	RCTL0	
IERBFL	RBFL7	RBFL6	RBFL5	RBFL4	RBFL3	RBFL2	RBFL1	RBFL0	
IERBR	RBR7	RBR6	RBR5	RBR4	RBR3	RBR2	RBR1	RBR0	

IEREF	—	—	—	—	OVE	RTME	DLE	PE	
PHDDR	PH7DDR	PH6DDR	PH5DDR	PH4DDR	PH3DDR	PH2DDR	PH1DDR	PH0DDR	P
PJDDR	PJ7DDR	PJ6DDR	PJ5DDR	PJ4DDR	PJ3DDR	PJ2DDR	PJ1DDR	PJ0DDR	
PHDR	PH7DR	PH6DR	PH5DR	PH4DR	PH3DR	PH2DR	PH1DR	PH0DR	
PJDR	PJ7DR	PJ6DR	PJ5DR	PJ4DR	PJ3DR	PJ2DR	PJ1DR	PJ0DR	
PORTH	PH7	PH6	PH5	PH4	PH3	PH2	PH1	PH0	
PORTJ	PJ7	PJ6	PJ5	PJ4	PJ3	PJ2	PJ1	PJ0	
ICCR1_0	ICE	RCVD	MST	TRS	CKS3	CKS2	CKS1	CKS0	II
ICCR2_0	BBSY	SCP	SDAO	SDAOP	SCLO	—	IICRST	—	
ICMR_0	MLS	WAIT	—	—	BCWP	BC2	BC1	BC0	
ICIER_0	TIE	TEIE	RIE	NAKIE	STIE	ACKE	ACKBR	ACKBT	
ICSR_0	TDRE	TEND	RDRF	NACKF	STOP	AL/OVE	AAS	ADZ	
SAR_0	SVA6	SVA5	SVA4	SVA3	SVA2	SVA1	SVA0	FS	
ICDRT_0	ICDRT7	ICDRT6	ICDRT5	ICDRT4	ICDRT3	ICDRT2	ICDRT1	ICDRT0	
ICDRR_0	ICDRR7	ICDRR6	ICDRR5	ICDRR4	ICDRR3	ICDRR2	ICDRR1	ICDRR0	
ICCR1_1	ICE	RCVD	MST	TRS	CKS3	CKS2	CKS1	CKS0	II
ICCR2_1	BBSY	SCP	SDAO	SDAOP	SCLO	—	IICRST	—	
ICMR_1	MLS	WAIT	—	—	BCWP	BC2	BC1	BC0	
ICIER_1	TIE	TEIE	RIE	NAKIE	STIE	ACKE	ACKBR	ACKBT	
ICSR_1	TDRE	TEND	RDRF	NACKF	STOP	AL/OVE	AAS	ADZ	
SAR_1	SVA6	SVA5	SVA4	SVA3	SVA2	SVA1	SVA0	FS	
ICDRT_1	ICDRT7	ICDRT6	ICDRT5	ICDRT4	ICDRT3	ICDRT2	ICDRT1	ICDRT0	
ICDRR_1	ICDRR7	ICDRR6	ICDRR5	ICDRR4	ICDRR3	ICDRR2	ICDRR1	ICDRR0	

	TXPR15	TXPR14	TXPR13	TXPR12	TXPR11	TXPR10	TXPR9	TXPR8
TXCR	TXCR7	TXCR6	TXCR5	TXCR4	TXCR3	TXCR2	TXCR1	—
	TXCR15	TXCR14	TXCR13	TXCR12	TXCR11	TXCR10	TXCR9	TXCR8
TXACK	TXACK7	TXACK6	TXACK5	TXACK4	TXACK3	TXACK2	TXACK1	—
	TXACK15	TXACK14	TXACK13	TXACK12	TXACK11	TXACK10	TXACK9	TXACK8
ABACK	ABACK7	ABACK6	ABACK5	ABACK4	ABACK3	ABACK2	ABACK1	—
	ABACK15	ABACK14	ABACK13	ABACK12	ABACK11	ABACK10	ABACK9	ABACK8
RXPR	RXPR7	RXPR6	RXPR5	RXPR4	RXPR3	RXPR2	RXPR1	RXPR0
	RXPR15	RXPR14	RXPR13	RXPR12	RXPR11	RXPR10	RXPR9	RXPR8
RFPR	RFPR7	RFPR6	RFPR5	RFPR4	RFPR3	RFPR2	RFPR1	RFPR0
	RFPR15	RFPR14	RFPR13	RFPR12	RFPR11	RFPR10	RFPR9	RFPR8
IRR	IRR7	IRR6	IRR5	IRR4	IRR3	IRR2	IRR1	IRR0
	—	—	—	IRR12	—	—	IRR9	IRR8
MBIMR	MBIMR7	MBIMR6	MBIMR5	MBIMR4	MBIMR3	MBIMR2	MBIMR1	MBIMR0
	MBIMR15	MBIMR14	MBIMR13	MBIMR12	MBIMR11	MBIMR10	MBIMR9	MBIMR8
IMR	IMR7	IMR6	IMR5	IMR4	IMR3	IMR2	IMR1	—
	—	—	—	IMR12	—	—	IMR9	IMR8
REC	Bit7	Bit6	Bit5	Bit4	Bit3	Bit2	Bit1	Bit0
TEC	Bit7	Bit6	Bit5	Bit4	Bit3	Bit2	Bit1	Bit0
UMSR	UMSR7	UMSR6	UMSR5	UMSR4	UMSR3	UMSR2	UMSR1	UMSR0
	UMSR15	UMSR14	UMSR13	UMSR12	UMSR11	UMSR10	UMSR9	UMSR8
LAFML	LAFML7	LAFML6	LAFML5	LAFML4	LAFML3	LAFML2	LAFML1	LAFML0
	LAFML15	LAFML14	LAFML13	LAFML12	LAFML11	LAFML10	LAFML9	LAFML8

MC0[6]	ID-28	ID-27	ID-26	ID-25	ID-24	ID-23	ID-22	ID-21
MC0[7]	ID-7	ID-6	ID-5	ID-4	ID-3	ID-2	ID-1	ID-0
MC0[8]	ID-15	ID-14	ID-13	ID-12	ID-11	ID-10	ID-9	ID-8
MC1[1]	—	—	—	—	DLC3	DLC2	DLC1	DLC0
MC1[2]	—	—	—	—	—	—	—	—
MC1[3]	—	—	—	—	—	—	—	—
MC1[4]	—	—	—	—	—	—	—	—
MC1[5]	ID-20	ID-19	ID-18	RTR	IDE	—	ID-17	ID-16
MC1[6]	ID-28	ID-27	ID-26	ID-25	ID-24	ID-23	ID-22	ID-21
MC1[7]	ID-7	ID-6	ID-5	ID-4	ID-3	ID-2	ID-1	ID-0
MC1[8]	ID-15	ID-14	ID-13	ID-12	ID-11	ID-10	ID-9	ID-8
MC2[1]	—	—	—	—	DLC3	DLC2	DLC1	DLC0
MC2[2]	—	—	—	—	—	—	—	—
MC2[3]	—	—	—	—	—	—	—	—
MC2[4]	—	—	—	—	—	—	—	—
MC2[5]	ID-20	ID-19	ID-18	RTR	IDE	—	ID-17	ID-16
MC2[6]	ID-28	ID-27	ID-26	ID-25	ID-24	ID-23	ID-22	ID-21
MC2[7]	ID-7	ID-6	ID-5	ID-4	ID-3	ID-2	ID-1	ID-0
MC2[8]	ID-15	ID-14	ID-13	ID-12	ID-11	ID-10	ID-9	ID-8
MC3[1]	—	—	—	—	DLC3	DLC2	DLC1	DLC0
MC3[2]	—	—	—	—	—	—	—	—
MC3[3]	—	—	—	—	—	—	—	—
MC3[4]	—	—	—	—	—	—	—	—
MC3[5]	ID-20	ID-19	ID-18	RTR	IDE	—	ID-17	ID-16

MC4[5]	ID-20	ID-19	ID-18	RTR	IDE	—	ID-17	ID-16
MC4[6]	ID-28	ID-27	ID-26	ID-25	ID-24	ID-23	ID-22	ID-21
MC4[7]	ID-7	ID-6	ID-5	ID-4	ID-3	ID-2	ID-1	ID-0
MC4[8]	ID-15	ID-14	ID-13	ID-12	ID-11	ID-10	ID-9	ID-8
MC5[1]	—	—	—	—	DLC3	DLC2	DLC1	DLC0
MC5[2]	—	—	—	—	—	—	—	—
MC5[3]	—	—	—	—	—	—	—	—
MC5[4]	—	—	—	—	—	—	—	—
MC5[5]	ID-20	ID-19	ID-18	RTR	IDE	—	ID-17	ID-16
MC5[6]	ID-28	ID-27	ID-26	ID-25	ID-24	ID-23	ID-22	ID-21
MC5[7]	ID-7	ID-6	ID-5	ID-4	ID-3	ID-2	ID-1	ID-0
MC5[8]	ID-15	ID-14	ID-13	ID-12	ID-11	ID-10	ID-9	ID-8
MC6[1]	—	—	—	—	DLC3	DLC2	DLC1	DLC0
MC6[2]	—	—	—	—	—	—	—	—
MC6[3]	—	—	—	—	—	—	—	—
MC6[4]	—	—	—	—	—	—	—	—
MC6[5]	ID-20	ID-19	ID-18	RTR	IDE	—	ID-17	ID-16
MC6[6]	ID-28	ID-27	ID-26	ID-25	ID-24	ID-23	ID-22	ID-21
MC6[7]	ID-7	ID-6	ID-5	ID-4	ID-3	ID-2	ID-1	ID-0
MC6[8]	ID-15	ID-14	ID-13	ID-12	ID-11	ID-10	ID-9	ID-8
MC7[1]	—	—	—	—	DLC3	DLC2	DLC1	DLC0
MC7[2]	—	—	—	—	—	—	—	—
MC7[3]	—	—	—	—	—	—	—	—
MC7[4]	—	—	—	—	—	—	—	—

MC8[4]	—	—	—	—	—	—	—	—
MC8[5]	ID-20	ID-19	ID-18	RTR	IDE	—	ID-17	ID-16
MC8[6]	ID-28	ID-27	ID-26	ID-25	ID-24	ID-23	ID-22	ID-21
MC8[7]	ID-7	ID-6	ID-5	ID-4	ID-3	ID-2	ID-1	ID-0
MC8[8]	ID-15	ID-14	ID-13	ID-12	ID-11	ID-10	ID-9	ID-8
MC9[1]	—	—	—	—	DLC3	DLC2	DLC1	DLC0
MC9[2]	—	—	—	—	—	—	—	—
MC9[3]	—	—	—	—	—	—	—	—
MC9[4]	—	—	—	—	—	—	—	—
MC9[5]	ID-20	ID-19	ID-18	RTR	IDE	—	ID-17	ID-16
MC9[6]	ID-28	ID-27	ID-26	ID-25	ID-24	ID-23	ID-22	ID-21
MC9[7]	ID-7	ID-6	ID-5	ID-4	ID-3	ID-2	ID-1	ID-0
MC9[8]	ID-15	ID-14	ID-13	ID-12	ID-11	ID-10	ID-9	ID-8
MC10[1]	—	—	—	—	DLC3	DLC2	DLC1	DLC0
MC10[2]	—	—	—	—	—	—	—	—
MC10[3]	—	—	—	—	—	—	—	—
MC10[4]	—	—	—	—	—	—	—	—
MC10[5]	ID-20	ID-19	ID-18	RTR	IDE	—	ID-17	ID-16
MC10[6]	ID-28	ID-27	ID-26	ID-25	ID-24	ID-23	ID-22	ID-21
MC10[7]	ID-7	ID-6	ID-5	ID-4	ID-3	ID-2	ID-1	ID-0
MC10[8]	ID-15	ID-14	ID-13	ID-12	ID-11	ID-10	ID-9	ID-8
MC11[1]	—	—	—	—	DLC3	DLC2	DLC1	DLC0
MC11[2]	—	—	—	—	—	—	—	—
MC11[3]	—	—	—	—	—	—	—	—

MC12[3]	—	—	—	—	—	—	—	—
MC12[4]	—	—	—	—	—	—	—	—
MC12[5]	ID-20	ID-19	ID-18	RTR	IDE	—	ID-17	ID-16
MC12[6]	ID-28	ID-27	ID-26	ID-25	ID-24	ID-23	ID-22	ID-21
MC12[7]	ID-7	ID-6	ID-5	ID-4	ID-3	ID-2	ID-1	ID-0
MC12[8]	ID-15	ID-14	ID-13	ID-12	ID-11	ID-10	ID-9	ID-8
MC13[1]	—	—	—	—	DLC3	DLC2	DLC1	DLC0
MC13[2]	—	—	—	—	—	—	—	—
MC13[3]	—	—	—	—	—	—	—	—
MC13[4]	—	—	—	—	—	—	—	—
MC13[5]	ID-20	ID-19	ID-18	RTR	IDE	—	ID-17	ID-16
MC13[6]	ID-28	ID-27	ID-26	ID-25	ID-24	ID-23	ID-22	ID-21
MC13[7]	ID-7	ID-6	ID-5	ID-4	ID-3	ID-2	ID-1	ID-0
MC13[8]	ID-15	ID-14	ID-13	ID-12	ID-11	ID-10	ID-9	ID-8
MC14[1]	—	—	—	—	DLC3	DLC2	DLC1	DLC0
MC14[2]	—	—	—	—	—	—	—	—
MC14[3]	—	—	—	—	—	—	—	—
MC14[4]	—	—	—	—	—	—	—	—
MC14[5]	ID-20	ID-19	ID-18	RTR	IDE	—	ID-17	ID-16
MC14[6]	ID-28	ID-27	ID-26	ID-25	ID-24	ID-23	ID-22	ID-21
MC14[7]	ID-7	ID-6	ID-5	ID-4	ID-3	ID-2	ID-1	ID-0
MC14[8]	ID-15	ID-14	ID-13	ID-12	ID-11	ID-10	ID-9	ID-8
MC15[1]	—	—	—	—	DLC3	DLC2	DLC1	DLC0
MC15[2]	—	—	—	—	—	—	—	—

MD0[2]	Bit7	Bit6	Bit5	Bit4	Bit3	Bit2	Bit1	Bit0
MD0[3]	Bit7	Bit6	Bit5	Bit4	Bit3	Bit2	Bit1	Bit0
MD0[4]	Bit7	Bit6	Bit5	Bit4	Bit3	Bit2	Bit1	Bit0
MD0[5]	Bit7	Bit6	Bit5	Bit4	Bit3	Bit2	Bit1	Bit0
MD0[6]	Bit7	Bit6	Bit5	Bit4	Bit3	Bit2	Bit1	Bit0
MD0[7]	Bit7	Bit6	Bit5	Bit4	Bit3	Bit2	Bit1	Bit0
MD0[8]	Bit7	Bit6	Bit5	Bit4	Bit3	Bit2	Bit1	Bit0
MD1[1]	Bit7	Bit6	Bit5	Bit4	Bit3	Bit2	Bit1	Bit0
MD1[2]	Bit7	Bit6	Bit5	Bit4	Bit3	Bit2	Bit1	Bit0
MD1[3]	Bit7	Bit6	Bit5	Bit4	Bit3	Bit2	Bit1	Bit0
MD1[4]	Bit7	Bit6	Bit5	Bit4	Bit3	Bit2	Bit1	Bit0
MD1[5]	Bit7	Bit6	Bit5	Bit4	Bit3	Bit2	Bit1	Bit0
MD1[6]	Bit7	Bit6	Bit5	Bit4	Bit3	Bit2	Bit1	Bit0
MD1[7]	Bit7	Bit6	Bit5	Bit4	Bit3	Bit2	Bit1	Bit0
MD1[8]	Bit7	Bit6	Bit5	Bit4	Bit3	Bit2	Bit1	Bit0
MD2[1]	Bit7	Bit6	Bit5	Bit4	Bit3	Bit2	Bit1	Bit0
MD2[2]	Bit7	Bit6	Bit5	Bit4	Bit3	Bit2	Bit1	Bit0
MD2[3]	Bit7	Bit6	Bit5	Bit4	Bit3	Bit2	Bit1	Bit0
MD2[4]	Bit7	Bit6	Bit5	Bit4	Bit3	Bit2	Bit1	Bit0
MD2[5]	Bit7	Bit6	Bit5	Bit4	Bit3	Bit2	Bit1	Bit0
MD2[6]	Bit7	Bit6	Bit5	Bit4	Bit3	Bit2	Bit1	Bit0
MD2[7]	Bit7	Bit6	Bit5	Bit4	Bit3	Bit2	Bit1	Bit0
MD2[8]	Bit7	Bit6	Bit5	Bit4	Bit3	Bit2	Bit1	Bit0
MD3[1]	Bit7	Bit6	Bit5	Bit4	Bit3	Bit2	Bit1	Bit0

MD4[1]	Bit7	Bit6	Bit5	Bit4	Bit3	Bit2	Bit1	Bit0
MD4[2]	Bit7	Bit6	Bit5	Bit4	Bit3	Bit2	Bit1	Bit0
MD4[3]	Bit7	Bit6	Bit5	Bit4	Bit3	Bit2	Bit1	Bit0
MD4[4]	Bit7	Bit6	Bit5	Bit4	Bit3	Bit2	Bit1	Bit0
MD4[5]	Bit7	Bit6	Bit5	Bit4	Bit3	Bit2	Bit1	Bit0
MD4[6]	Bit7	Bit6	Bit5	Bit4	Bit3	Bit2	Bit1	Bit0
MD4[7]	Bit7	Bit6	Bit5	Bit4	Bit3	Bit2	Bit1	Bit0
MD4[8]	Bit7	Bit6	Bit5	Bit4	Bit3	Bit2	Bit1	Bit0
MD5[1]	Bit7	Bit6	Bit5	Bit4	Bit3	Bit2	Bit1	Bit0
MD5[2]	Bit7	Bit6	Bit5	Bit4	Bit3	Bit2	Bit1	Bit0
MD5[3]	Bit7	Bit6	Bit5	Bit4	Bit3	Bit2	Bit1	Bit0
MD5[4]	Bit7	Bit6	Bit5	Bit4	Bit3	Bit2	Bit1	Bit0
MD5[5]	Bit7	Bit6	Bit5	Bit4	Bit3	Bit2	Bit1	Bit0
MD5[6]	Bit7	Bit6	Bit5	Bit4	Bit3	Bit2	Bit1	Bit0
MD5[7]	Bit7	Bit6	Bit5	Bit4	Bit3	Bit2	Bit1	Bit0
MD5[8]	Bit7	Bit6	Bit5	Bit4	Bit3	Bit2	Bit1	Bit0
MD6[1]	Bit7	Bit6	Bit5	Bit4	Bit3	Bit2	Bit1	Bit0
MD6[2]	Bit7	Bit6	Bit5	Bit4	Bit3	Bit2	Bit1	Bit0
MD6[3]	Bit7	Bit6	Bit5	Bit4	Bit3	Bit2	Bit1	Bit0
MD6[4]	Bit7	Bit6	Bit5	Bit4	Bit3	Bit2	Bit1	Bit0
MD6[5]	Bit7	Bit6	Bit5	Bit4	Bit3	Bit2	Bit1	Bit0
MD6[6]	Bit7	Bit6	Bit5	Bit4	Bit3	Bit2	Bit1	Bit0
MD6[7]	Bit7	Bit6	Bit5	Bit4	Bit3	Bit2	Bit1	Bit0
MD6[8]	Bit7	Bit6	Bit5	Bit4	Bit3	Bit2	Bit1	Bit0

MD7[8]	Bit7	Bit6	Bit5	Bit4	Bit3	Bit2	Bit1	Bit0
MD8[1]	Bit7	Bit6	Bit5	Bit4	Bit3	Bit2	Bit1	Bit0
MD8[2]	Bit7	Bit6	Bit5	Bit4	Bit3	Bit2	Bit1	Bit0
MD8[3]	Bit7	Bit6	Bit5	Bit4	Bit3	Bit2	Bit1	Bit0
MD8[4]	Bit7	Bit6	Bit5	Bit4	Bit3	Bit2	Bit1	Bit0
MD8[5]	Bit7	Bit6	Bit5	Bit4	Bit3	Bit2	Bit1	Bit0
MD8[6]	Bit7	Bit6	Bit5	Bit4	Bit3	Bit2	Bit1	Bit0
MD8[7]	Bit7	Bit6	Bit5	Bit4	Bit3	Bit2	Bit1	Bit0
MD8[8]	Bit7	Bit6	Bit5	Bit4	Bit3	Bit2	Bit1	Bit0
MD9[1]	Bit7	Bit6	Bit5	Bit4	Bit3	Bit2	Bit1	Bit0
MD9[2]	Bit7	Bit6	Bit5	Bit4	Bit3	Bit2	Bit1	Bit0
MD9[3]	Bit7	Bit6	Bit5	Bit4	Bit3	Bit2	Bit1	Bit0
MD9[4]	Bit7	Bit6	Bit5	Bit4	Bit3	Bit2	Bit1	Bit0
MD9[5]	Bit7	Bit6	Bit5	Bit4	Bit3	Bit2	Bit1	Bit0
MD9[6]	Bit7	Bit6	Bit5	Bit4	Bit3	Bit2	Bit1	Bit0
MD9[7]	Bit7	Bit6	Bit5	Bit4	Bit3	Bit2	Bit1	Bit0
MD9[8]	Bit7	Bit6	Bit5	Bit4	Bit3	Bit2	Bit1	Bit0
MD10[1]	Bit7	Bit6	Bit5	Bit4	Bit3	Bit2	Bit1	Bit0
MD10[2]	Bit7	Bit6	Bit5	Bit4	Bit3	Bit2	Bit1	Bit0
MD10[3]	Bit7	Bit6	Bit5	Bit4	Bit3	Bit2	Bit1	Bit0
MD10[4]	Bit7	Bit6	Bit5	Bit4	Bit3	Bit2	Bit1	Bit0
MD10[5]	Bit7	Bit6	Bit5	Bit4	Bit3	Bit2	Bit1	Bit0
MD10[6]	Bit7	Bit6	Bit5	Bit4	Bit3	Bit2	Bit1	Bit0
MD10[7]	Bit7	Bit6	Bit5	Bit4	Bit3	Bit2	Bit1	Bit0

MD11[8]	Bit7	Bit6	Bit5	Bit4	Bit3	Bit2	Bit1	Bit0
MD12[1]	Bit7	Bit6	Bit5	Bit4	Bit3	Bit2	Bit1	Bit0
MD12[2]	Bit7	Bit6	Bit5	Bit4	Bit3	Bit2	Bit1	Bit0
MD12[3]	Bit7	Bit6	Bit5	Bit4	Bit3	Bit2	Bit1	Bit0
MD12[4]	Bit7	Bit6	Bit5	Bit4	Bit3	Bit2	Bit1	Bit0
MD12[5]	Bit7	Bit6	Bit5	Bit4	Bit3	Bit2	Bit1	Bit0
MD12[6]	Bit7	Bit6	Bit5	Bit4	Bit3	Bit2	Bit1	Bit0
MD12[7]	Bit7	Bit6	Bit5	Bit4	Bit3	Bit2	Bit1	Bit0
MD12[8]	Bit7	Bit6	Bit5	Bit4	Bit3	Bit2	Bit1	Bit0
MD13[1]	Bit7	Bit6	Bit5	Bit4	Bit3	Bit2	Bit1	Bit0
MD13[2]	Bit7	Bit6	Bit5	Bit4	Bit3	Bit2	Bit1	Bit0
MD13[3]	Bit7	Bit6	Bit5	Bit4	Bit3	Bit2	Bit1	Bit0
MD13[4]	Bit7	Bit6	Bit5	Bit4	Bit3	Bit2	Bit1	Bit0
MD13[5]	Bit7	Bit6	Bit5	Bit4	Bit3	Bit2	Bit1	Bit0
MD13[6]	Bit7	Bit6	Bit5	Bit4	Bit3	Bit2	Bit1	Bit0
MD13[7]	Bit7	Bit6	Bit5	Bit4	Bit3	Bit2	Bit1	Bit0
MD13[8]	Bit7	Bit6	Bit5	Bit4	Bit3	Bit2	Bit1	Bit0
MD14[1]	Bit7	Bit6	Bit5	Bit4	Bit3	Bit2	Bit1	Bit0
MD14[2]	Bit7	Bit6	Bit5	Bit4	Bit3	Bit2	Bit1	Bit0
MD14[3]	Bit7	Bit6	Bit5	Bit4	Bit3	Bit2	Bit1	Bit0
MD14[4]	Bit7	Bit6	Bit5	Bit4	Bit3	Bit2	Bit1	Bit0
MD14[5]	Bit7	Bit6	Bit5	Bit4	Bit3	Bit2	Bit1	Bit0
MD14[6]	Bit7	Bit6	Bit5	Bit4	Bit3	Bit2	Bit1	Bit0

MD15[6]	Bit7	Bit6	Bit5	Bit4	Bit3	Bit2	Bit1	Bit0	
MD15[7]	Bit7	Bit6	Bit5	Bit4	Bit3	Bit2	Bit1	Bit0	
MD15[8]	Bit7	Bit6	Bit5	Bit4	Bit3	Bit2	Bit1	Bit0	
DADR0	Bit7	Bit6	Bit5	Bit4	Bit3	Bit2	Bit1	Bit0	D
DADR1	Bit7	Bit6	Bit5	Bit4	Bit3	Bit2	Bit1	Bit0	c
DACR	DAOE1	DAOE0	DAE	—	—	—	—	—	
TCR_2	CMIEB	CMIEA	OVIE	CCLR1	CCLR0	CKS2	CKS1	CKS0	T
TCR_3	CMIEB	CMIEA	OVIE	CCLR1	CCLR0	CKS2	CKS1	CKS0	T
TCSR_2	CMFB	CMFA	OVF	—	OS3	OS2	OS1	OS0	T
TCSR_3	CMFB	CMFA	OVF	—	OS3	OS2	OS1	OS0	T
TCORA_2	Bit7	Bit6	Bit5	Bit4	Bit3	Bit2	Bit1	Bit0	T
TCORA_3	Bit7	Bit6	Bit5	Bit4	Bit3	Bit2	Bit1	Bit0	T
TCORB_2	Bit7	Bit6	Bit5	Bit4	Bit3	Bit2	Bit1	Bit0	T
TCORB_3	Bit7	Bit6	Bit5	Bit4	Bit3	Bit2	Bit1	Bit0	T
TCNT_2	Bit7	Bit6	Bit5	Bit4	Bit3	Bit2	Bit1	Bit0	T
TCNT_3	Bit7	Bit6	Bit5	Bit4	Bit3	Bit2	Bit1	Bit0	T
SMR_3*[1]	C/$\overline{A}$ (GM)	CHR (BLK)	PE (PE)	O/$\overline{E}$ (O/$\overline{E}$)	STOP (BCP1)	MP (BCP0)	CKS1 (CKS1)	CKS0 (CKS0)	S
BRR_3	Bit7	Bit6	Bit5	Bit4	Bit3	Bit2	Bit1	Bit0	
SCR_3	TIE	RIE	TE	RE	MPIE	TEIE	CKE1	CKE0	
TDR_3	Bit7	Bit6	Bit5	Bit4	Bit3	Bit2	Bit1	Bit0	
SSR_3*[1]	TDRE (TDRE)	RDRF (RDRF)	ORER (ORER)	FER (ERS)	PER (PER)	TEND (TEND)	MPB (MPB)	MPBT (MPBT)	
RDR_3	Bit7	Bit6	Bit5	Bit4	Bit3	Bit2	Bit1	Bit0	

RDR_4	Bit7	Bit6	Bit5	Bit4	Bit3	Bit2	Bit1	Bit0	
SCMR_4	—	—	—	—	SDIR	SINV	—	SMIF	
ICPCR	—	—	—	—	BUFGC2	BUFGC1	—	—	PC
SYSCR2	—	—	—	—	FLSHE	—	—	—	FL
SBYCR	SSBY	STS2	STS1	STS0	OPE	—	—	—	SY
SYSCR	—	—	INTM1	INTM0	NMIEG	MRESE	—	RAME	
SCKCR	PSTOP	—	—	—	STCS	SCK2	SCK1	SCK0	
MDCR	—	—	—	—	—	MDS2	MDS1	MDS0	
MSTPCRA	MSTPA7	MSTPA6	MSTPA5	MSTPA4	MSTPA3	MSTPA2	MSTPA1	MSTPA0	
MSTPCRB	MSTPB7	MSTPB6	MSTPB5	MSTPB4	MSTPB3	MSTPB2	MSTPB1	MSTPB0	
MSTPCRC	MSTPC7	MSTPC6	MSTPC5	MSTPC4	MSTPC3	MSTPC2	MSTPC1	MSTPC0	
PFCR	—	—	BUZZE	—	AE3	AE2	AE1	AE0	BS
LPWRCR	DTON	LSON	NESEL	SUBSTP	RFCUT	—	STC1	STC0	SY
BARA	—	—	—	—	—	—	—	—	PB
	BAA23	BAA22	BAA21	BAA20	BAA19	BAA18	BAA17	BAA16	
	BAA15	BAA14	BAA13	BAA12	BAA11	BAA10	BAA9	BAA8	
	BAA7	BAA6	BAA5	BAA4	BAA3	BAA2	BAA1	BAA0	
BARB	—	—	—	—	—	—	—	—	
	BAB23	BAB22	BAB21	BAB20	BAB19	BAB18	BAB17	BAB16	
	BAB15	BAB14	BAB13	BAB12	BAB11	BAB10	BAB9	BAB8	
	BAB7	BAB6	BAB5	BAB4	BAB3	BAB2	BAB1	BAB0	
BCRA	CMFA	CDA	BAMRA2	BAMRA1	BAMRA0	CSELA1	CSELA0	BIEA	
BCRB	CMFB	CDB	BAMRB2	BAMRB1	BAMRB0	CSELB1	CSELB0	BIEB	

DTCERD	—	—	DTCED5	DTCED4	DTCED3	DTCED2	DTCED1	DTCED0
DTCERE	—	—	—	—	DTCEE3	DTCEE2	DTCEE1	DTCEE0
DTCERF	DTCEF7	DTCEF6	DTCEF5	DTCEF4	DTCEF3	DTCEF2	DTCEF1	DTCEF0
DTCERG	—	DTCEG6	DTCEG5	—	DTCEG3	DTCEG2	—	—
DTCERI	DTCEI7	DTCEI6	DTCEI5	DTCEI4	—	—	—	—
DTVECR	SWDTE	DTVEC6	DTVEC5	DTVEC4	DTVEC3	DTVEC2	DTVEC1	DTVEC0
P1DDR	P17DDR	P16DDR	P15DDR	P14DDR	P13DDR	P12DDR	P11DDR	P10DDR
P2DDR	P27DDR	P26DDR	P25DDR	P24DDR	P23DDR	P22DDR	P21DDR	P20DDR
P3DDR	P37DDR	P36DDR	P35DDR	P34DDR	P33DDR	P32DDR	P31DDR	P30DDR
P5DDR	—	—	—	—	—	P52DDR	P51DDR	P50DDR
P7DDR	P77DDR	P76DDR	P75DDR	P74DDR	P73DDR	P72DDR	P71DDR	P70DDR
PADDR	PA7DDR	PA6DDR	PA5DDR	PA4DDR	PA3DDR	PA2DDR	PA1DDR	PA0DDR
PBDDR	PB7DDR	PB6DDR	PB5DDR	PB4DDR	PB3DDR	PB2DDR	PB1DDR	PB0DDR
PCDDR	PC7DDR	PC6DDR	PC5DDR	PC4DDR	PC3DDR	PC2DDR	PC1DDR	PC0DDR
PDDDR	PD7DDR	PD6DDR	PD5DDR	PD4DDR	PD3DDR	PD2DDR	PD1DDR	PD0DDR
PEDDR	PE7DDR	PE6DDR	PE5DDR	PE4DDR	PE3DDR	PE2DDR	PE1DDR	PE0DDR
PFDDR	PF7DDR	PF6DDR	PF5DDR	PF4DDR	PF3DDR*[2]	PF2DDR*[2]	PF1DDR	PF0DDR
PGDDR	—	—	—	PG4DDR	PG3DDR	PG2DDR	PG1DDR	PG0DDR
PAPCR	PA7PCR	PA6PCR	PA5PCR	PA4PCR	PA3PCR	PA2PCR	PA1PCR	PA0PCR
PBPCR	PB7PCR	PB6PCR	PB5PCR	PB4PCR	PB3PCR	PB2PCR	PB1PCR	PB0PCR
PCPCR	PC7PCR	PC6PCR	PC5PCR	PC4PCR	PC3PCR	PC2PCR	PC1PCR	PC0PCR
PDPCR	PD7PCR	PD6PCR	PD5PCR	PD4PCR	PD3PCR	PD2PCR	PD1PCR	PD0PCR
PEPCR	PE7PCR	PE6PCR	PE5PCR	PE4PCR	PE3PCR	PE2PCR	PE1PCR	PE0PCR
P3ODR	P37ODR	P36ODR	P35ODR	P34ODR	P33ODR	P32ODR	P31ODR	P30ODR

TCNT_3	Bit15	Bit14	Bit13	Bit12	Bit11	Bit10	Bit9	Bit8	
	Bit7	Bit6	Bit5	Bit4	Bit3	Bit2	Bit1	Bit0	
TGRA_3	Bit15	Bit14	Bit13	Bit12	Bit11	Bit10	Bit9	Bit8	
	Bit7	Bit6	Bit5	Bit4	Bit3	Bit2	Bit1	Bit0	
TGRB_3	Bit15	Bit14	Bit13	Bit12	Bit11	Bit10	Bit9	Bit8	
	Bit7	Bit6	Bit5	Bit4	Bit3	Bit2	Bit1	Bit0	
TGRC_3	Bit15	Bit14	Bit13	Bit12	Bit11	Bit10	Bit9	Bit8	
	Bit7	Bit6	Bit5	Bit4	Bit3	Bit2	Bit1	Bit0	
TGRD_3	Bit15	Bit14	Bit13	Bit12	Bit11	Bit10	Bit9	Bit8	
	Bit7	Bit6	Bit5	Bit4	Bit3	Bit2	Bit1	Bit0	
TCR_4	—	CCLR1	CCLR0	CKEG1	CKEG0	TPSC2	TPSC1	TPSC0	TP
TMDR_4	—	—	—	—	MD3	MD2	MD1	MD0	
TIOR_4	IOB3	IOB2	IOB1	IOB0	IOA3	IOA2	IOA1	IOA0	
TIER_4	TTGE	—	TCIEU	TCIEV	—	—	TGIEB	TGIEA	
TSR_4	TCFD	—	TCFU	TCFV	—	—	TGFB	TGFA	
TCNT_4	Bit15	Bit14	Bit13	Bit12	Bit11	Bit10	Bit9	Bit8	
	Bit7	Bit6	Bit5	Bit4	Bit3	Bit2	Bit1	Bit0	
TGRA_4	Bit15	Bit14	Bit13	Bit12	Bit11	Bit10	Bit9	Bit8	
	Bit7	Bit6	Bit5	Bit4	Bit3	Bit2	Bit1	Bit0	
TGRB_4	Bit15	Bit14	Bit13	Bit12	Bit11	Bit10	Bit9	Bit8	
	Bit7	Bit6	Bit5	Bit4	Bit3	Bit2	Bit1	Bit0	
TCR_5	—	CCLR1	CCLR0	CKEG1	CKEG0	TPSC2	TPSC1	TPSC0	TP
TMDR_5	—	—	—	—	MD3	MD2	MD1	MD0	
TIOR_5	IOB3	IOB2	IOB1	IOB0	IOA3	IOA2	IOA1	IOA0	

	Bit7	Bit6	Bit5	Bit4	Bit3	Bit2	Bit1	Bit0	
TSTR	—	—	CST5	CST4	CST3	CST2	CST1	CST0	T
TSYR	—	—	SYNC5	SYNC4	SYNC3	SYNC2	SYNC1	SYNC0	
IPRA	—	IPR6	IPR5	IPR4	—	IPR2	IPR1	IPR0	I
IPRB	—	IPR6	IPR5	IPR4	—	IPR2	IPR1	IPR0	
IPRC	—	IPR6	IPR5	IPR4	—	IPR2	IPR1	IPR0	
IPRD	—	IPR6	IPR5	IPR4	—	—	—	—	
IPRE	—	IPR6	IPR5	IPR4	—	IPR2	IPR1	IPR0	
IPRF	—	IPR6	IPR5	IPR4	—	IPR2	IPR1	IPR0	
IPRG	—	IPR6	IPR5	IPR4	—	IPR2	IPR1	IPR0	
IPRH	—	IPR6	IPR5	IPR4	—	IPR2	IPR1	IPR0	
IPRI	—	IPR6	IPR5	IPR4	—	IPR2	IPR1	IPR0	
IPRJ	—	—	—	—	—	IPR2	IPR1	IPR0	
IPRK	—	IPR6	IPR5	IPR4	—	IPR2	IPR1	IPR0	
IPRL	—	IPR6	IPR5	IPR4	—	IPR2	IPR1	IPR0	
IPRM	—	IPR6	IPR5	IPR4	—	IPR2	IPR1	IPR0	
IPRO	—	IPR6	IPR5	IPR4	—	IPR2	IPR1	IPR0	
ABWCR	ABW7	ABW6	ABW5	ABW4	ABW3	ABW2	ABW1	ABW0	B
ASTCR	AST7	AST6	AST5	AST4	AST3	AST2	AST1	AST0	
WCRH	W71	W70	W61	W60	W51	W50	W41	W40	
WCRL	W31	W30	W21	W20	W11	W10	W01	W00	
BCRH	ICIS1	ICIS0	BRSTRM	BRSTS1	BRSTS0	—	—	—	
BCRL	BRLE	—	—	—	—	—	—	WAITE	
RAMER	—	—	—	—	RAMS	RAM2	RAM1	RAM0	F

PDDR	PD7DR	PD6DR	PD5DR	PD4DR	PD3DR	PD2DR	PD1DR	PD0DR	
PEDR	PE7DR	PE6DR	PE5DR	PE4DR	PE3DR	PE2DR	PE1DR	PE0DR	
PFDR	PF7DR	PF6DR	PF5DR	PF4DR	PF3DR	PF2DR	PF1DR	PF0DR	
PGDR	—	—	—	PG4DR	PG3DR*[2]	PG2DR*[2]	PG1DR	PG0DR	
TCR_0	CCLR2	CCLR1	CCLR0	CKEG1	CKEG0	TPSC2	TPSC1	TPSC0	TP
TMDR_0	—	—	BFB	BFA	MD3	MD2	MD1	MD0	
TIORH_0	IOB3	IOB2	IOB1	IOB0	IOA3	IOA2	IOA1	IOA0	
TIORL_0	IOD3	IOD2	IOD1	IOD0	IOC3	IOC2	IOC1	IOC0	
TIER_0	TTGE	—	—	TCIEV	TGIED	TGIEC	TGIEB	TGIEA	
TSR_0	—	—	—	TCFV	TGFD	TGFC	TGFB	TGFA	
TCNT_0	Bit15	Bit14	Bit13	Bit12	Bit11	Bit10	Bit9	Bit8	
	Bit7	Bit6	Bit5	Bit4	Bit3	Bit2	Bit1	Bit0	
TGRA_0	Bit15	Bit14	Bit13	Bit12	Bit11	Bit10	Bit9	Bit8	
	Bit7	Bit6	Bit5	Bit4	Bit3	Bit2	Bit1	Bit0	
TGRB_0	Bit15	Bit14	Bit13	Bit12	Bit11	Bit10	Bit9	Bit8	
	Bit7	Bit6	Bit5	Bit4	Bit3	Bit2	Bit1	Bit0	
TGRC_0	Bit15	Bit14	Bit13	Bit12	Bit11	Bit10	Bit9	Bit8	
	Bit7	Bit6	Bit5	Bit4	Bit3	Bit2	Bit1	Bit0	
TGRD_0	Bit15	Bit14	Bit13	Bit12	Bit11	Bit10	Bit9	Bit8	
	Bit7	Bit6	Bit5	Bit4	Bit3	Bit2	Bit1	Bit0	
TCR_1	—	CCLR1	CCLR0	CKEG1	CKEG0	TPSC2	TPSC1	TPSC0	TP
TMDR_1	—	—	—	—	MD3	MD2	MD1	MD0	
TIOR_1	IOB3	IOB2	IOB1	IOB0	IOA3	IOA2	IOA1	IOA0	

	Bit7	Bit6	Bit5	Bit4	Bit3	Bit2	Bit1	Bit0	
TCR_2	—	CCLR1	CCLR0	CKEG1	CKEG0	TPSC2	TPSC1	TPSC0	T
TMDR_2	—	—	—	—	MD3	MD2	MD1	MD0	
TIOR_2	IOB3	IOB2	IOB1	IOB0	IOA3	IOA2	IOA1	IOA0	
TIER_2	TTGE	—	TCIEU	TCIEV	—	—	TGIEB	TGIEA	
TSR_2	TCFD	—	TCFU	TCFV	—	—	TGFB	TGFA	
TCNT_2	Bit15	Bit14	Bit13	Bit12	Bit11	Bit10	Bit9	Bit8	
	Bit7	Bit6	Bit5	Bit4	Bit3	Bit2	Bit1	Bit0	
TGRA_2	Bit15	Bit14	Bit13	Bit12	Bit11	Bit10	Bit9	Bit8	
	Bit7	Bit6	Bit5	Bit4	Bit3	Bit2	Bit1	Bit0	
TGRB_2	Bit15	Bit14	Bit13	Bit12	Bit11	Bit10	Bit9	Bit8	
	Bit7	Bit6	Bit5	Bit4	Bit3	Bit2	Bit1	Bit0	
TCR_0	CMIEB	CMIEA	OVIE	CCLR1	CCLR0	CKS2	CKS1	CKS0	T
TCR_1	CMIEB	CMIEA	OVIE	CCLR1	CCLR0	CKS2	CKS1	CKS0	T
TCSR_0	CMFB	CMFA	OVF	ADTE	OS3	OS2	OS1	OS0	T
TCSR_1	CMFB	CMFA	OVF	—	OS3	OS2	OS1	OS0	T
TCORA_0	Bit7	Bit6	Bit5	Bit4	Bit3	Bit2	Bit1	Bit0	T
TCORA_1	Bit7	Bit6	Bit5	Bit4	Bit3	Bit2	Bit1	Bit0	T
TCORB_0	Bit7	Bit6	Bit5	Bit4	Bit3	Bit2	Bit1	Bit0	T
TCORB_1	Bit7	Bit6	Bit5	Bit4	Bit3	Bit2	Bit1	Bit0	T
TCNT_0	Bit7	Bit6	Bit5	Bit4	Bit3	Bit2	Bit1	Bit0	T
TCNT_1	Bit7	Bit6	Bit5	Bit4	Bit3	Bit2	Bit1	Bit0	T

SSR_0*[1]	TDRE (TDRE)	RDRF (RDRF)	ORER (ORER)	FER (ERS)	PER (PER)	TEND (TEND)	MPB (MPB)	MPBT (MPBT)	
RDR_0	Bit7	Bit6	Bit5	Bit4	Bit3	Bit2	Bit1	Bit0	
SCMR_0	—	—	—	—	SDIR	SINV	—	SMIF	
SMR_1*[1]	C/$\overline{A}$ (GM)	CHR (BLK)	PE (PE)	O/$\overline{E}$ (O/$\overline{E}$)	STOP (BCP1)	MP (BCP0)	CKS1 (CKS1)	CKS0 (CKS0)	SC
BRR_1	Bit7	Bit6	Bit5	Bit4	Bit3	Bit2	Bit1	Bit0	
SCR_1	TIE	RIE	TE	RE	MPIE	TEIE	CKE1	CKE0	
TDR_1	Bit7	Bit6	Bit5	Bit4	Bit3	Bit2	Bit1	Bit0	
SSR_1*[1]	TDRE (TDRE)	RDRF (RDRF)	ORER (ORER)	FER (ERS)	PER (PER)	TEND (TEND)	MPB (MPB)	MPBT (MPBT)	
RDR_1	Bit7	Bit6	Bit5	Bit4	Bit3	Bit2	Bit1	Bit0	
SCMR_1	—	—	—	—	SDIR	SINV	—	SMIF	
SMR_2*[1]	C/$\overline{A}$ (GM)	CHR (BLK)	PE (PE)	O/$\overline{E}$ (O/$\overline{E}$)	STOP (BCP1)	MP (BCP0)	CKS1 (CKS1)	CKS0 (CKS0)	SC
BRR_2	Bit7	Bit6	Bit5	Bit4	Bit3	Bit2	Bit1	Bit0	
SCR_2	TIE	RIE	TE	RE	MPIE	TEIE	CKE1	CKE0	
TDR_2	Bit7	Bit6	Bit5	Bit4	Bit3	Bit2	Bit1	Bit0	
SSR_2*[1]	TDRE (TDRE)	RDRF (RDRF)	ORER (ORER)	FER (ERS)	PER (PER)	TEND (TEND)	MPB (MPB)	MPBT (MPBT)	
RDR_2	Bit7	Bit6	Bit5	Bit4	Bit3	Bit2	Bit1	Bit0	
SCMR_2	—	—	—	—	SDIR	SINV	—	SMIF	

ADDRDE	AD1	AD0	—	—	—	—	—	—	
ADCSR	ADF	ADIE	ADST	SCAN	CH3	CH2	CH1	CH0	
ADCR	TRGS1	TRGS0	—	—	CKS1	CKS0	—	—	
TCSR_1	OVF	WT/$\overline{IT}$	TME	PSS	RST/$\overline{NMI}$	CKS2	CKS1	CKS0	W
TCNT_1	Bit7	Bit6	Bit5	Bit4	Bit3	Bit2	Bit1	Bit0	
FCCS	—	—	—	FLER	—	—	—	SCO	F
FPCS	—	—	—	—	—	—	—	PPVS	
FECS	—	—	—	—	—	—	—	EPVB	
FKEY	K7	K6	K5	K4	K3	K2	K1	K0	
FMATS	MS7	MS6	MS5	MS4	MS3	MS2	MS1	MS0	
FTDAR	TDER	TDA6	TDA5	TDA4	TDA3	TDA2	TDA1	TDA0	
FVACR	FVCHGE	—	—	—	—	—	—	—	
FVADRR	Bit31	Bit30	Bit29	Bit28	Bit27	Bit26	Bit25	Bit24	
FVADRE	Bit23	Bit22	Bit21	Bit20	Bit19	Bit18	Bit17	Bit16	
FVADRH	Bit15	Bit14	Bit13	Bit12	Bit11	Bit10	Bit9	Bit8	
FVADRL	Bit7	Bit6	Bit5	Bit4	Bit3	Bit2	Bit1	Bit0	
PORT1	P17	P16	P15	P14	P13	P12	P11	P10	P
PORT2	P27	P26	P25	P24	P23	P22	P21	P20	
PORT3	P37	P36	P35	P34	P33	P32	P31	P30	
PORT4	P47	P46	P45	P44	P43	P42	P41	P40	
PORT5	—	—	—	—	—	P52	P51	P50	
PORT7	P77	P76	P75	P74	P73	P72	P71	P70	
PORT9	P97	P96	P95	P94	P93	P92	P91	P90	
PORTA	PA7	PA6	PA5	PA4	PA3	PA2	PA1	PA0	

2. Reserved in the H8S/2556 Group.

IECTR	Initialized	—	—	—	Initialized	Initialized	Initialized	Initialized
IECMR	Initialized	—	—	—	Initialized	Initialized	Initialized	Initialized
IEMCR	Initialized	—	—	—	Initialized	Initialized	Initialized	Initialized
IEAR1	Initialized	—	—	—	Initialized	Initialized	Initialized	Initialized
IEAR2	Initialized	—	—	—	Initialized	Initialized	Initialized	Initialized
IESA1	Initialized	—	—	—	Initialized	Initialized	Initialized	Initialized
IESA2	Initialized	—	—	—	Initialized	Initialized	Initialized	Initialized
IETBFL	Initialized	—	—	—	Initialized	Initialized	Initialized	Initialized
IETBR	Initialized	—	—	—	Initialized	Initialized	Initialized	Initialized
IEMA1	Initialized	—	—	—	Initialized	Initialized	Initialized	Initialized
IEMA2	Initialized	—	—	—	Initialized	Initialized	Initialized	Initialized
IERCTL	Initialized	—	—	—	Initialized	Initialized	Initialized	Initialized
IERBFL	Initialized	—	—	—	Initialized	Initialized	Initialized	Initialized
IERBR	Initialized	—	—	—	Initialized	Initialized	Initialized	Initialized
IELA1	Initialized	—	—	—	Initialized	Initialized	Initialized	Initialized
IELA2	Initialized	—	—	—	Initialized	Initialized	Initialized	Initialized
IEFLG	Initialized	—	—	—	Initialized	Initialized	Initialized	Initialized
IETSR	Initialized	—	—	—	Initialized	Initialized	Initialized	Initialized
IEIET	Initialized	—	—	—	Initialized	Initialized	Initialized	Initialized
IETEF	Initialized	—	—	—	Initialized	Initialized	Initialized	Initialized
IERSR	Initialized	—	—	—	Initialized	Initialized	Initialized	Initialized
IEIER	Initialized	—	—	—	Initialized	Initialized	Initialized	Initialized
IEREF	Initialized	—	—	—	Initialized	Initialized	Initialized	Initialized

ICMR_0	Initialized	—	—	—	—	—	—	Initialized	
ICIER_0	Initialized	—	—	—	—	—	—	Initialized	
ICSR_0	Initialized	—	—	—	—	—	—	Initialized	
SAR_0	Initialized	—	—	—	—	—	—	Initialized	
ICDRT_0	Initialized	—	—	—	—	—	—	Initialized	
ICDRR_0	Initialized	—	—	—	—	—	—	Initialized	
ICCR1_1	Initialized	—	—	—	—	—	—	Initialized	I
ICCR2_1	Initialized	—	—	—	—	—	—	Initialized	
ICMR_1	Initialized	—	—	—	—	—	—	Initialized	
ICIER_1	Initialized	—	—	—	—	—	—	Initialized	
ICSR_1	Initialized	—	—	—	—	—	—	Initialized	
SAR_1	Initialized	—	—	—	—	—	—	Initialized	
ICDRT_1	Initialized	—	—	—	—	—	—	Initialized	
ICDRR_1	Initialized	—	—	—	—	—	—	Initialized	
MCR	Initialized	—	—	—	Initialized	Initialized	Initialized	Initialized	H
GSR	Initialized	—	—	—	Initialized	Initialized	Initialized	Initialized	
BCR	Initialized	—	—	—	Initialized	Initialized	Initialized	Initialized	
MBCR	Initialized	—	—	—	Initialized	Initialized	Initialized	Initialized	
TXPR	Initialized	—	—	—	Initialized	Initialized	Initialized	Initialized	
TXCR	Initialized	—	—	—	Initialized	Initialized	Initialized	Initialized	
TXACK	Initialized	—	—	—	Initialized	Initialized	Initialized	Initialized	
ABACK	Initialized	—	—	—	Initialized	Initialized	Initialized	Initialized	
RXPR	Initialized	—	—	—	Initialized	Initialized	Initialized	Initialized	
RFPR	Initialized	—	—	—	Initialized	Initialized	Initialized	Initialized	

MC0[1]	Initialized	—	—	—	Initialized	Initialized	Initialized	Initialized
MC0[2]	Initialized	—	—	—	Initialized	Initialized	Initialized	Initialized
MC0[3]	Initialized	—	—	—	Initialized	Initialized	Initialized	Initialized
MC0[4]	Initialized	—	—	—	Initialized	Initialized	Initialized	Initialized
MC0[5]	Initialized	—	—	—	Initialized	Initialized	Initialized	Initialized
MC0[6]	Initialized	—	—	—	Initialized	Initialized	Initialized	Initialized
MC0[7]	Initialized	—	—	—	Initialized	Initialized	Initialized	Initialized
MC0[8]	Initialized	—	—	—	Initialized	Initialized	Initialized	Initialized
MC1[1]	Initialized	—	—	—	Initialized	Initialized	Initialized	Initialized
MC1[2]	Initialized	—	—	—	Initialized	Initialized	Initialized	Initialized
MC1[3]	Initialized	—	—	—	Initialized	Initialized	Initialized	Initialized
MC1[4]	Initialized	—	—	—	Initialized	Initialized	Initialized	Initialized
MC1[5]	Initialized	—	—	—	Initialized	Initialized	Initialized	Initialized
MC1[6]	Initialized	—	—	—	Initialized	Initialized	Initialized	Initialized
MC1[7]	Initialized	—	—	—	Initialized	Initialized	Initialized	Initialized
MC1[8]	Initialized	—	—	—	Initialized	Initialized	Initialized	Initialized
MC2[1]	Initialized	—	—	—	Initialized	Initialized	Initialized	Initialized
MC2[2]	Initialized	—	—	—	Initialized	Initialized	Initialized	Initialized
MC2[3]	Initialized	—	—	—	Initialized	Initialized	Initialized	Initialized
MC2[4]	Initialized	—	—	—	Initialized	Initialized	Initialized	Initialized
MC2[5]	Initialized	—	—	—	Initialized	Initialized	Initialized	Initialized
MC2[6]	Initialized	—	—	—	Initialized	Initialized	Initialized	Initialized
MC2[7]	Initialized	—	—	—	Initialized	Initialized	Initialized	Initialized
MC2[8]	Initialized	—	—	—	Initialized	Initialized	Initialized	Initialized

MC4[1]	Initialized	—	—	—	Initialized	Initialized	Initialized	Initialized
MC4[2]	Initialized	—	—	—	Initialized	Initialized	Initialized	Initialized
MC4[3]	Initialized	—	—	—	Initialized	Initialized	Initialized	Initialized
MC4[4]	Initialized	—	—	—	Initialized	Initialized	Initialized	Initialized
MC4[5]	Initialized	—	—	—	Initialized	Initialized	Initialized	Initialized
MC4[6]	Initialized	—	—	—	Initialized	Initialized	Initialized	Initialized
MC4[7]	Initialized	—	—	—	Initialized	Initialized	Initialized	Initialized
MC4[8]	Initialized	—	—	—	Initialized	Initialized	Initialized	Initialized
MC5[1]	Initialized	—	—	—	Initialized	Initialized	Initialized	Initialized
MC5[2]	Initialized	—	—	—	Initialized	Initialized	Initialized	Initialized
MC5[3]	Initialized	—	—	—	Initialized	Initialized	Initialized	Initialized
MC5[4]	Initialized	—	—	—	Initialized	Initialized	Initialized	Initialized
MC5[5]	Initialized	—	—	—	Initialized	Initialized	Initialized	Initialized
MC5[6]	Initialized	—	—	—	Initialized	Initialized	Initialized	Initialized
MC5[7]	Initialized	—	—	—	Initialized	Initialized	Initialized	Initialized
MC5[8]	Initialized	—	—	—	Initialized	Initialized	Initialized	Initialized
MC6[1]	Initialized	—	—	—	Initialized	Initialized	Initialized	Initialized
MC6[2]	Initialized	—	—	—	Initialized	Initialized	Initialized	Initialized
MC6[3]	Initialized	—	—	—	Initialized	Initialized	Initialized	Initialized
MC6[4]	Initialized	—	—	—	Initialized	Initialized	Initialized	Initialized
MC6[5]	Initialized	—	—	—	Initialized	Initialized	Initialized	Initialized
MC6[6]	Initialized	—	—	—	Initialized	Initialized	Initialized	Initialized
MC6[7]	Initialized	—	—	—	Initialized	Initialized	Initialized	Initialized
MC6[8]	Initialized	—	—	—	Initialized	Initialized	Initialized	Initialized

MC8[1]	Initialized	—	—	—	Initialized	Initialized	Initialized	Initialized
MC8[2]	Initialized	—	—	—	Initialized	Initialized	Initialized	Initialized
MC8[3]	Initialized	—	—	—	Initialized	Initialized	Initialized	Initialized
MC8[4]	Initialized	—	—	—	Initialized	Initialized	Initialized	Initialized
MC8[5]	Initialized	—	—	—	Initialized	Initialized	Initialized	Initialized
MC8[6]	Initialized	—	—	—	Initialized	Initialized	Initialized	Initialized
MC8[7]	Initialized	—	—	—	Initialized	Initialized	Initialized	Initialized
MC8[8]	Initialized	—	—	—	Initialized	Initialized	Initialized	Initialized
MC9[1]	Initialized	—	—	—	Initialized	Initialized	Initialized	Initialized
MC9[2]	Initialized	—	—	—	Initialized	Initialized	Initialized	Initialized
MC9[3]	Initialized	—	—	—	Initialized	Initialized	Initialized	Initialized
MC9[4]	Initialized	—	—	—	Initialized	Initialized	Initialized	Initialized
MC9[5]	Initialized	—	—	—	Initialized	Initialized	Initialized	Initialized
MC9[6]	Initialized	—	—	—	Initialized	Initialized	Initialized	Initialized
MC9[7]	Initialized	—	—	—	Initialized	Initialized	Initialized	Initialized
MC9[8]	Initialized	—	—	—	Initialized	Initialized	Initialized	Initialized
MC10[1]	Initialized	—	—	—	Initialized	Initialized	Initialized	Initialized
MC10[2]	Initialized	—	—	—	Initialized	Initialized	Initialized	Initialized
MC10[3]	Initialized	—	—	—	Initialized	Initialized	Initialized	Initialized
MC10[4]	Initialized	—	—	—	Initialized	Initialized	Initialized	Initialized
MC10[5]	Initialized	—	—	—	Initialized	Initialized	Initialized	Initialized
MC10[6]	Initialized	—	—	—	Initialized	Initialized	Initialized	Initialized
MC10[7]	Initialized	—	—	—	Initialized	Initialized	Initialized	Initialized
MC10[8]	Initialized	—	—	—	Initialized	Initialized	Initialized	Initialized

MC12[1]	Initialized	—	—	—	Initialized	Initialized	Initialized	Initialized
MC12[2]	Initialized	—	—	—	Initialized	Initialized	Initialized	Initialized
MC12[3]	Initialized	—	—	—	Initialized	Initialized	Initialized	Initialized
MC12[4]	Initialized	—	—	—	Initialized	Initialized	Initialized	Initialized
MC12[5]	Initialized	—	—	—	Initialized	Initialized	Initialized	Initialized
MC12[6]	Initialized	—	—	—	Initialized	Initialized	Initialized	Initialized
MC12[7]	Initialized	—	—	—	Initialized	Initialized	Initialized	Initialized
MC12[8]	Initialized	—	—	—	Initialized	Initialized	Initialized	Initialized
MC13[1]	Initialized	—	—	—	Initialized	Initialized	Initialized	Initialized
MC13[2]	Initialized	—	—	—	Initialized	Initialized	Initialized	Initialized
MC13[3]	Initialized	—	—	—	Initialized	Initialized	Initialized	Initialized
MC13[4]	Initialized	—	—	—	Initialized	Initialized	Initialized	Initialized
MC13[5]	Initialized	—	—	—	Initialized	Initialized	Initialized	Initialized
MC13[6]	Initialized	—	—	—	Initialized	Initialized	Initialized	Initialized
MC13[7]	Initialized	—	—	—	Initialized	Initialized	Initialized	Initialized
MC13[8]	Initialized	—	—	—	Initialized	Initialized	Initialized	Initialized
MC14[1]	Initialized	—	—	—	Initialized	Initialized	Initialized	Initialized
MC14[2]	Initialized	—	—	—	Initialized	Initialized	Initialized	Initialized
MC14[3]	Initialized	—	—	—	Initialized	Initialized	Initialized	Initialized
MC14[4]	Initialized	—	—	—	Initialized	Initialized	Initialized	Initialized
MC14[5]	Initialized	—	—	—	Initialized	Initialized	Initialized	Initialized
MC14[6]	Initialized	—	—	—	Initialized	Initialized	Initialized	Initialized
MC14[7]	Initialized	—	—	—	Initialized	Initialized	Initialized	Initialized
MC14[8]	Initialized	—	—	—	Initialized	Initialized	Initialized	Initialized

MD0[1]	Initialized	—	—	—	Initialized	Initialized	Initialized	Initialized
MD0[2]	Initialized	—	—	—	Initialized	Initialized	Initialized	Initialized
MD0[3]	Initialized	—	—	—	Initialized	Initialized	Initialized	Initialized
MD0[4]	Initialized	—	—	—	Initialized	Initialized	Initialized	Initialized
MD0[5]	Initialized	—	—	—	Initialized	Initialized	Initialized	Initialized
MD0[6]	Initialized	—	—	—	Initialized	Initialized	Initialized	Initialized
MD0[7]	Initialized	—	—	—	Initialized	Initialized	Initialized	Initialized
MD0[8]	Initialized	—	—	—	Initialized	Initialized	Initialized	Initialized
MD1[1]	Initialized	—	—	—	Initialized	Initialized	Initialized	Initialized
MD1[2]	Initialized	—	—	—	Initialized	Initialized	Initialized	Initialized
MD1[3]	Initialized	—	—	—	Initialized	Initialized	Initialized	Initialized
MD1[4]	Initialized	—	—	—	Initialized	Initialized	Initialized	Initialized
MD1[5]	Initialized	—	—	—	Initialized	Initialized	Initialized	Initialized
MD1[6]	Initialized	—	—	—	Initialized	Initialized	Initialized	Initialized
MD1[7]	Initialized	—	—	—	Initialized	Initialized	Initialized	Initialized
MD1[8]	Initialized	—	—	—	Initialized	Initialized	Initialized	Initialized
MD2[1]	Initialized	—	—	—	Initialized	Initialized	Initialized	Initialized
MD2[2]	Initialized	—	—	—	Initialized	Initialized	Initialized	Initialized
MD2[3]	Initialized	—	—	—	Initialized	Initialized	Initialized	Initialized
MD2[4]	Initialized	—	—	—	Initialized	Initialized	Initialized	Initialized
MD2[5]	Initialized	—	—	—	Initialized	Initialized	Initialized	Initialized
MD2[6]	Initialized	—	—	—	Initialized	Initialized	Initialized	Initialized
MD2[7]	Initialized	—	—	—	Initialized	Initialized	Initialized	Initialized
MD2[8]	Initialized	—	—	—	Initialized	Initialized	Initialized	Initialized

MD4[1]	Initialized	—	—	—	Initialized	Initialized	Initialized	Initialized
MD4[2]	Initialized	—	—	—	Initialized	Initialized	Initialized	Initialized
MD4[3]	Initialized	—	—	—	Initialized	Initialized	Initialized	Initialized
MD4[4]	Initialized	—	—	—	Initialized	Initialized	Initialized	Initialized
MD4[5]	Initialized	—	—	—	Initialized	Initialized	Initialized	Initialized
MD4[6]	Initialized	—	—	—	Initialized	Initialized	Initialized	Initialized
MD4[7]	Initialized	—	—	—	Initialized	Initialized	Initialized	Initialized
MD4[8]	Initialized	—	—	—	Initialized	Initialized	Initialized	Initialized
MD5[1]	Initialized	—	—	—	Initialized	Initialized	Initialized	Initialized
MD5[2]	Initialized	—	—	—	Initialized	Initialized	Initialized	Initialized
MD5[3]	Initialized	—	—	—	Initialized	Initialized	Initialized	Initialized
MD5[4]	Initialized	—	—	—	Initialized	Initialized	Initialized	Initialized
MD5[5]	Initialized	—	—	—	Initialized	Initialized	Initialized	Initialized
MD5[6]	Initialized	—	—	—	Initialized	Initialized	Initialized	Initialized
MD5[7]	Initialized	—	—	—	Initialized	Initialized	Initialized	Initialized
MD5[8]	Initialized	—	—	—	Initialized	Initialized	Initialized	Initialized
MD6[1]	Initialized	—	—	—	Initialized	Initialized	Initialized	Initialized
MD6[2]	Initialized	—	—	—	Initialized	Initialized	Initialized	Initialized
MD6[3]	Initialized	—	—	—	Initialized	Initialized	Initialized	Initialized
MD6[4]	Initialized	—	—	—	Initialized	Initialized	Initialized	Initialized
MD6[5]	Initialized	—	—	—	Initialized	Initialized	Initialized	Initialized
MD6[6]	Initialized	—	—	—	Initialized	Initialized	Initialized	Initialized
MD6[7]	Initialized	—	—	—	Initialized	Initialized	Initialized	Initialized
MD6[8]	Initialized	—	—	—	Initialized	Initialized	Initialized	Initialized

MD8[1]	Initialized	—	—	—	Initialized	Initialized	Initialized	Initialized
MD8[2]	Initialized	—	—	—	Initialized	Initialized	Initialized	Initialized
MD8[3]	Initialized	—	—	—	Initialized	Initialized	Initialized	Initialized
MD8[4]	Initialized	—	—	—	Initialized	Initialized	Initialized	Initialized
MD8[5]	Initialized	—	—	—	Initialized	Initialized	Initialized	Initialized
MD8[6]	Initialized	—	—	—	Initialized	Initialized	Initialized	Initialized
MD8[7]	Initialized	—	—	—	Initialized	Initialized	Initialized	Initialized
MD8[8]	Initialized	—	—	—	Initialized	Initialized	Initialized	Initialized
MD9[1]	Initialized	—	—	—	Initialized	Initialized	Initialized	Initialized
MD9[2]	Initialized	—	—	—	Initialized	Initialized	Initialized	Initialized
MD9[3]	Initialized	—	—	—	Initialized	Initialized	Initialized	Initialized
MD9[4]	Initialized	—	—	—	Initialized	Initialized	Initialized	Initialized
MD9[5]	Initialized	—	—	—	Initialized	Initialized	Initialized	Initialized
MD9[6]	Initialized	—	—	—	Initialized	Initialized	Initialized	Initialized
MD9[7]	Initialized	—	—	—	Initialized	Initialized	Initialized	Initialized
MD9[8]	Initialized	—	—	—	Initialized	Initialized	Initialized	Initialized
MD10[1]	Initialized	—	—	—	Initialized	Initialized	Initialized	Initialized
MD10[2]	Initialized	—	—	—	Initialized	Initialized	Initialized	Initialized
MD10[3]	Initialized	—	—	—	Initialized	Initialized	Initialized	Initialized
MD10[4]	Initialized	—	—	—	Initialized	Initialized	Initialized	Initialized
MD10[5]	Initialized	—	—	—	Initialized	Initialized	Initialized	Initialized
MD10[6]	Initialized	—	—	—	Initialized	Initialized	Initialized	Initialized
MD10[7]	Initialized	—	—	—	Initialized	Initialized	Initialized	Initialized
MD10[8]	Initialized	—	—	—	Initialized	Initialized	Initialized	Initialized

MD12[1]	Initialized	—	—	—	Initialized	Initialized	Initialized	Initialized
MD12[2]	Initialized	—	—	—	Initialized	Initialized	Initialized	Initialized
MD12[3]	Initialized	—	—	—	Initialized	Initialized	Initialized	Initialized
MD12[4]	Initialized	—	—	—	Initialized	Initialized	Initialized	Initialized
MD12[5]	Initialized	—	—	—	Initialized	Initialized	Initialized	Initialized
MD12[6]	Initialized	—	—	—	Initialized	Initialized	Initialized	Initialized
MD12[7]	Initialized	—	—	—	Initialized	Initialized	Initialized	Initialized
MD12[8]	Initialized	—	—	—	Initialized	Initialized	Initialized	Initialized
MD13[1]	Initialized	—	—	—	Initialized	Initialized	Initialized	Initialized
MD13[2]	Initialized	—	—	—	Initialized	Initialized	Initialized	Initialized
MD13[3]	Initialized	—	—	—	Initialized	Initialized	Initialized	Initialized
MD13[4]	Initialized	—	—	—	Initialized	Initialized	Initialized	Initialized
MD13[5]	Initialized	—	—	—	Initialized	Initialized	Initialized	Initialized
MD13[6]	Initialized	—	—	—	Initialized	Initialized	Initialized	Initialized
MD13[7]	Initialized	—	—	—	Initialized	Initialized	Initialized	Initialized
MD13[8]	Initialized	—	—	—	Initialized	Initialized	Initialized	Initialized
MD14[1]	Initialized	—	—	—	Initialized	Initialized	Initialized	Initialized
MD14[2]	Initialized	—	—	—	Initialized	Initialized	Initialized	Initialized
MD14[3]	Initialized	—	—	—	Initialized	Initialized	Initialized	Initialized
MD14[4]	Initialized	—	—	—	Initialized	Initialized	Initialized	Initialized
MD14[5]	Initialized	—	—	—	Initialized	Initialized	Initialized	Initialized
MD14[6]	Initialized	—	—	—	Initialized	Initialized	Initialized	Initialized
MD14[7]	Initialized	—	—	—	Initialized	Initialized	Initialized	Initialized
MD14[8]	Initialized	—	—	—	Initialized	Initialized	Initialized	Initialized

DADR0	Initialized	—	—	—	—	—	—	Initialized
DADR1	Initialized	—	—	—	—	—	—	Initialized
DACR	Initialized	—	—	—	—	—	—	Initialized
TCR_2	Initialized	—	—	—	—	—	—	Initialized
TCR_3	Initialized	—	—	—	—	—	—	Initialized
TCSR_2	Initialized	—	—	—	—	—	—	Initialized
TCSR_3	Initialized	—	—	—	—	—	—	Initialized
TCORA_2	Initialized	—	—	—	—	—	—	Initialized
TCORA_3	Initialized	—	—	—	—	—	—	Initialized
TCORB_2	Initialized	—	—	—	—	—	—	Initialized
TCORB_3	Initialized	—	—	—	—	—	—	Initialized
TCNT_2	Initialized	—	—	—	—	—	—	Initialized
TCNT_3	Initialized	—	—	—	—	—	—	Initialized
SMR_3	Initialized	—	—	—	—	—	—	Initialized
BRR_3	Initialized	—	—	—	—	—	—	Initialized
SCR_3	Initialized	—	—	—	—	—	—	Initialized
TDR_3	Initialized	—	—	—	Initialized	Initialized	Initialized	Initialized
SSR_3	Initialized	—	—	—	Initialized	Initialized	Initialized	Initialized
RDR_3	Initialized	—	—	—	Initialized	Initialized	Initialized	Initialized
SCMR_3	Initialized	—	—	—	—	—	—	Initialized
SMR_4	Initialized	—	—	—	—	—	—	Initialized
BRR_4	Initialized	—	—	—	—	—	—	Initialized
SCR_4	Initialized	—	—	—	—	—	—	Initialized
TDR_4	Initialized	—	—	—	Initialized	Initialized	Initialized	Initialized

MDCR	Initialized*	—	—	—	—	—	—	Initialized	
MSTPCRA	Initialized	—	—	—	—	—	—	Initialized	
MSTPCRB	Initialized	—	—	—	—	—	—	Initialized	
MSTPCRC	Initialized	—	—	—	—	—	—	Initialized	
PFCR	Initialized*	—	—	—	—	—	—	Initialized	B
LPWRCR	Initialized*	—	—	—	—	—	—	Initialized	S
BARA	Initialized*	—	—	—	—	—	—	Initialized	P
BARB	Initialized*	—	—	—	—	—	—	Initialized	
BCRA	Initialized*	—	—	—	—	—	—	Initialized	
BCRB	Initialized*	—	—	—	—	—	—	Initialized	
ISCRH	Initialized	—	—	—	—	—	—	Initialized	I
ISCRL	Initialized	—	—	—	—	—	—	Initialized	
IER	Initialized	—	—	—	—	—	—	Initialized	
ISR	Initialized	—	—	—	—	—	—	Initialized	
DTCERA	Initialized	—	—	—	—	—	—	Initialized	D
DTCERB	Initialized	—	—	—	—	—	—	Initialized	
DTCERC	Initialized	—	—	—	—	—	—	Initialized	
DTCERD	Initialized	—	—	—	—	—	—	Initialized	
DTCERE	Initialized	—	—	—	—	—	—	Initialized	
DTCERF	Initialized	—	—	—	—	—	—	Initialized	
DTCERG	Initialized	—	—	—	—	—	—	Initialized	
DTCERI	Initialized	—	—	—	—	—	—	Initialized	
DTVECR	Initialized	—	—	—	—	—	—	Initialized	

PDDDR	Initialized*	—	—	—	—	—	—	Initialized
PEDDR	Initialized*	—	—	—	—	—	—	Initialized
PFDDR	Initialized*	—	—	—	—	—	—	Initialized
PGDDR	Initialized*	—	—	—	—	—	—	Initialized
PAPCR	Initialized*	—	—	—	—	—	—	Initialized
PBPCR	Initialized*	—	—	—	—	—	—	Initialized
PCPCR	Initialized*	—	—	—	—	—	—	Initialized
PDPCR	Initialized*	—	—	—	—	—	—	Initialized
PEPCR	Initialized*	—	—	—	—	—	—	Initialized
P3ODR	Initialized*	—	—	—	—	—	—	Initialized
PAODR	Initialized*	—	—	—	—	—	—	Initialized
TCR_3	Initialized	—	—	—	—	—	—	Initialized
TMDR_3	Initialized	—	—	—	—	—	—	Initialized
TIORH_3	Initialized	—	—	—	—	—	—	Initialized
TIORL_3	Initialized	—	—	—	—	—	—	Initialized
TIER_3	Initialized	—	—	—	—	—	—	Initialized
TSR_3	Initialized	—	—	—	—	—	—	Initialized
TCNT_3	Initialized	—	—	—	—	—	—	Initialized
TGRA_3	Initialized	—	—	—	—	—	—	Initialized
TGRB_3	Initialized	—	—	—	—	—	—	Initialized
TGRC_3	Initialized	—	—	—	—	—	—	Initialized
TGRD_3	Initialized	—	—	—	—	—	—	Initialized
TCR_4	Initialized	—	—	—	—	—	—	Initialized
TMDR_4	Initialized	—	—	—	—	—	—	Initialized

TIOR_5	Initialized	—	—	—	—	—	—	Initialized	
TIER_5	Initialized	—	—	—	—	—	—	Initialized	
TSR_5	Initialized	—	—	—	—	—	—	Initialized	
TCNT_5	Initialized	—	—	—	—	—	—	Initialized	
TGRA_5	Initialized	—	—	—	—	—	—	Initialized	
TGRB_5	Initialized	—	—	—	—	—	—	Initialized	
TSTR	Initialized	—	—	—	—	—	—	Initialized	T
TSYR	Initialized	—	—	—	—	—	—	Initialized	
IPRA	Initialized	—	—	—	—	—	—	Initialized	I
IPRB	Initialized	—	—	—	—	—	—	Initialized	
IPRC	Initialized	—	—	—	—	—	—	Initialized	
IPRD	Initialized	—	—	—	—	—	—	Initialized	
IPRE	Initialized	—	—	—	—	—	—	Initialized	
IPRF	Initialized	—	—	—	—	—	—	Initialized	
IPRG	Initialized	—	—	—	—	—	—	Initialized	
IPRH	Initialized	—	—	—	—	—	—	Initialized	
IPRI	Initialized	—	—	—	—	—	—	Initialized	
IPRJ	Initialized	—	—	—	—	—	—	Initialized	
IPRK	Initialized	—	—	—	—	—	—	Initialized	
IPRL	Initialized	—	—	—	—	—	—	Initialized	
IPRM	Initialized	—	—	—	—	—	—	Initialized	
IPRO	Initialized	—	—	—	—	—	—	Initialized	
ABWCR	Initialized*	—	—	—	—	—	—	Initialized	B
ASTCR	Initialized*	—	—	—	—	—	—	Initialized	

P5DR	Initialized*	—	—	—	—	—	—	Initialized
P7DR	Initialized*	—	—	—	—	—	—	Initialized
PADR	Initialized*	—	—	—	—	—	—	Initialized
PBDR	Initialized*	—	—	—	—	—	—	Initialized
PCDR	Initialized*	—	—	—	—	—	—	Initialized
PDDR	Initialized*	—	—	—	—	—	—	Initialized
PEDR	Initialized*	—	—	—	—	—	—	Initialized
PFDR	Initialized*	—	—	—	—	—	—	Initialized
PGDR	Initialized*	—	—	—	—	—	—	Initialized
TCR_0	Initialized	—	—	—	—	—	—	Initialized
TMDR_0	Initialized	—	—	—	—	—	—	Initialized
TIORH_0	Initialized	—	—	—	—	—	—	Initialized
TIORL_0	Initialized	—	—	—	—	—	—	Initialized
TIER_0	Initialized	—	—	—	—	—	—	Initialized
TSR_0	Initialized	—	—	—	—	—	—	Initialized
TCNT_0	Initialized	—	—	—	—	—	—	Initialized
TGRA_0	Initialized	—	—	—	—	—	—	Initialized
TGRB_0	Initialized	—	—	—	—	—	—	Initialized
TGRC_0	Initialized	—	—	—	—	—	—	Initialized
TGRD_0	Initialized	—	—	—	—	—	—	Initialized
TCR_1	Initialized	—	—	—	—	—	—	Initialized
TMDR_1	Initialized	—	—	—	—	—	—	Initialized
TIOR_1	Initialized	—	—	—	—	—	—	Initialized
TIER_1	Initialized	—	—	—	—	—	—	Initialized

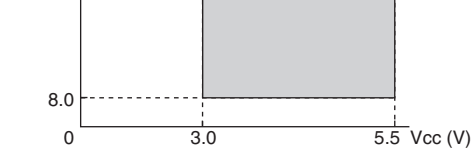
TSR_2	Initialized	—	—	—	—	—	—	Initialized	
TCNT_2	Initialized	—	—	—	—	—	—	Initialized	
TGRA_2	Initialized	—	—	—	—	—	—	Initialized	
TGRB_2	Initialized	—	—	—	—	—	—	Initialized	T
TCR_0	Initialized	—	—	—	—	—	—	Initialized	T
TCR_1	Initialized	—	—	—	—	—	—	Initialized	T
TCSR_0	Initialized	—	—	—	—	—	—	Initialized	T
TCSR_1	Initialized	—	—	—	—	—	—	Initialized	T
TCORA_0	Initialized	—	—	—	—	—	—	Initialized	T
TCORA_1	Initialized	—	—	—	—	—	—	Initialized	T
TCORB_0	Initialized	—	—	—	—	—	—	Initialized	T
TCORB_1	Initialized	—	—	—	—	—	—	Initialized	T
TCNT_0	Initialized	—	—	—	—	—	—	Initialized	T
TCNT_1	Initialized	—	—	—	—	—	—	Initialized	T
TCSR_0	Initialized	—	—	—	—	—	—	Initialized	W
TCNT_0	Initialized	—	—	—	—	—	—	Initialized	
RSTCSR	Initialized	—	—	—	—	—	—	Initialized	
SMR_0	Initialized	—	—	—	—	—	—	Initialized	S
BRR_0	Initialized	—	—	—	—	—	—	Initialized	
SCR_0	Initialized	—	—	—	—	—	—	Initialized	
TDR_0	Initialized	—	—	—	Initialized	Initialized	Initialized	Initialized	
SSR_0	Initialized	—	—	—	Initialized	Initialized	Initialized	Initialized	
RDR_0	Initialized	—	—	—	Initialized	Initialized	Initialized	Initialized	
SCMR_0	Initialized	—	—	—	—	—	—	Initialized	

BRR_2	Initialized	—	—	—	—	—	—	Initialized
SCR_2	Initialized	—	—	—	—	—	—	Initialized
TDR_2	Initialized	—	—	—	Initialized	Initialized	Initialized	Initialized
SSR_2	Initialized	—	—	—	Initialized	Initialized	Initialized	Initialized
RDR_2	Initialized	—	—	—	Initialized	Initialized	Initialized	Initialized
SCMR_2	Initialized	—	—	—	—	—	—	Initialized
ADDRAH	Initialized	—	—	—	Initialized	Initialized	Initialized	Initialized
ADDRAL	Initialized	—	—	—	Initialized	Initialized	Initialized	Initialized
ADDRBH	Initialized	—	—	—	Initialized	Initialized	Initialized	Initialized
ADDRBL	Initialized	—	—	—	Initialized	Initialized	Initialized	Initialized
ADDRCH	Initialized	—	—	—	Initialized	Initialized	Initialized	Initialized
ADDRCL	Initialized	—	—	—	Initialized	Initialized	Initialized	Initialized
ADDRDH	Initialized	—	—	—	Initialized	Initialized	Initialized	Initialized
ADDRDL	Initialized	—	—	—	Initialized	Initialized	Initialized	Initialized
ADCSR	Initialized	—	—	—	Initialized	Initialized	Initialized	Initialized
ADCR	Initialized	—	—	—	Initialized	Initialized	Initialized	Initialized
TCSR_1	Initialized	—	—	—	—	—	—	Initialized
TCNT_1	Initialized	—	—	—	—	—	—	Initialized
FCCS	Initialized	—	—	—	—	—	Initialized	Initialized
FPCS	Initialized	—	—	—	—	—	Initialized	Initialized
FECS	Initialized	—	—	—	—	—	Initialized	Initialized
FKEY	Initialized	—	—	—	—	—	Initialized	Initialized
FMATS	Initialized	—	—	—	—	—	Initialized	Initialized
FTDAR	Initialized	—	—	—	—	—	Initialized	Initialized

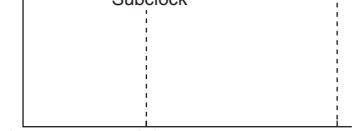
PORT4	—	—	—	—	—	—	—	—
PORT5	—	—	—	—	—	—	—	—
PORT7	—	—	—	—	—	—	—	—
PORT9	—	—	—	—	—	—	—	—
PORTA	—	—	—	—	—	—	—	—
PORTB	—	—	—	—	—	—	—	—
PORTC	—	—	—	—	—	—	—	—
PORTD	—	—	—	—	—	—	—	—
PORTE	—	—	—	—	—	—	—	—
PORTF	—	—	—	—	—	—	—	—
PORTG	—	—	—	—	—	—	—	—

Notes: — is not initialized.

* Not initialized by a manual reset.

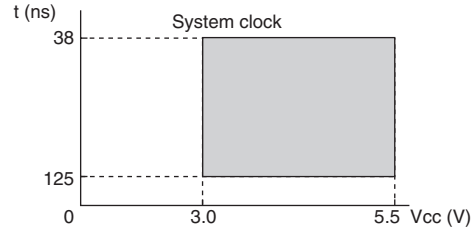


• Active (high-speed/medium-speed) mode
• Sleep mode



• All operating modes

(2) Power supply voltage and range of instruction execution



• Active (high-speed/medium-speed) mode

Note: * When the IEBus is used in the H8S/2552 Group, the system clock should be selected among 12
12.58 MHz, 18 MHz, 18.87 MHz, 24 MHz, and 25.16 MHz.
When not using the IEBus or in the H8S/2506 Group, the system clock can be selected arbitrarily
8 MHz to 26 MHz.

Figure 24.1 (1) Power Supply Voltage and Operating Ranges (H8S/2552 Group, H8S/2506 Group)

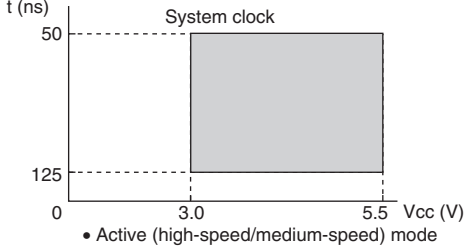


Figure 24.1 (2) Power Supply Voltage and Operating Ranges (H8S/2556 Gro

Input voltage (ports 1, 2, 3, 7)	V_{in}	–0.3 to $P2V_{CC}$ +0.3
Input voltage (ports 5, A to H, J)	V_{in}	–0.3 to $P1V_{CC}$ +0.3
Input voltage (port HRxD)*2	V_{in}	–0.3 to $P1V_{CC}$ +0.3
Input voltage (others)*3	V_{in}	–0.3 to V_{CC} +0.3
Reference voltage	V_{ref}	–0.3 to AV_{CC} +0.3
Analog power supply voltage	AV_{CC}	–0.3 to +6.5
Analog input voltage	V_{AN}	–0.3 to AV_{CC} +0.3
Operating temperature	T_{opr}	Regular specifications: –20 to +75
		Wide-range specifications: –40 to +8
Storage temperature	T_{stg}	–55 to +125

Caution: Permanent damage to the chip may result if absolute maximum rating are ex

Notes: 1. Do not apply a power supply voltage to the VCL pin. The capacitor should be to GND externally.

2. HRxD is supported in the H8S/2556 Group only. Do not apply a power supply HTxD because it is an output pin.

3. Do not connect to the OSC1 pin or OSC2 pin without a 32.768-kHz crystal re no subclock is required, connect the OSC1 pin to V_{SS} and leave the OSC2 pin

Item		Symbol	Min.	Typ.	Max.	Unit	Test Con
Schmitt trigger input voltage	$\overline{IRQ0}$, $\overline{IRQ1}$, $\overline{IRQ4}$, $\overline{IRQ5}$	VT^-	$P2V_{CC} \times 0.2$	—	—	V	
		VT^+	—	—	$P2V_{CC} \times 0.8$	V	
		$VT^+ - VT^-$	$P2V_{CC} \times 0.05$	—	—	V	$P2V_{CC}$ = 5 V
	$\overline{IRQ2}$, $\overline{IRQ3}$, $\overline{IRQ6}$, $\overline{IRQ7}$	VT^-	$P1V_{CC} \times 0.2$	—	—	V	
		VT^+	—	—	$P1V_{CC} \times 0.8$	V	
		$VT^+ - VT^-$	$P1V_{CC} \times 0.05$	—	—	V	$P1V_{CC}$ = 5 V
Input high voltage	$\overline{RES}$, $\overline{STBY}$, NMI, MD2 to MD0	V_{IH}	$V_{CC} \times 0.9$	—	$V_{CC} + 0.3$	V	
	EXTAL		$V_{CC} \times 0.8$	—	$V_{CC} + 0.3$	V	
	Ports 1 to 3, 7		$P2V_{CC} \times 0.8$	—	$P2V_{CC} + 0.3$	V	
	Ports 5, A to H, J, HRxD*5		$P1V_{CC} \times 0.8$	—	$P1V_{CC} + 0.3$	V	
	Ports 4, 9		$AV_{CC} \times 0.8$	—	$AV_{CC} + 0.3$	V	
Input low voltage	$\overline{RES}$, $\overline{STBY}$, NMI MD2 to MD0	V_{IL}	- 0.3	—	$V_{CC} \times 0.1$	V	
	EXTAL, TEST		- 0.3	—	$V_{CC} \times 0.2$	V	
	Ports 1 to 3, 7		- 0.3	—	$P2V_{CC} \times 0.2$	V	
	Ports 5, A to H, J, HRxD*5		- 0.3	—	$P1V_{CC} \times 0.2$	V	
	Ports 4, 9		- 0.3	—	$AV_{CC} \times 0.2$	V	

Output low voltage*6	All output pins*4	V_{OL}	—	—	0.4	V	I_{OL} = 0.8
Input leakage current	$\overline{RES}$	$\lvert I_{in} \rvert$	—	—	1.0	µA	V_{in} = 0.2 V
	$\overline{STBY}$, NMI, MD2 to MD0, TEST		—	—	1.0	µA	V_{in} = 0.2 V
	HRxD*5		—	—	1.0	µA	V_{in} = 0.2 - 0.2 V
	Ports 4, 9		—	—	1.0	µA	V_{in} = 0.2 - 0.2 V
Three-state leakage current (off state)	Ports 1 to 3, 7	$\lvert I_{TSI} \rvert$	—	—	1.0	µA	V_{in} = 0.2 - 0.2 V
	Ports 5, A to H, J		—	—	1.0	µA	V_{in} = 0.2 - 0.2 V
Input pull-up MOS current	Ports A to E	$-I_P$	10	—	300	µA	V_{in} = 0 V

Notes: 1. The regular specifications are supported in the H8S/2506 Group only.

2. If the A/D and D/A converters are not used, do not leave the AV_{CC}, V_{ref}, and AV open. Apply a voltage 3.0 V to 5.5 V to the AV_{CC} and V_{ref} pins by connecting th for instance. Set $V_{ref} \leq AV_{CC}$.

3. P35/SCK1/SCL0 and P34/SDA0 are NMOS push/pull outputs.
   To output high level signal, pull-up resistance must be connected externally.

4. When ICE = 0. To output low when bus drive function is selected is determine 24.4, Bus Drive Characteristics.

5. HRxD and HTxD are supported in the H8S/2556 Group only.

6. When $P1V_{CC}$ = 5.0 V ±0.5 V (BUFGC1 in ICPCR is 0), $P1V_{CC}$ = 3.3 V ±0.3 V (I in ICPCR is 1), $P2V_{CC}$ = 5.0 V ±0.5 V (BUFGC2 in ICPCR is 0), and $P2V_{CC}$ = 3 V (BUFGC2 in ICPCR is 1).

	All input pins except $\overline{\text{RES}}$, NMI, and P32 to P35		—	—	15	pF	
Current consumption*3	Normal operation	I_{CC}*4	—	34 V_{CC} = 5.0 V	50 V_{CC} = 5.5 V	mA	f = 26 MHz
	Sleep mode		—	26 V_{CC} = 5.0 V	40 V_{CC} = 5.5 V	mA	f = 26 MHz
	All modules stopped		—	20 V_{CC} = 5.0 V	—	mA	f = 26 MHz (reference
	Medium-speed mode (ϕ/32)		—	22 V_{CC} = 5.0 V	—	mA	f = 26 MHz (reference
	Watch mode		—	30 V_{CC} = 5.0 V	150 V_{CC} = 5.5 V	µA	Using 32.7 crystal reso
	Standby mode		—	22 V_{CC} = 5.0 V	100 V_{CC} = 5.5 V	µA	$T_a \leq 50°C$ 32.768 kHz
			—	—	140 V_{CC} = 5.5 V		$50°C < T_a$ 32.768 kHz
Analog power supply current	During A/D conversion, D/A conversion	AI_{CC}	—	3.0	5.0	mA	
	Waiting for A/D conversion, D/A conversion		—	2.0	10	µA	

Notes: 1. The regular specifications are supported in the H8S/2506 Group only.

2. If the A/D and D/A converters are not used, do not leave the AV_{CC}, V_{ref}, and A open. Apply a voltage 3.0 V to 5.5 V to the AV_{CC} and V_{ref} pins by connecting th for instance. Set $V_{ref} \leq AV_{CC}$.

3. Current consumption values are for $P1V_{CC} = P2V_{CC} = AV_{CC} = V_{CC}$, V_{IH} min. = V_C V_{IL} max. = 0.2 V with all output pins unloaded and the on-chip pull-up MOS in state.

4. I_{CC} depends on V_{CC} and f as follows:
   I_{CC} max. = –9.49 (mA) + 5.31 (mA/V) × Vcc + 1.00 (mA/MHz) × f + 0.03 (mA/(M V_{CC} × f (normal operation)
   I_{CC} max. = –8.92 (mA) + 5.22 (mA/V) × Vcc + 0.63 (mA/MHz) × f + 0.027 (mA/(× V_{CC} × f (sleep mode)

	All input pins except $\overline{\text{RES}}$, NMI, and P32 to P35		—	—	15	pF	
Current consumption*2	Normal operation	I_{CC}*3	—	28 V_{CC} =5.0 V	43 V_{CC} = 5.5 V	mA	f = 20 MHz
	Sleep mode		—	22 V_{CC} = 5.0 V	34 V_{CC} = 5.5 V	mA	f = 20 MHz
	All modules stopped		—	16 V_{CC} = 5.0 V	—	mA	f = 20 MHz (reference
	Medium-speed mode (ϕ/32)		—	18 V_{CC} = 5.0 V	—	mA	f = 20 MHz (reference
	Watch mode		—	30 V_{CC} = 5.0 V	150 V_{CC} = 5.5 V	µA	Using 32.7 crystal reso
	Standby mode		—	22 V_{CC} = 5.0 V	100 V_{CC} = 5.5 V	µA	$T_a \leq 50°C$ 32.768 kHz
			—	—	140 V_{CC} = 5.5 V		$50°C < T_a$ 32.768 kHz
Analog power supply current	During A/D conversion, D/A conversion	AI_{CC}	—	3.0	5.0	mA	
	Waiting for A/D conversion, D/A conversion		—	2.0	10	µA	

Notes: 1. If the A/D and D/A converters are not used, do not leave the AV_{CC}, V_{ref}, and AV open. Apply a voltage 3.0 V to 5.5 V to the AV_{CC} and V_{ref} pins by connecting th for instance. Set $V_{ref} \leq AV_{CC}$.

2. Current consumption values are for $P1V_{CC} = P2V_{CC} = AV_{CC} = V_{CC}$, V_{IH} min. = V_C V_{IL} max. = 0.2 V with all output pins unloaded and the on-chip pull-up MOS in state.

3. I_{CC} depends on V_{CC} and f as follows:

   I_{CC} max. = –9.49 (mA) + 5.31 (mA/V) × Vcc + 1.00 (mA/MHz) × f + 0.03 (mA/(M V_{CC} × f (normal operation)

   I_{CC} max. = –8.58 (mA) + 5.04 (mA/V) × Vcc + 0.60 (mA/MHz) × f + 0.026 (mA/(× V_{CC} × f (sleep mode)

Permissible output low current (total)	Total of all output pins	$\sum I_{OL}$	—	—	80	mA
Permissible output high current (per pin)	All output pins	$-I_{OH}$	—	—	1.0	mA
Permissible output high current (total)	Total of all output pins	$\sum -I_{OH}$	—	—	40	mA

Notes: To protect chip reliability, do not exceed the output current values in table 24.3.

* The regular specifications are supported in the H8S/2506 Group only.

						V/ 3.3 V ± 0.
	$VT^+ - VT^-$	$P2V_{CC} \times 0.05$	—	—		$P2V_{CC}$ = 5 V
Input high voltage	V_{IH}	$P2V_{CC} \times 0.7$	—	$P2V_{CC} + 0.5$	V	$P2V_{CC}$ = 5 V/ 3.3 V ± 0.
Input low voltage	V_{IL}	-0.5	—	$P2V_{CC} \times 0.25$	V	$P2V_{CC}$ = 5 V/ 3.3 V ± 0.
Output low voltage	V_{OL}	—	—	0.5	V	I_{OL} = 8 m
		—	—	0.4		I_{OL} = 3 m
Input capacitance	C_{IN}	—	—	20	pF	V_{IN} = 0 V, f = 8 MH T_a=25°C
Three-state leakage current (off state)	$\lvert I_{STI} \rvert$	—	—	1.0	µA	V_{IN}=0.2 to V

Notes: 1. The regular specifications are supported in the H8S/2506 Group only.

2. If the A/D and D/A converters are not used, do not leave the AV_{CC}, V_{ref} , and A open. Apply a voltage 5.0 V ±0.5 V/3.3 V ±0.3 V to the AV_{CC} and V_{ref} pins by con them to V_{CC}, for instance. Set $V_{ref} \leq AV_{CC}$

3. Test Conditions are for $P1V_{CC} = P2V_{CC} = AV_{CC} = V_{CC}$.

Figure 24.2 Output Load Circuit

24.4.1 Power-On/Off Timing

Table 24.5 Power-On/Off Timing

Condition A: V_{CC} = 3.0 V to 5.5 V, $P1V_{CC}$ = 3.0 V to 5.5 V, $P2V_{CC}$ = 3.0 V to 5.5 V, AV_{CC} = 3.0 V to 5.5 V, V_{ref} = 3.0 V to AV_{CC}, V_{SS} = AV_{SS} = 0 V, T_a = –20°C to +75°C (regular specifications)*, T_a = –40°C to +85°C (wide-range specific

Item	Symbol	Min.	Typ.	Max.	Unit	Test Cond
Time taken to switch V_{CC} on	t_{VCCS}	–1	—	—	ms	Figure 24.
V_{CC} hold time when PV_{CC} is switched off	t_{VCCH}	–1	—	—	ms	
V_{CC} start voltage	$V_{CCSTART}$	—	0	0.8	V	Figure 24.
V_{CC} rising gradient	SV_{CC}	—	—	20	ms/V	

Note: * The regular specifications are supported in the H8S/2506 Group only.

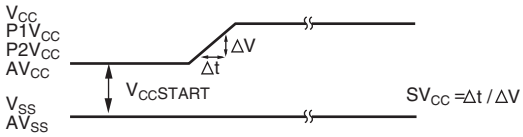


Figure 24.4 Power-On Timing

specifications)

Item		Symbol	Min.	Max.	Unit	Test Cond
Clock cycle time		t_{cyc}	38	125	ns	Figure 24.5
Clock high pulse width		t_{CH}	12	—	ns	
Clock low pulse width		t_{CL}	12	—	ns	
Clock rise time		t_{Cr}	—	5	ns	
Clock fall time		t_{Cf}	—	5	ns	
EXTAL clock input frequency	PLL1 multiplication	f_{EX}	8	26	MHz	Figure 21.5
	PLL2 multiplication		8	13		
Clock oscillator settling time at reset (crystal)		t_{OSC1}	20	—	ms	Figure 24.6
Clock oscillator settling time at reset (external clock)			20	—	ms	
Clock oscillator settling time in software standby (crystal)		t_{OSC2}	8	—	ms	Figure 22.3
External clock settling delay time		t_{DEXT}	8	—	ms	Figure 24.6
Subclock oscillator settling time		t_{OSC3}	2	—	s	
Subclock oscillator frequency		f_{SUB}		32.768	kHz	
Subclock (ϕ_{SUB}) cycle time		t_{SUB}		30.5	μs	

Note: * The regular specifications are supported in the H8S/2506 Group only.

Clock low pulse width		t_{CL}	18	—	ns	
Clock rise time		t_{Cr}	—	5	ns	
Clock fall time		t_{Cf}	—	5	ns	
EXTAL clock input frequency	PLL1 multiplication	f_{EX}	8	20	MHz	Figure 21
	PLL2 multiplication		8	10		
Clock oscillator settling time at reset (crystal)		t_{OSC1}	20	—	ms	Figure 24
Clock oscillator settling time at reset (external clock)			20	—	ms	
Clock oscillator settling time in software standby (crystal)		t_{OSC2}	8	—	ms	Figure 22
External clock settling delay time		t_{DEXT}	8	—	ms	Figure 24
Subclock oscillator settling time		t_{OSC3}	2	—	s	
Subclock oscillator frequency		f_{SUB}		32.768	kHz	
Subclock (ϕ_{SUB}) cycle time		t_{SUB}		30.5	µs	

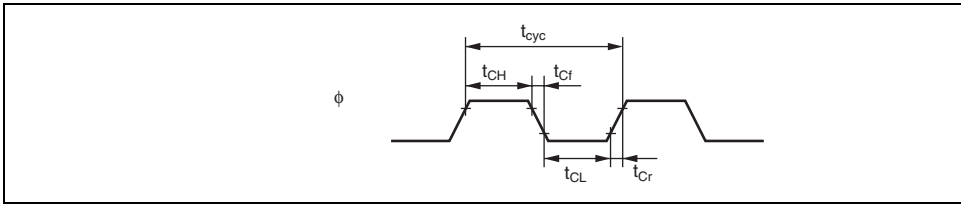


Figure 24.5 System Clock Timing

Figure 24.6 Oscillator Settling Timing

24.4.3 Control Signal Timing

Table 24.7 lists the control signal timing.

Table 24.7 Control Signal Timing

Condition A (for H8S/2552 Group, H8S/2506 Group):
V_{CC} = $P1V_{CC}$ = $P2V_{CC}$ = 5.0 V ± 0.5 V (BUFGC1 and BUFGC2 in ICPCR are 0)/
3.3 V ± 0.3 V (BUFGC1 and BUFGC2 in ICPCR are 1),
AV_{CC} = 3.0 V to 5.5 V, V_{ref} = 3.0 V to AV_{CC}, V_{SS} = AV_{SS} = 0 V, ϕ= 32.768 kHz,
8 to 26 MHz, T_a = –20°C to +75°C (regular specifications)*, T_a = –40°C to +85°C (wide-specifications)

Condition B (for H8S/2556 Group):
V_{CC} = $P1V_{CC}$ = $P2V_{CC}$ = 5.0 V ± 0.5 V (BUFGC1 and BUFGC2 in ICPCR are 0)/
3.3 V ± 0.3 V (BUFGC1 and BUFGC2 in ICPCR are 1),
AV_{CC} = 3.0 V to 5.5 V, V_{ref} = 3.0 V to AV_{CC}, V_{SS} = AV_{SS} = 0 V, ϕ= 32.768 kHz,
8 to 20 MHz, Ta = –40°C to +85°C (wide-range specifications)

Item	Symbol	Min.	Max.	Unit	Test Cond
$\overline{RES}$ setup time	t_{RESS}	250	—	ns	Figure 24.7
$\overline{RES}$ pulse width	t_{RESW}	20	—	t_{cyc}	
$\overline{MRES}$ setup time	t_{MRESS}	250	—	ns	
$\overline{MRES}$ pulse width	t_{MRESW}	20	—	t_{cyc}	

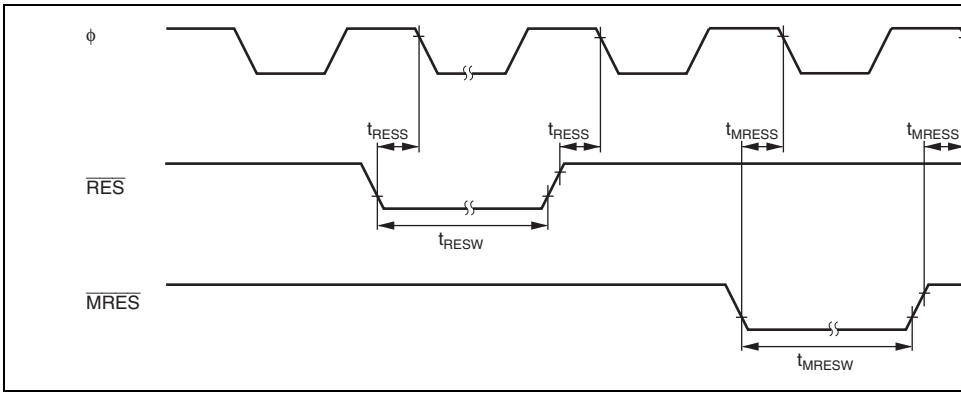


Figure 24.7 Reset Input Timing

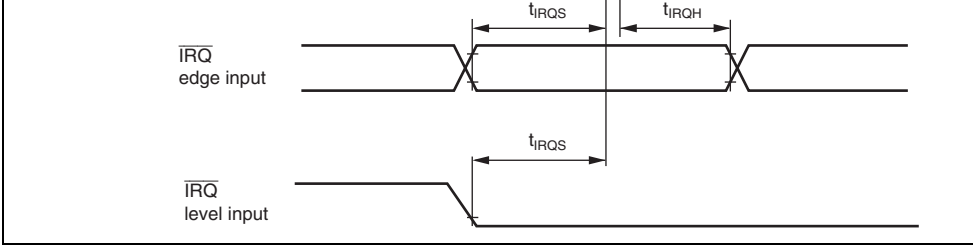


Figure 24.8 Interrupt Input Timing

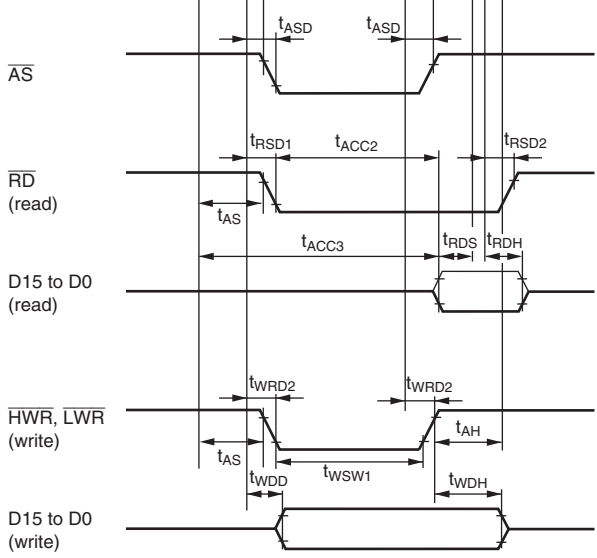
specifications)

Condition B (for H8S/2556 Group):
V_{CC} = P1V_{CC} = P2V_{CC} = 5.0 V ± 0.5 V (BUFGC1 and BUFGC2 in ICPCR are 0)/ 3.3 V ± 0.3 V (BUFGC1 and BUFGC2 in ICPCR are 1),
AV_{CC} = 3.0 V to 5.5 V, V_{ref} = 3.0 V to AV_{CC}, V_{SS} = AV_{SS} = 0 V, φ= 32.768 kHz, 8 to 20 MHz, Ta = –40°C to +85°C (wide-range specifications)

		Condition A		Condition B			Tes
Item	**Symbol**	**Min.**	**Max.**	**Min.**	**Max.**	**Unit**	**Co**
Address delay time	t_{AD}	—	30	—	35	ns	Fig to
Address setup time	t_{AS}	0.5 × t_{cyc} - 15	—	0.5 × t_{cyc} - 15	—	ns	
Address hold time	t_{AH}	0.5 × t_{cyc} - 8	—	0.5 × t_{cyc} - 8	—	ns	
$\overline{CS}$ delay time	t_{CSD}	—	30	—	35	ns	
$\overline{AS}$ delay time	t_{ASD}	—	25	—	25	ns	
$\overline{RD}$ delay time 1	t_{RSD1}	—	25	—	25	ns	
$\overline{RD}$ delay time 2	t_{RSD2}	—	25	—	25	ns	
Read data setup time	t_{RDS}	30	—	30	—	ns	
Read data hold time	t_{RDH}	10	—	10	—	ns	
Read data access time 2	t_{ACC2}	—	1.5 × t_{cyc} - 40	—	1.5 × t_{cyc} - 40	ns	
Read data access time 3	t_{ACC3}	—	2.0 × t_{cyc} - 40	—	2.0 × t_{cyc} - 50	ns	

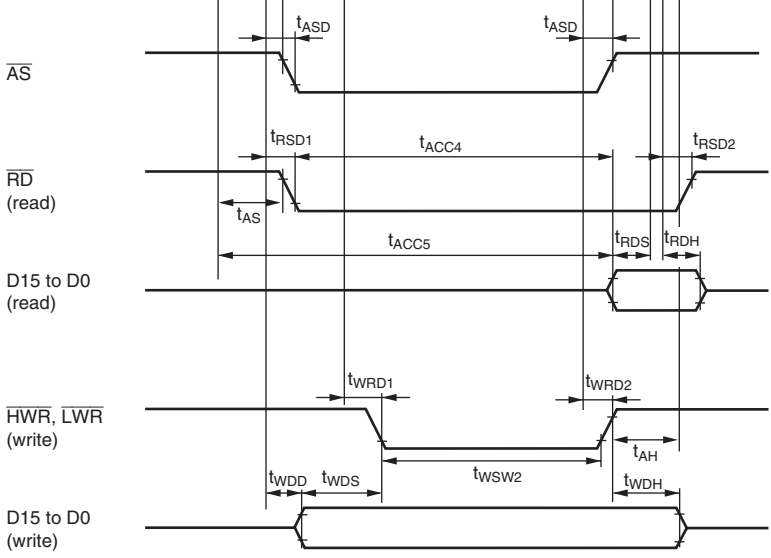
$\overline{WR}$ pulse width 2	t_{WSW2}	1.5 × t_{cyc} - 15	—	1.5 × t_{cyc} - 20	—	ns	
Write data delay time	t_{WDD}	—	25	—	40	ns	
Write data setup time	t_{WDS}	0.5 × t_{cyc} - 19	—	0.5 × t_{cyc} - 25	—	ns	Figu
Write data hold time	t_{WDH}	0.5 × t_{cyc} - 12	—	0.5 × t_{cyc} - 20	—	ns	
$\overline{WAIT}$ setup time	t_{WTS}	25	—	25	—	ns	Figu
$\overline{WAIT}$ hold time	t_{WTH}	10	—	10	—	ns	
$\overline{BREQ}$ setup time	t_{BRQS}	25	—	30	—	ns	Figu
$\overline{BACK}$ delay time	t_{BACD}	—	25	—	40	ns	
Bus floating time	t_{BZD}	—	38	—	50	ns	

Note: * The regular specifications are supported in the H8S/2506 Group only.



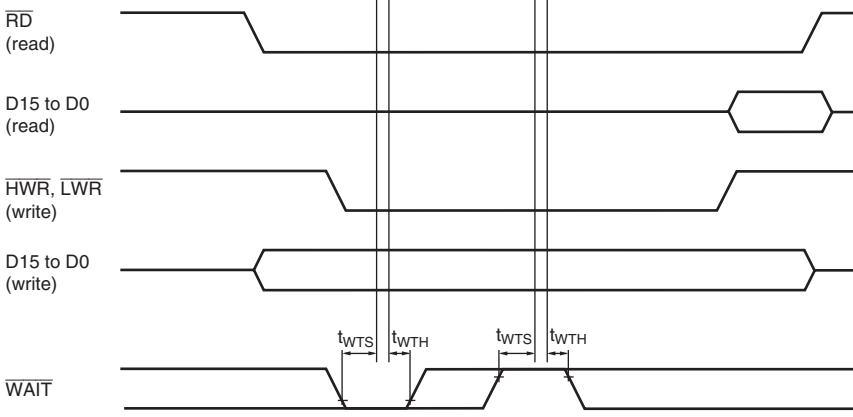
Note: * $\overline{CS1}$ and $\overline{CS2}$ are not supported in the H8S/2556 Group.

Figure 24.9 Basic Bus Timing: Two-State Access



Note: * $\overline{CS1}$ and $\overline{CS2}$ are not supported in the H8S/2556 Group.

Figure 24.10 Basic Bus Timing: Three-State Access



Note: * CS1 and CS2 are not supported in the H8S/2556 Group.

Figure 24.11 Basic Bus Timing: Three-State Access, One Wait

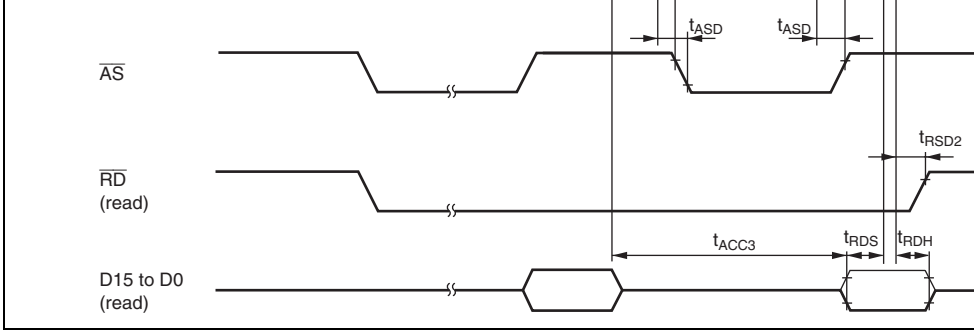


Figure 24.12 Burst ROM Access Timing: Two-State Access

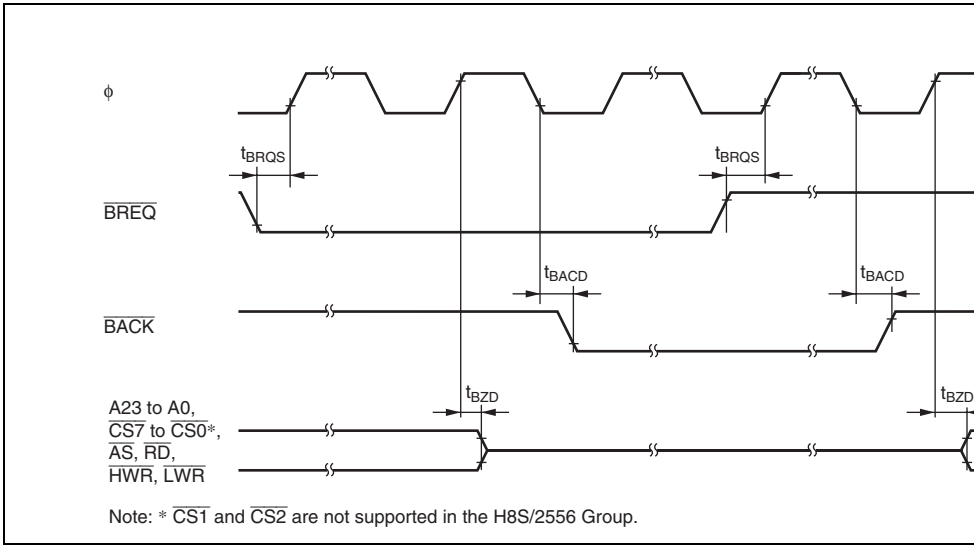


Figure 24.13 External Bus-Released Timing

specifications)

Condition B (for H8S/2556 Group):
V_{CC} = $P1V_{CC}$ = $P2V_{CC}$ = 5.0 V ± 0.5 V (BUFGC1 and BUFGC2 in ICPCR are 0)/ 3.3 V ± 0.3 V (BUFGC1 and BUFGC2 in ICPCR are 1),
AV_{CC} = 3.0 V to 5.5 V, V_{ref} = 3.0 V to AV_{CC}, V_{SS} = AV_{SS} = 0 V, ϕ= 32.768 kHz, 8 to 20 MHz, Ta = –40°C to +85°C (wide-range specifications)

				Condition A		Condition B			Te
Item			**Symbol**	**Min.**	**Max.**	**Min.**	**Max.**	**Unit**	**Co**
I/O ports *3	Output data delay time		t_{PWD}	—	38	—	50	ns	Fi
	Input data setup time		t_{PRS}	28	—	30	—		
	Input data hold time		t_{PRH}	28	—	30	—		
TPU	Timer output delay time		t_{TOCD}	—	38	—	50	ns	Fi
	Timer input setup time		t_{TICS}	28	—	30	—		
	Timer clock input setup time		t_{TCKS}	28	—	30	—	ns	Fi
	Timer clock pulse width	Single edge	t_{TCKWH}	1.5	—	1.5	—	t_{cyc}	
		Both edges	t_{TCKWL}	2.5	—	2.5	—		
TMR	Timer output delay time		t_{TMOD}	—	38	—	50	ns	Fi
	Timer reset input setup time		t_{TMRS}	28	—	30	—	ns	Fi
	Timer clock input setup time		t_{TMCS}	28	—	30	—	ns	Fi
	Timer clock pulse width	Single edge	t_{TMCWH}	1.5	—	1.5	—	t_{cyc}	
		Both edges	t_{TMCWL}	2.5	—	2.5	—		
WDT_1	BUZZ output delay time		t_{BUZD}	—	38	—	50	ns	Fi

	(synchronous)							
	Receive data hold time (synchronous)	t_{RXH}	38	—	50	—	ns	
A/D converter	Trigger input setup time	t_{TRGS}	28	—	30	—	ns	Figu
HCAN*2	Transmit data delay time	t_{HTXD}	—	—	—	50	ns	Figu
	Transmit data setup time	t_{HRXS}	—	—	30	—	ns	
	Transmit data hold time	t_{HRXH}	—	—	30	—	ns	

Notes: 1. The regular specifications are supported in the H8S/2506 Group only.

2. (For H8S/2556 Group) The HCAN input signal is asynchronous, but checked a been changed at the ϕ clock rise (two clock intervals) shown in figure 24.24. Th output signal is asynchronous, but it changes at the ϕ clock rise (two clock inte shown in figure 24.24.

3. The P35/SCK1/SCK4 and P34 pins are driven high by NMOS. To output high, up resistor should be connected externally.

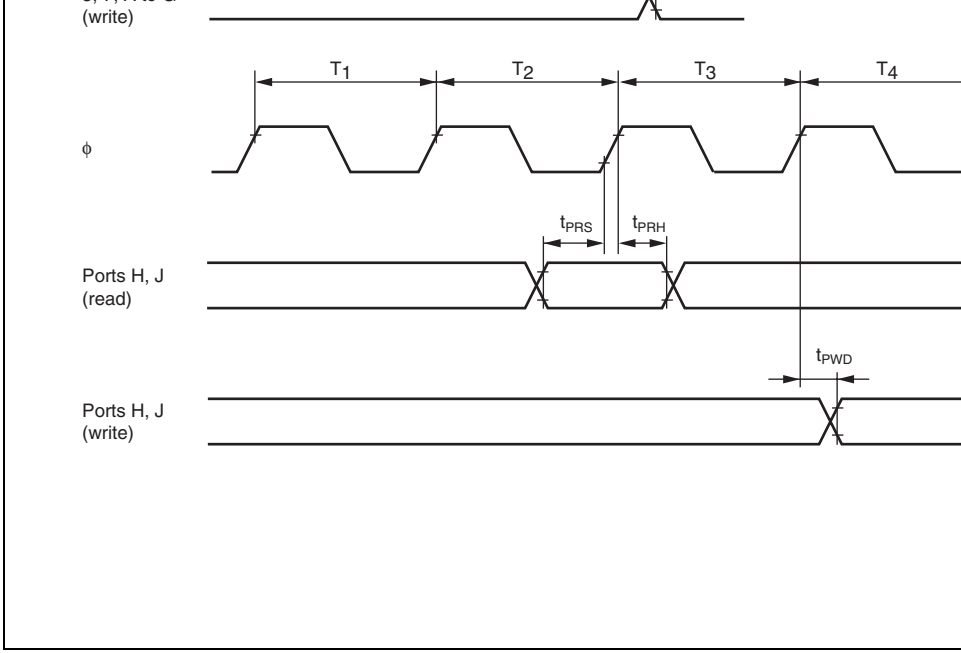


Figure 24.14 I/O Port Input/Output Timing

Figure 24.15 TPU Input/Output Timing

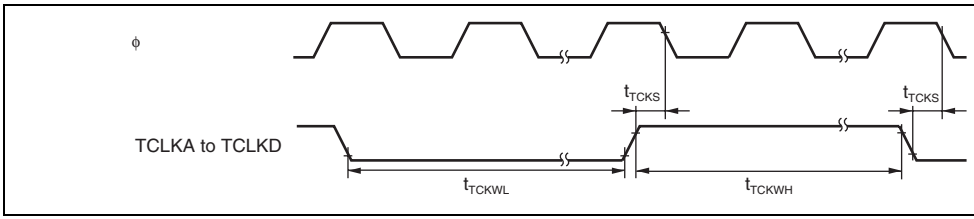


Figure 24.16 TPU Clock Input Timing

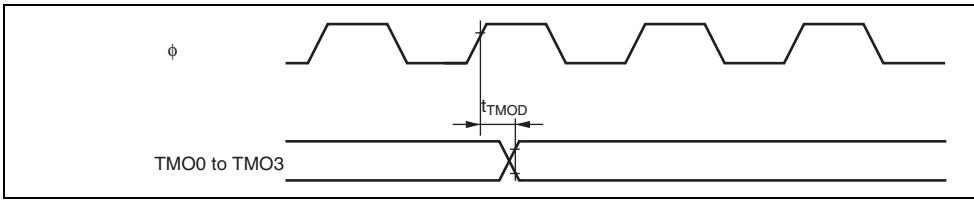


Figure 24.17 8-Bit Timer Output Timing

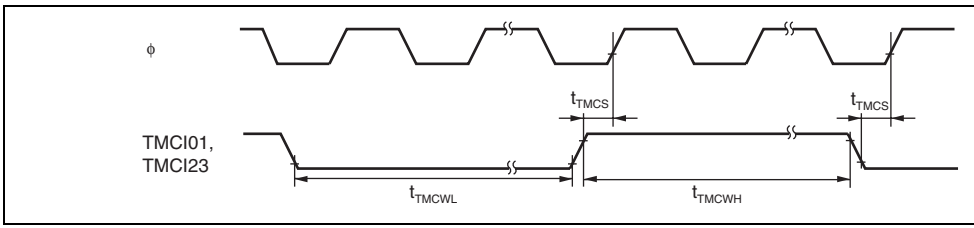


Figure 24.18 8-Bit Timer Clock Input Timing

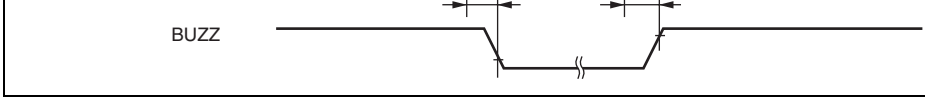


Figure 24.20 WDT_1 Output Timing

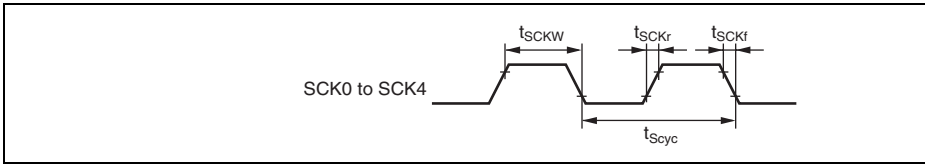


Figure 24.21 SCK Clock Input Timing

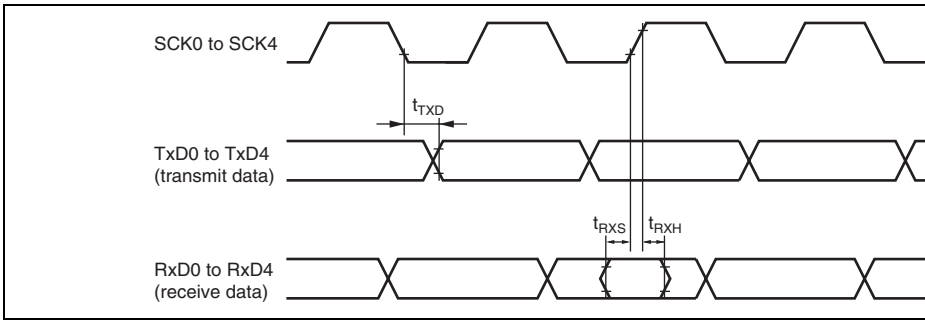


Figure 24.22 SCI Input/Output Timing/Synchronous Mode

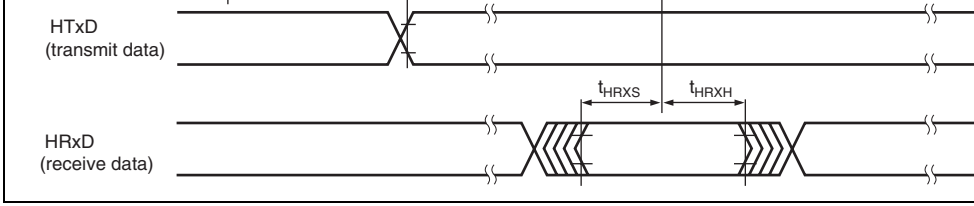


Figure 24.24 HCAN Input/Output Timing

SCL, SDA input spike pulse elimination time	t_{SP}	—	—	$1t_{cyc}$	ns
SDA input bus free time	t_{BUF}	$5t_{cyc}$	—	—	ns
Start condition input hold time	t_{STAH}	$3t_{cyc}$	—	—	ns
Retransmission start condition input setup time	t_{STAS}	$3t_{cyc}$	—	—	ns
Stop condition input setup time	t_{STOS}	$3t_{cyc}$	—	—	ns
Data input setup time	t_{SDAS}	$1t_{cyc}$ + 20	—	—	ns
Data input hold time	t_{SDAH}	0*	—	—	ns
SCL, SDA load capacitance	C_b	0	—	400	pF
SCL, SDA output fall time	t_{Sf}	20 + 0.1 Cb	—	250	ns

Note: * Configure the system in which the SDA data input hold time (t_{SDAH}) to SCL low
0.25 × V_{CC}) should be 0 ns or longer since the SDA is not changed while the S
is high.

Note: * S, P, and Sr indicate the following conditions.
S: Start condition
P: Stop condition
Sr: Retransmission start condition

Figure 24.25 I²C Bus Interface 2 Input/Output Timing

specifications)

Condition B (for H8S/2556 Group):
V_{CC} = $P1V_{CC}$ = $P2V_{CC}$ = 5.0 V ± 0.5 V (BUFGC1 and BUFGC2 in ICPCR are 0)/
3.3 V ± 0.3 V (BUFGC1 and BUFGC2 in ICPCR are 1),
AV_{CC} = 3.0 V to 5.5 V, V_{ref} = 3.0 V to AV_{CC}, V_{SS} = AV_{SS} = 0 V, ϕ= 32.768 kHz,
8 to 20 MHz, Ta = –40°C to +85°C (wide-range specifications)

Item	Min.	Typ.	Max.	Uni
Resolution	10	10	10	bits
Conversion time	9.8	—	—	µs
Analog input capacitance	—	—	20	pF
Permissible signal-source impedance	—	—	5	kΩ
Non-linearity error	—	—	±6.0	LSE
Offset error	—	—	±4.0	LSE
Full-scale error	—	—	±4.0	LSE
Quantization error	—	—	±0.5	LSE
Absolute accuracy	—	—	±8.0	LSE

Note: * The regular specifications are supported in the H8S/2506 Group only.

specifications)

Condition B (for H8S/2556 Group):
V_{CC} = $P1V_{CC}$ = $P2V_{CC}$ = 5.0 V ± 0.5 V (BUFGC1 and BUFGC2 in ICPCR are 0)/
3.3 V ± 0.3 V (BUFGC1 and BUFGC2 in ICPCR are 1),
AV_{CC} = 3.0 V to 5.5 V, V_{ref} = 3.0 V to AV_{CC}, V_{SS} = AV_{SS} = 0 V, ϕ= 32.768 kHz,
8 to 20 MHz, Ta = –40°C to +85°C (wide-range specifications)

Item	Min.	Typ.	Max.	Unit	Test Conditions
Resolution	8	8	8	bits	
Conversion time	—	—	10	µs	Load capacitance: 20 pF
Absolute accuracy*[2]	—	±2.0	±3.0	LSB	Load resistance: 2 MΩ
	—	—	±2.0	LSB	Load resistance: 4 MΩ

Notes: 1. The regular specifications are supported in the H8S/2506 Group only.
2. Except module stop, software standby, and watch modes.

specifications)

Condition B (for H8S/2556 Group):
V_{CC} = $P1V_{CC}$ = $P2V_{CC}$ = 5.0 V ± 0.5 V (BUFGC1 and BUFGC2 in ICPCR are 0)/
3.3 V ± 0.3 V (BUFGC1 and BUFGC2 in ICPCR are 1),
AV_{CC} = 3.0 V to 5.5 V, V_{ref} = 3.0 V to AV_{CC}, V_{SS} = AV_{SS} = 0 V, ϕ= 32.768 kHz,
8 to 20 MHz, Ta = –40°C to +85°C (wide-range specifications)

Item	Symbol	Min.	Typ.	Max.	Unit	Test Con
Programming time*[1]*[2]*[4]	t_p	—	3	30	ms/128 bytes	
Erase time*[1]*[3]	t_E	—	80	800	ms/4 kbytes	
		—	500	5000	ms/32 kbytes	
		—	1000	10000	ms/64 kbytes	
Programming time (total) *[1]*[2]*[4]	$\sum_{IP}$	—	7.5	22.5	s/384 kbytes	T_a = 0
		—	10	30	s/512 kbytes	
Erase time (total) *[1]*[2]*[4]	$\sum_{IE}$	—	7.5	22.5	s/384 kbytes	T_a =
		—	10	30	s/512 kbytes	
Programming/Erase time (total) *[1]*[2]*[4]	$\sum_{IPE}$	—	15	45	s/384 kbytes	T_a =
		—	20	60	s/512 kbytes	
Count of reprogramming	N_{WEC}	100*[3]	—	—	Times	
Data hold time*[4]	t_{DRP}	10	—	—	Year	

Notes: 1. Programming/Erase time depends on the data.
2. Programming/Erase time does not include the data transfer time.
3. The minimum times that all characteristics after reprogramming are guarante range between 1 and a minimum value is guaranteed.)
4. Data hold characteristics are when reprogramming is performed within the ra specifications including a minimum value.
5. The regular specifications are supported in the H8S/2506 Group only.

Port 3	6, 7	T	keep	T	keep	keep	I/O
Port 4	6, 7	T	T	T	T	T	Inp
Port 5	6, 7	T	keep	T	keep	keep	I/O
P77 to P74	6, 7	T	keep	T	keep	keep	I/O
P73/TMO1/$\overline{CS7}$	7	T	keep	T	keep	keep	I/O
P72/TMO0/$\overline{CS6}$	6	T	keep	T	[DDR • OPE = 0]	T	[DD
P71/TMRI23/ TMCI23/$\overline{CS5}$					T		Inp
					[DDR • OPE = 1]		[DD
P70/TMRI01/ TMCI01/$\overline{CS4}$					H		$\overline{CS}$
P97/DA1	6, 7	T	T	T	[DAOEn = 1]	keep	Inp
P96/DA0					keep		
					[DAOEn = 0]		
					T		
P95 to P90	6, 7	T	T	T	T	T	Inp
Port A	7	T	Keep	T	keep	keep	I/O
Address output selection with AEn bit	6	T	Keep	T	[OPE = 0]	T	Ad
					T		
					[OPE = 1]		
					keep		
Port selection	6	T	keep	T	keep	keep	I/O

		selection						
Port C		6	T	keep	T	[DDR • OPE = 0]	T	[DDR
						T		Input
						[DDR • OPE = 1]		[DDR
						keep		Addr
		7	T	keep	T	keep	keep	I/O p
Port D		6	T	T	T	T	T	Data
		7	T	keep	T	keep	keep	I/O p
Port E	8-bit bus	6	T	keep	T	keep	keep	I/O p
	16-bit bus	6	T	T	T	T	T	Data
		7	T	keep	T	keep	keep	I/O p
PF7/ϕ		6	Clock output*3	[DDR = 0]	T	[DDR = 0]	[DDR = 0]	[DDR
				Input port		Input port	Input port	Input
				[DDR = 1]		[DDR = 1]	[DDR = 1]	[DDR
				Clock output		H	Clock output	Clock
		7	T	keep	T	[DDR = 0]	[DDR = 0]	[DDR
						Input port	Input port	Input
						[DDR = 1]	[DDR = 1]	[DDR
						H	Clock output	Clock

ADTRG/IRQ3							
	8-bit bus	6	T	keep	T	keep	keep	I/O
		7	T	keep	T	keep	keep	I/O
	16-bit bus	6		H	T	[OPE = 0] T [OPE = 1] H	T	$\overline{LW}$
		7	T	keep	T	keep	keep	I/O
PF2/$\overline{WAIT}$		6	T	keep	T	[WAITE = 0] keep [WAITE = 1] T	[WAITE = 0] keep [WAITE = 1] T	[W I/O [W $\overline{WA}$
		7	T	keep	T	keep	keep	I/O
PF1/$\overline{BACK}$/BUZZ		6	T	keep	T	[BRLE = 0] keep [BRLE = 1] H	L	[BF I/O [BF $\overline{BA}$
		7	T	keep	T	keep	keep	I/O
PF0/$\overline{BREQ}$/$\overline{IRQ2}$		6	T	keep	T	[BRLE = 0] keep [BRLE = 1] T	T	[BF I/O [BF $\overline{BR}$
		7	T	keep	T	keep	keep	I/O

PG3/RX/CS1*[1]	6	T	keep	T	[DDR • OPE = 0]	T	[DDR
PG2/$\overline{TX}$/$\overline{CS2}$*[1]					T		Input
PG1/$\overline{CS3}$/$\overline{IRQ7}$					[DDR • OPE = 1]		[DDR
					H		$\overline{CS1}$
	7	T	keep	T	keep	keep	I/O p
PG0/$\overline{IRQ6}$	6, 7	T	keep	T	keep	keep	I/O p
Port H	6, 7	T	keep	T	keep	keep	I/O p
Port J	6, 7	T	keep	T	keep	keep	I/O p
THxD*[2]	6, 7	H*[3]	H	T	H	keep	THxD
HRxD*[2]	6, 7	Input	Input	T	T	Input	HRxD

Legend:

H: High level

L: Low level

T: High-impedance

Keep: Input port becomes high-impedance, output port retains state

DDR: Data direction register

OPE: Output port enable

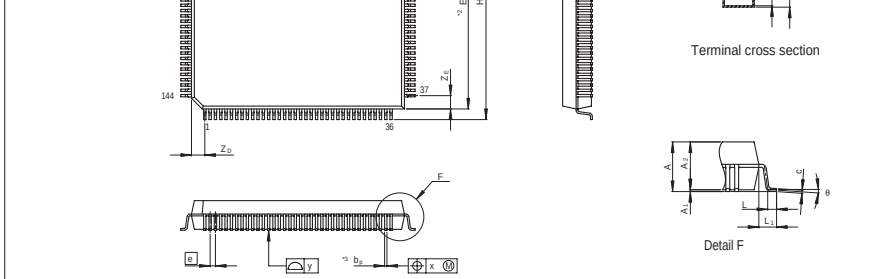
WAITE: Wait input enable

BRLE: Bus release enable

Notes: 1. PG3 and PG2 are not supported in the H8S/2556 Group.
PG3/$\overline{RX}$/$\overline{CS1}$ and PG2/$\overline{TX}$/$\overline{CS2}$ are supported in the H8S/2552 Group. (When 1, $\overline{RX}$ and $\overline{TX}$ are valid.)
PG3/$\overline{CS1}$ and PG2/$\overline{CS2}$ are supported in the H8S/2506 Group.

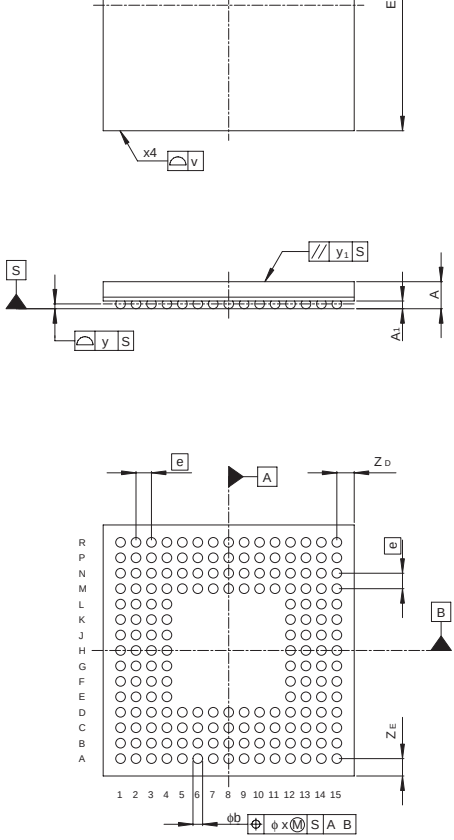
2. Supported only in the H8S/2556 Group.
3. Output pins are in the high-impedance state when the power is supplied.

H8S/2506	Flash memory version	HD64F2506	HD64F2506FC26	144-pin QFP (FP-144J,
			HD64F2506BR26	176-pin LFBGA (BP-176
H8S/2505	Flash memory version	HD64F2505	HD64F2505FC26	144-pin QFP (FP-144J,
			HD64F2505BR26	176-pin LFBGA (BP-176



Reference Symbol	Dime Mi
D	—
E	—
A_2	—
H_D	21.
H_E	21.
A	—
A_1	0.0
b_p	0.1
b_1	—
c	0.1
c_1	—
θ	0
e	—
x	—
y	—
Z_D	—
Z_E	—
L	0.
L_1	—

Figure C.1 FP-144J and FP-144JV Package Dimensions



Reference Symbol	Dimension in Millimet		
	Min	Nom	M
D	——	13.0	—
E	——	13.0	—
v	——	——	0
w	——	——	0
A	——	——	1
A_1	0.35	0.40	0
e	——	0.80	—
b	0.45	0.50	0
x	——	——	0
y	——	——	0
y_1	——	——	0
S_D	——	——	—
S_E	——	——	—
Z_D	——	0.90	—
Z_E	——	0.90	—

Figure C.2 BP-176V Package Dimensions

WAITE	0		1	
PF2DDR	0	1	—	0
Pin function	PF2 input	PF2 output	$\overline{\text{WAIT}}$ input	PF2 input

12.3.1 Timer Counter (TCNT)	380	Description added TCNT is initialized to H'00 when the TME bit in TCSR cleared to 0. To initialize TCNT to H'00 during timer operation, wri of H'00 directly to TCNT. For details, see 12.6.7, Initi of TCNT by the TME Bit.
12.6.7 Initialization of TCNT by the TME Bit	393	Newly added
13.3.7 Serial Status Register (SSR) • Normal Serial Communication Interface Mode (When SMIF in SCMR Is 0)	410	Table amended and note added

Bit	Bit Name	Initial Value	R/W	Description
7	TDRE	1	R/(W)*1	Transmit Data Register Empty Displays whether TDR contains transmit d [Setting conditions] • When the TE bit in SCR is 0 • When data is transferred from TDR to can be written to TDR [Clearing conditions] • When 0 is written to TDRE after readi • When the DTC*2 is activated by a TXI request and writes data to TDR
6	RDRF	0	R/(W)*1	Receive Data Register Full Indicates that the received data is stored i [Setting condition] • When serial reception ends normally a is transferred from RSR to RDR [Clearing conditions] • When 0 is written to RDRF after readi • When the DTC*2 is activated by an RX transfers data from RDR The RDRF flag is not affected and retains values when the RE bit in SCR is cleared If reception of the next data is completed flag is still set to 1, an overrun error will oc receive data will be lost.

				The ORER flag is not affected and retains it state when the RE bit in SCR is cleared to 0
4	FER	0	R/(W)*[1]	Framing Error Indicates that a framing error occurred durin asynchronous mode, causing abnormal terr [Setting condition] • When the stop bit is 0 In 2 stop bit mode, only the first stop bit is c value to 1; the second stop bit is not checke framing error occurs, the receive data is tra RDR but the RDRF flag is not set. Also, sub serial reception cannot be continued while t set to 1. In clocked synchronous mode, seri transmission cannot be continued, either. [Clearing condition] • When 0 is written to FER after reading F In 2-stop-bit mode, only the first stop bit is c The FER flag is not affected and retains its when the RE bit in SCR is cleared to 0.

Bit	Bit Name	Initial Value	R/W	Description
3	PER	0	R/(W)*[1]	Parity Error Indicates that a parity error occurred during using parity addition in asynchronous mode, abnormal termination. [Setting condition] • When a parity error is detected during re If a parity error occurs, the receive data is tra RDR but the RDRF flag is not set. Also, sub serial reception cannot be continued while th is set to 1. In clocked synchronous mode, se transmission cannot be continued, either. [Clearing condition] • When 0 is written to PER after reading P The PER flag is not affected and retains its pr when the RE bit in SCR is cleared to 0.

Notes:

3. To clear the flag by using the CPU, write 0 to the f then read it once again.

Bit	Bit Name	Initial Value	R/W	Description
6	RDRF	0	R/(W)*1	Receive Data Register Full Indicates that the received data is stored i [Setting condition] • When serial reception ends normally a is transferred from RSR to RDR [Clearing conditions] • When 0 is written to RDRF after readi • When the DTC*2 is activated by an RX transfers data from RDR The RDRF flag is not affected and retains values when the RE bit in SCR is cleared If reception of the next data is completed flag is still set to 1, an overrun error will oc receive data will be lost.
5	ORER	0	R/(W)*1	Overrun Error Indicates that an overrun error occurred d causing abnormal termination. [Setting condition] • When the next serial reception is comp RDRF = 1 The receive data prior to the overrun erro RDR, and the data received subsequently subsequent serial cannot be continued wh flag is set to 1. In clocked synchronous m transmission cannot be continued, either. [Clearing condition] • When 0 is written to ORER after readi The ORER flag is not affected and retains state when the RE bit in SCR is cleared to
4	ERS	0	R/(W)*1	Error Signal Status Indicates that the status of an error, signa from the reception side at reception [Setting condition] • When the low level of the error signal [Clearing condition] • When 0 is written to ERS after reading The ERS flag is not affected and retains it when the TE bit in SCR is cleared to 0.

The PER flag is not affected and retains its
when the RE bit in SCR is cleared to 0.

416 Notes:

3. To clear the flag by using the CPU, write 0 to the f
then read it once again.

20.1.2 Operating Mode

669 Table amended and note added

Table 20.1 MD Pin Setting and Operating Mode

Pin	Reset state	On-chip ROM valid mode*1	User program mode*2	User boot mode	Boot	mode	
RES	0	1	1	1		1	
MD0*3	0/1	0/1	0/1	1		0/1	
MD1	0/1	1	1	0		1	
MD2	0/1	1	1	0		0	

Notes:

3. In case of On-chip ROM valid mode, User program
and Boot mode, when the MD0 pin sets to 0, the m
be Expanded mode, otherwise, when the pin sets
mode will be Single chip mode. However, in case
boot mode, there is no Expanded mode.

20.4.3 User Boot Mode

716 Description amended

The area that can be executed in the steps of the use
procedure program (on-chip RAM and user MAT) i
in section 20.4.4, Procedure Program and Storable Ar
Programming Data.

717 Description amended

The area that can be executed in the steps of the use
procedure program (on-chip RAM and user MAT) i
in section 20.4.4, Procedure Program and Storable Ar
Programming Data.

Item	RAM	Memory	(Expanded Mode)	User MAT
Execution of Writing SCO = 1 to FCCS (Download)	○	×	×	

Table 20.9 (3) Useable Area for Programming in User Boot Mode

722

Table amended

	Storable/Executable Area			Selecte	
Item	On-Chip RAM	User Boot MAT	User MAT	User Boot MAT	E P S
Storage Area for Program Data	○	×*1	—	—	
Operation for Selection of On-chip Program to be Downloaded	○	○		○	
Operation for Writing H'A5 to FKEY	○	○		○	
Execution of Writing SCO = 1 to FCCS (Download)	○	×			
Operation for FKEY Clear	○	○		○	
Determination of Download Result	○	○		○	
Operation for Download Error	○	○		○	
Operation for Settings of Initial Parameter	○	○		○	

Switching MATs by FMATS	○	×	○	
Operation for Writing H'5A to FKEY	○	×	○	
Operation for Settings of Program Parameter	○	×	○	
Execution of Programming	○	×	○	
Determination of Program Result	○	×	○	
Operation for Program Error	○	×[*2]	○	
Operation for FKEY Clear	○	×	○	
Switching MATs by FMATS	○	×		○

Clear					
Determination of Download Result	○	○			○
Operation for Download Error	○	○			○
Operation for Settings of Initial Parameter	○	○			○
Execution of Initialization	○	×			○
Determination of Initialization Result	○	○			○
Operation for Initialization Error	○	○			○
NMI Handling Routine	○	×			○
Operation for Interrupt Inhibit	○	○			○
Switching MATs by FMATS	○	×		○	
Operation for Writing H'5A to FKEY	○	×		○	

725 Table amended

	Storable/Executable Area			Selected MAT	
Item	**On-Chip RAM**	**User Boot MAT**	**User MAT**	**User Boot MAT**	**Embe Stora**
Operation for Settings of Erasure Parameter	○	×	○		
Execution of Erasure	○	×	○		
Determination of Erasure Result	○	×	○		
Operation for Erasure Error	○	×*	○		
Operation for FKEY Clear	○	×	○		
Switching MATs by FMATS	○	×		○	

B

D

E

R

S

Renesas 16-Bit Single-Chip Microcomputer
Hardware Manual
H8S/2556 Group, H8S/2552 Group, H8S/2506 Group

Publication Date: 1st Edition, March, 2003
Rev.6.00, September 24, 2009
Published by: Sales Strategic Planning Div.
Renesas Technology Corp.
Edited by: Customer Support Department
Global Strategic Communication Div.
Renesas Solutions Corp.

RENESAS SALES OFFICES

http://www.ren

Refer to "**http://www.renesas.com/en/network**" for the latest and detailed information.

Renesas Technology America, Inc.
450 Holger Way, San Jose, CA 95134-1368, U.S.A
Tel: <1> (408) 382-7500, Fax: <1> (408) 382-7501

Renesas Technology Europe Limited
Dukes Meadow, Millboard Road, Bourne End, Buckinghamshire, SL8 5FH, U.K.
Tel: <44> (1628) 585-100, Fax: <44> (1628) 585-900

Renesas Technology (Shanghai) Co., Ltd.
Unit 204, 205, AZIACenter, No.1233 Lujiazui Ring Rd, Pudong District, Shanghai, China 200120
Tel: <86> (21) 5877-1818, Fax: <86> (21) 6887-7858/7898

Renesas Technology Hong Kong Ltd.
7th Floor, North Tower, World Finance Centre, Harbour City, Canton Road, Tsimshatsui, Kowloon, Hong Kong
Tel: <852> 2265-6688, Fax: <852> 2377-3473

Renesas Technology Taiwan Co., Ltd.
10th Floor, No.99, Fushing North Road, Taipei, Taiwan
Tel: <886> (2) 2715-2888, Fax: <886> (2) 3518-3399

Renesas Technology Singapore Pte. Ltd.
1 Harbour Front Avenue, #06-10, Keppel Bay Tower, Singapore 098632
Tel: <65> 6213-0200, Fax: <65> 6278-8001

Renesas Technology Korea Co., Ltd.
Kukje Center Bldg. 18th Fl., 191, 2-ka, Hangang-ro, Yongsan-ku, Seoul 140-702, Korea
Tel: <82> (2) 796-3115, Fax: <82> (2) 796-2145

Renesas Technology Malaysia Sdn. Bhd
Unit 906, Block B, Menara Amcorp, Amcorp Trade Centre, No.18, Jln Persiaran Barat, 46050 Petaling Jaya, Selangor Darul Ehsan, Ma
Tel: <603> 7955-9390, Fax: <603> 7955-9510

H8S/2556 Group, H8S/2552 Group, H8S/2506 Group

Hardware Manual

Renesas Electronics Corporation
1753, Shimonumabe, Nakahara-ku, Kawasaki-shi, Kanagawa 211-8668 Japan

REJ09B0099-0600

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