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/2472, H8S/2463, H8S/2462 Group

Hardware Manual

Renesas 16-Bit Single-Chip Microcomputer
H8S Family / H8S/2400 Series

H8S/2472 R4F2472
H8S/2463 R4F2463
H8S/2462 R4F2462

Renesas Electronics
www.renesas.com

Rev.2.00 2008.08

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 dis 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 infor included in this document.
6. When using or otherwise relying on the information in this document, you should evaluate the informati light of the total system before deciding about the applicability of such information to the intended appli 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 o 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 re 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 malfunc 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 inj 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 is high. You should implement safety measures so that Renesas products may not be easily detached fro 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 writt approval from Renesas.
13. Please contact a Renesas sales office if you have any questions regarding the information contained in document, Renesas semiconductor products, or if you have any other inquiries.

RENESAS

Generally, the input pins of CMOS products are high-impedance input pins. If un
are in their open states, intermediate levels are induced by noise in the vicinity, a
through current flows internally, and a malfunction may occur.

3. Processing before Initialization

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

4. Prohibition of Access to Undefined or Reserved Addresses

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

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

The configuration of the functional description of each module differs according to t
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. Ea
includes notes in relation to the descriptions given, and usage notes are given, as require
final part of each section.

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

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

11. Index

circuits, and microcomputers.

Objective: This manual was written to explain the hardware functions and electrical characteristics of the H8S/2472 Group, H8S/2463 Group, and H8S/2462 G the target users.
Refer to the H8S/2600 Series, H8S/2000 Series Software Manual for a detailed description of the instruction set.

Notes on reading this manual:

- In order to understand the overall functions of the chip
  Read the manual according to the contents. This manual can be roughly categorized i 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 sam similar function, e.g. 16-bit timer pulse unit or serial communication, is implemented on more than one cha XXX_N (XXX is the register name and N is the chann number)

Bit order: The MSB is on the left and the LSB is on the right.

User's manuals for development tools:

Document Title	Documen
H8S, H8/300 Series C/C++ Compiler, Assembler, Optimizing Linkage Editor User's Manual	REJ10B0
Microcomputer Development Environment System H8S, H8/300 Series Simulator/Debugger User's Manual	ADE-702-
H8S, H8/300 Series High-performance Embedded Workshop 3 Tutorial	REJ10B0
H8S, H8/300 Series High-performance Embedded Workshop 3 User's Manual	REJ10B0

Section 7 Data Transfer Controller (DTC)

Section 26 Boundary Scan (JTAG)

Appendix

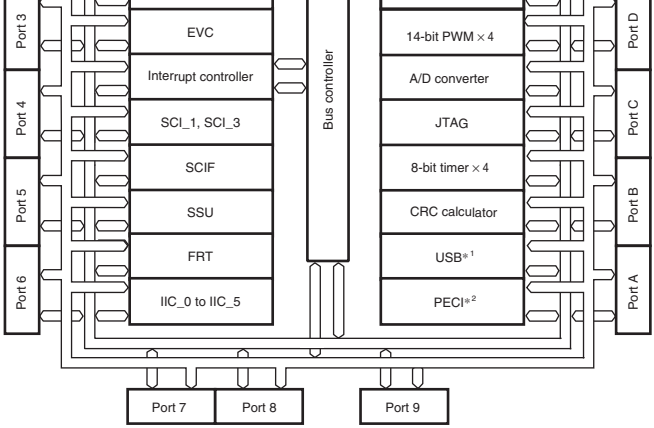
Appendix

- Various peripheral functions
  Data transfer controller (DTC)
  14-bit PWM timer (PWMX)
  16-bit free-running timer (FRT)
  8-bit timer (TMR)
  Watchdog timer (WDT)
  Asynchronous or synchronous serial communication interface (SCI)
  CRC operation circuit (CRC)
  Serial communication interface with FIFO (SCIF)
  Synchronous serial communication unit (SSU)
  I^2C bus interface (IIC)
  LPC interface (LPC)
  Ethernet controller (EtherC)
  Direct memory access controller for Ethernet controller (E-DMAC)
  USB function module (USB)*[1]
  10-bit A/D converter
  Platform Environment Control Interface (PECI)*[2]
  Boundary scan (JTAG)
  Clock pulse generator

Notes: 1. Supported only by the H8S/2472 Group.
2. Supported only by the H8S/2472 Group and the H8S/2462 Group.

- Reprogramming count: 1000 times (Tpy.)
- General I/O ports
  I/O pins: 110 (for 176-pin), 106 (for 144-pin)
  Input-only pins: 9
- Supports various power-down states
- Compact package

Package (code)	Body Size	Pin Pitch
PLBG0176GA-A	13 × 13 mm	0.8 mm
PTQP0144LC-A	16 × 16 mm	0.4 mm
PLQP0144KA-A	20 × 20 mm	0.5 mm



[Legend]

CPU: Central processing unit
DTC: Data transfer controller
EVC: Event counter
SCI: Serial communication interface
SCIF: Serial communication interface with FIFO
IIC: I^2C bus interface
EtherC: Ethernet controller
E-DMAC: Direct memory access controller for Ethernet controller
SSU: Synchronous serial communication unit
USB: USB function module
FRT: 16-bit free running timer
PWM: 14-bit PWM timer
LPC: LPC interface
WDT: Watchdog timer
JTAG: Boundary scan
PECI: PECI interface

Notes: 1. Supported only by the H8S/2472 Group.
2. Supported only by the H8S/2472 Group and the H8S/2462 Group.

Figure 1.1 Internal Block Diagram

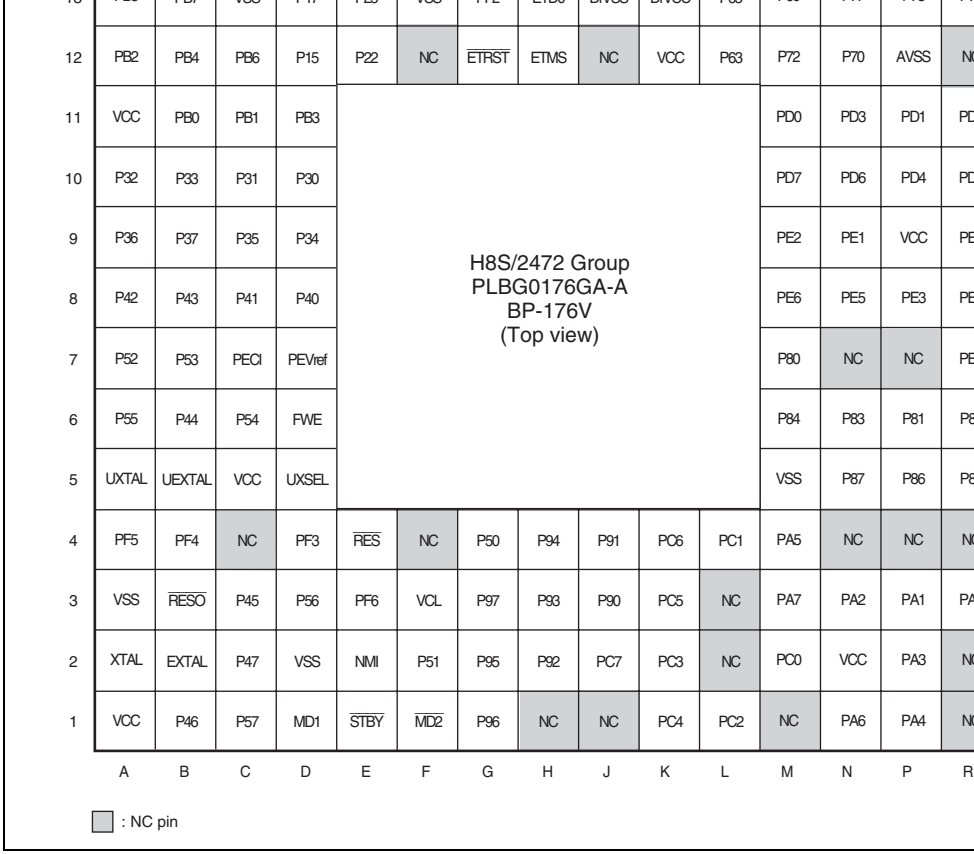


Figure 1.2 Pin Assignments (H8S/2472 Group)

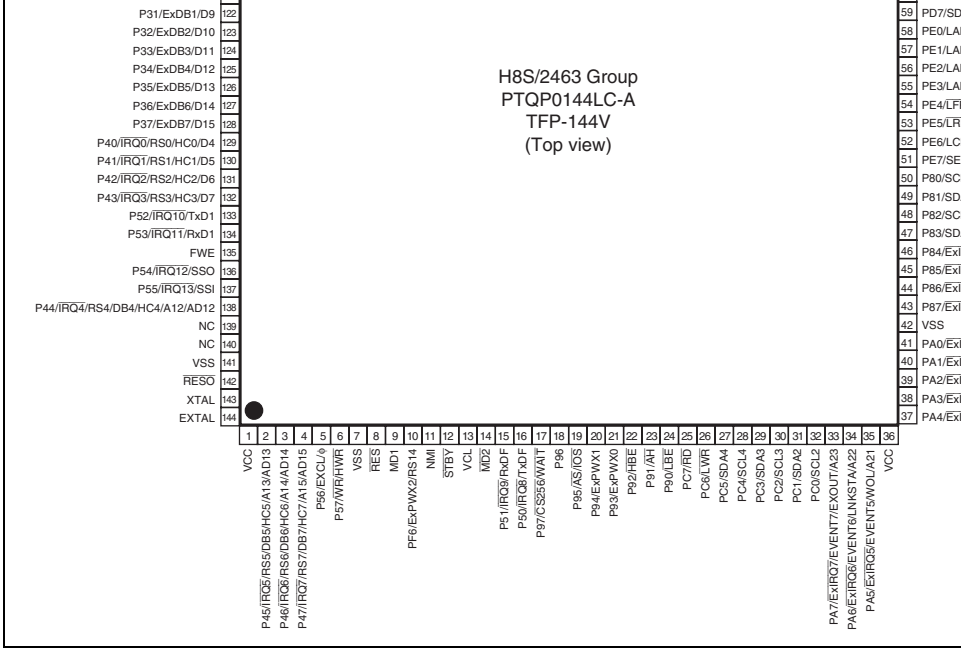


Figure 1.3 Pin Assignments (H8S/2463 Group)

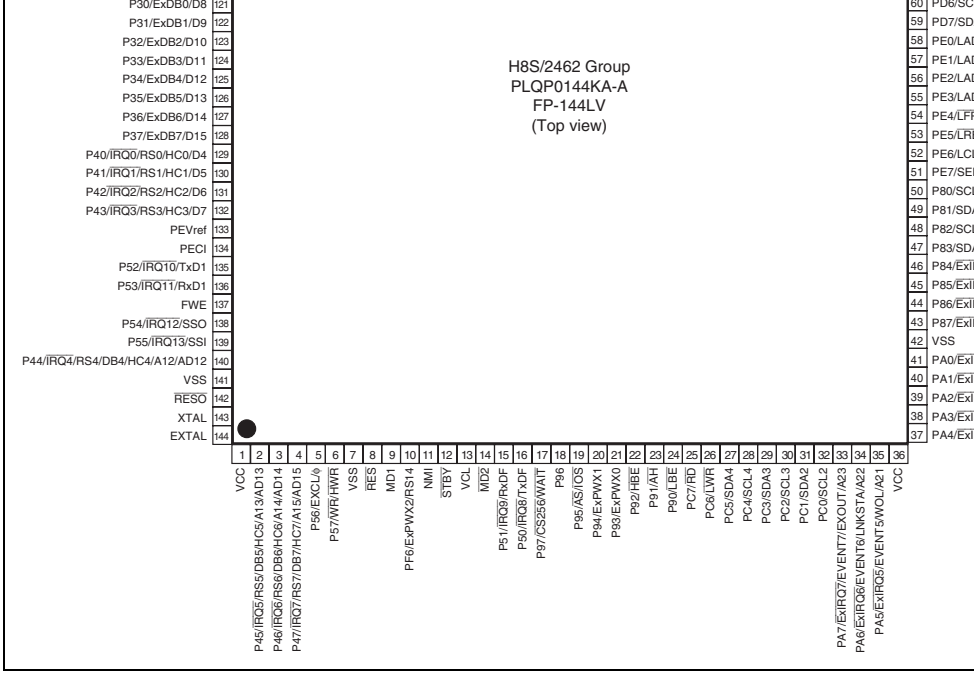


Figure 1.4 Pin Assignments (H8S/2462 Group)

3	3	B1	P46/$\overline{\text{IRQ6}}$/RS6/DB6/ HC6/A14/AD14	P46/$\overline{\text{IRQ6}}$/RS6/DB6/ HC6	FA14
4	4	C2	P47/$\overline{\text{IRQ7}}$/RS7/DB7/ HC7/A15/AD15	P47/$\overline{\text{IRQ7}}$/RS7/DB7/ HC7	FA15
5	5	D3	P56/EXCL/phi	P56/EXCL/phi	NC
6	6	C1	$\overline{\text{WR}}$/$\overline{\text{HWR}}$	P57	NC
7	7	D2	VSS	VSS	VSS
8	8	E4	$\overline{\text{RES}}$	$\overline{\text{RES}}$	$\overline{\text{RES}}$
9	9	D1	MD1	MD1	VSS
10	10	E3	PF6/ExPWX2/RS14	PF6/ExPWX2/RS14	VSS
11	11	E2	NMI	NMI	FA9
12	12	E1	$\overline{\text{STBY}}$	$\overline{\text{STBY}}$	VCC
—	—	F4	NC	NC	NC
13	13	F3	VCL	VCL	VCL
14	14	F1	$\overline{\text{MD2}}$	$\overline{\text{MD2}}$	VCC
15	15	F2	P51/$\overline{\text{IRQ9}}$/RxDF	P51/$\overline{\text{IRQ9}}$/RxDF	NC
16	16	G4	P50/$\overline{\text{IRQ8}}$/TxDF	P50/$\overline{\text{IRQ8}}$/TxDF	NC
17	17	G3	P97/$\overline{\text{CS256}}$/$\overline{\text{WAIT}}$	P97	NC
18	18	G1	P96	P96	NC
19	19	G2	$\overline{\text{AS}}$/$\overline{\text{IOS}}$	P95	NC
20	20	H4	P94/ExPWX1	P94/ExPWX1	NC
21	21	H3	P93/ExPWX0	P93/ExPWX0	NC
—	—	H1	NC	NC	NC
22	22	H2	P92/$\overline{\text{HBE}}$	P92	NC

28	28	K1	PC4/SCL4	PC4/SCL4	NC
29	29	K2	PC3/SDA3	PC3/SDA3	NC
—	—	L3	NC	NC	NC
30	30	L1	PC2/SCL3	PC2/SCL3	NC
—	—	L2	NC	NC	NC
31	31	L4	PC1/SDA2	PC1/SDA2	NC
—	—	M1	NC	NC	NC
32	32	M2	PC0/SCL2	PC0/SCL2	NC
33	33	M3	PA7/$\overline{\text{ExIRQ7}}$/ EVENT7/EXOUT/A23	PA7/$\overline{\text{ExIRQ7}}$/ EVENT7/EXOUT	VCC
34	34	N1	PA6/$\overline{\text{ExIRQ6}}$/ EVENT6/LNKSTA/A22	PA6/$\overline{\text{ExIRQ6}}$/ EVENT6/LNKSTA	VCC
35	35	M4	PA5/$\overline{\text{ExIRQ5}}$/ EVENT5/WOL/A21	PA5/$\overline{\text{ExIRQ5}}$/ EVENT5/WOL	VSS
36	36	N2	VCC	VCC	VCC
37	37	P1	PA4/$\overline{\text{ExIRQ4}}$/EVENT4/ A20	PA4/$\overline{\text{ExIRQ4}}$/EVENT4	$\overline{\text{CE}}$
38	38	P2	PA3/$\overline{\text{ExIRQ3}}$/EVENT3/ A19	PA3/$\overline{\text{ExIRQ3}}$/EVENT3	FA19
—	—	R1	NC	NC	NC
39	39	N3	PA2/$\overline{\text{ExIRQ2}}$/EVENT2/ A18	PA2/$\overline{\text{ExIRQ2}}$/EVENT2	FA18
—	—	R2	NC	NC	NC
40	40	P3	PA1/$\overline{\text{ExIRQ1}}$/EVENT1/ A17	PA1/$\overline{\text{ExIRQ1}}$/EVENT1	FA17

43	43	N5	P87/$\overline{\text{ExIRQ15}}$/TxD3/ $\overline{\text{ADTRG}}$	P87/$\overline{\text{ExIRQ15}}$/TxD3/ $\overline{\text{ADTRG}}$	NC
44	44	P5	P86/$\overline{\text{ExIRQ14}}$/RxD3	P86/$\overline{\text{ExIRQ14}}$/RxD3	NC
45	45	R5	P85/$\overline{\text{ExIRQ13}}$/SCK1	P85/$\overline{\text{ExIRQ13}}$/SCK1	NC
46	46	M6	P84/$\overline{\text{ExIRQ12}}$/SCK3	P84/$\overline{\text{ExIRQ12}}$/SCK3	NC
47	47	N6	P83/SDA1	P83/SDA1	NC
48	48	R6	P82/SCL1	P82/SCL1	NC
49	49	P6	P81/SDA0	P81/SDA0	NC
50	50	M7	P80/SCL0	P80/SCL0	NC
—	—	N7	NC	NC	NC
51	51	R7	PE7/SERIRQ	PE7/SERIRQ	NC
—	—	P7	NC	NC	NC
52	52	M8	PE6/LCLK	PE6/LCLK	NC
53	53	N8	PE5/$\overline{\text{LRESET}}$	PE5/$\overline{\text{LRESET}}$	NC
54	54	R8	PE4/$\overline{\text{LFRAME}}$	PE4/$\overline{\text{LFRAME}}$	NC
55	55	P8	PE3/LAD3	PE3/LAD3	NC
56	56	M9	PE2/LAD2	PE2/LAD2	NC
57	57	N9	PE1/LAD1	PE1/LAD1	NC
58	58	R9	PE0/LAD0	PE0/LAD0	NC
—	—	P9	VCC	VCC	NC
59	59	M10	PD7/SDA5	PD7/SDA5	NC
60	60	N10	PD6/SCL5	PD6/SCL5	NC
61	61	R10	PD5/$\overline{\text{LPCPD}}$	PD5/$\overline{\text{LPCPD}}$	NC
62	62	P10	PD4/$\overline{\text{CLKRUN}}$	PD4/$\overline{\text{CLKRUN}}$	NC

68	68	N12	P70/AN0	P70/AN0	NC
69	69	R13	P71/AN1	P71/AN1	NC
70	70	M12	P72/AN2	P72/AN2	NC
71	71	P13	P73/AN3	P73/AN3	NC
72	72	R14	P74/AN4	P74/AN4	NC
73	73	P14	P75/AN5	P75/AN5	NC
74	74	R15	P76/AN6	P76/AN6	NC
75	75	N13	P77/AN7	P77/AN7	NC
76	76	P15	AVCC	AVCC	VCC
77	—	N14	AVref	AVref	VCC
78	77	M13	P60/$\overline{\text{IRQ14}}$/PWX0/D0	P60/$\overline{\text{IRQ14}}$/PWX0	NC
79	78	N15	P61/$\overline{\text{IRQ15}}$/PWX1/D1	P61/$\overline{\text{IRQ15}}$/PWX1	NC
80	79	M14	P62/PWX2/D2	P62/PWX2	NC
81	80	L12	P63/PWX3/D3	P63/PWX3	NC
82	81	M15	P64/$\overline{\text{ExIRQ11}}$/$\overline{\text{CTS}}$	P64/$\overline{\text{ExIRQ11}}$/$\overline{\text{CTS}}$	NC
83	82	L13	P65/$\overline{\text{ExIRQ10}}$/$\overline{\text{RTS}}$	P65/$\overline{\text{ExIRQ10}}$/$\overline{\text{RTS}}$	NC
84	83	L14	P66/$\overline{\text{ExIRQ9}}$/$\overline{\text{SCS}}$	P66/$\overline{\text{ExIRQ9}}$/$\overline{\text{SCS}}$	NC
85	84	L15	P67/$\overline{\text{ExIRQ8}}$/SSCK	P67/$\overline{\text{ExIRQ8}}$/SSCK	NC
86	85	K12	VCC	VCC	VCC
—	—	K13	DrVCC	DrVCC	VCC
—	—	K15	USD–	USD–	NC
—	—	K14	USD+	USD+	NC

89	88	H15	ETDI	ETDI	NC
90	89	H14	ETCK	ETCK	NC
91	90	G12	$\overline{\text{ETRST}}$	$\overline{\text{ETRST}}$	$\overline{\text{RES}}$
—	—	G13	PF2/RS10	PF2/RS10	NC
—	91	—	NC	NC	NC
92	92	G15	PF1/RS9/MDC	PF1/RS9/MDC	NC
93	93	G14	PF0/RS8/MDIO	PF0/RS8/MDIO	NC
—	—	F12	NC	NC	NC
94	94	F13	VSS	VSS	VSS
95	95	F15	P27/$\overline{\text{DTR}}$	P27/$\overline{\text{DTR}}$	NC
96	96	F14	P26/$\overline{\text{DSR}}$	P26/$\overline{\text{DSR}}$	NC
97	97	E13	P25/$\overline{\text{RI}}$	P25/$\overline{\text{RI}}$	NC
98	98	E15	P24/$\overline{\text{DCD}}$	P24/$\overline{\text{DCD}}$	NC
99	99	E14	P23/A11/AD11	P23	FA11
100	100	E12	P22/A10/AD10	P22	FA10
101	101	D15	P21/A9/AD9	P21	$\overline{\text{OE}}$
102	102	D14	P20/A8/AD8	P20	FA8
103	103	D13	P17/A7/AD7	P17	FA7
104	104	C15	P16/A6/AD6	P16	FA6
105	105	D12	P15/A5/AD5	P15	FA5
106	106	C14	P14/A4/AD4	P14	FA4
107	107	B15	P13/A3/AD3	P13	FA3
108	108	B14	P12/A2/AD2	P12	FA2

			RM_CRS-DV	RM_CRS-DV	
114	114	A13	PB5/EVENT13/ RM_REF-CLK	PB5/EVENT13/ RM_REF-CLK	NC
115	115	B12	PB4/EVENT12/ RM_TX-EN	PB4/EVENT12/ RM_TX-EN	NC
116	116	D11	PB3/EVENT11/DB3/ RM_RXD1	PB3/EVENT11/DB3/ RM_RXD1	NC
117	117	A12	PB2/EVENT10/DB2/ RM_RXD0	PB2/EVENT10/DB2/ RM_RXD0	NC
118	118	C11	PB1/EVENT9/DB1/ RM_TXD1	PB1/EVENT9/DB1/ RM_TXD1	NC
119	119	B11	PB0/EVENT8/DB0/ RM_TXD0	PB0/EVENT8/DB0/ RM_TXD0	NC
120	120	A11	VCC	VCC	VCC
121	121	D10	D8	P30/ExDB0	FO0
122	122	C10	D9	P31/ExDB1	FO1
123	123	A10	D10	P32/ExDB2	FO2
124	124	B10	D11	P33/ExDB3	FO3
125	125	D9	D12	P34/ExDB4	FO4
126	126	C9	D13	P35/ExDB5	FO5
127	127	A9	D14	P36/ExDB6	FO6
128	128	B9	D15	P37/ExDB7	FO7
129	129	D8	P40/$\overline{\text{IRQ0}}$/RS0/HC0/ D4	P40/$\overline{\text{IRQ0}}$/RS0/HC0	NC
130	130	C8	P41/$\overline{\text{IRQ1}}$/RS1/HC1/ D5	P41/$\overline{\text{IRQ1}}$/RS1/HC1	NC

135	133	A7	P52/IRQ10/TxD1	P52/IRQ10/TxD1	VCC
136	134	B7	P53/IRQ11/RxD1	P53/IRQ11/RxD1	VSS
137	135	D6	FWE	FWE	FWE
138	136	C6	P54/IRQ12/SSO	P54/IRQ12/SSO	NC
139	137	A6	P55/IRQ13/SSI	P55/IRQ13/SSI	NC
140	138	B6	P44/IRQ4/RS4/DB4/ HC4/A12/AD12	P44/IRQ4/RS4/DB4/ HC4	FA12
—	—	C5	VCC	VCC	VCC
—	—	A5	UXTAL	UXTAL	NC
—	—	B5	UEXTAL	UEXTAL	NC
—	—	D5	UXSEL	UXSEL	NC
—	—	A4	PF5/RS13	PF5/RS13	NC
—	—	B4	PF4/RS12	PF4/RS12	NC
—	139	C4	NC	NC	NC
—	140	—	NC	NC	NC
141	141	A3	VSS	VSS	VSS
—	—	D4	PF3/ExPWX3/RS11	PF3/ExPWX3/RS11	NC
142	142	B3	RESO	RESO	NC
143	143	A2	XTAL	XTAL	XTAL
144	144	B2	EXTAL	EXTAL	EXTA

	VCL	13	13	F3	Input	External capacitance pin f internal step-down power. this pin to Vss through an capacitor (that is located n pin) to stabilize internal st power.
	VSS	7, 42, 94, 110, 141	7, 42, 94, 110, 141	D2, M5, F13, C13, A3	Input	Ground pins. Connect all t pins to the system power s (0V).
Clock	XTAL	143	143	A2	Input	For connection to a crysta resonator. An external clo supplied from the EXTAL an example of crystal reso connection, see section 2 Pulse Generator.
	EXTAL	144	144	B2	Input	
	UXTAL	—	—	A5	Input	For connection to a crysta resonator for USB
	UEXTAL	—	—	B5	Input	
	UXSEL	—	—	D5	Input	USB clock source select p
	ϕ	5	5	D3	Output	Supplies the system clock external devices.
	EXCL	5	5	D3	Input	32.768-kHz external clock clock should be supplied.
Operating mode control	$\overline{MD2}$ MD1	14 9	14 9	F1 D1	Input	These pins set the operati Inputs at these pins shoul changed during operation
System control	$\overline{RES}$	8	8	E4	Input	Reset pin. When this pin i chip is reset.

				P3, R3		
	A15 to A0	4 to 2, 140, 99 to 109, 111	4 to 2, 138, 99 to 109, 111	C2, B1, C3, B6, E14, E12, D15, D14, D13, C15, D12, C14, B15, B14, A15, A14		
Data bus	D15 to D8	128 to 121	128 to 121	B9, A9, C9, D9, B10, A10, C10, D10	Input/ Output	Upper 8 bits of bidirectio
	D7 to D0	132 to 129, 81 to 78	132 to 129, 80 to 77	B8, A8, C8, D8, L12, M14, N15, M13		Lower 8 bits of bidirectio

				C14, B15, B14, A15, A14		
Interrupts	NMI	11	11	E2	Input	Nonmaskable interrupt rec input pin
	$\overline{\text{IRQ15}}$ to $\overline{\text{IRQ0}}$	79, 78, 139, 138, 136, 135, 15, 16, 4 to 2, 140, 132 to 129	78, 77, 137, 136, 134, 133, 15, 16, 4 to 2, 138, 132 to 129	N15, M13, A6, C6, B7, D6, F2, G4, C2, B1, C3, B6, B8, A8, C8, D8	Input	These pins are used to re maskable interrupts. Either $\overline{\text{IRQn}}$ or $\overline{\text{ExIRQn}}$ ca selected as the IRQn inter signal input pin.
	$\overline{\text{ExIRQ15}}$ to $\overline{\text{ExIRQ0}}$	43 to 46, 82 to 85, 33 to 35, 37 to 41	43 to 46, 81 to 84, 33 to 35, 37 to 41	N5, P5, R5, M6, M15, L13, L14, L15, M3, N1, M4, P1, P2, N3, P3, R3	Input	These pins are used to re maskable interrupts. Either $\overline{\text{IRQn}}$ or $\overline{\text{ExIRQn}}$ ca selected as the IRQn inter signal input pin.
Bus control	$\overline{\text{WAIT}}$	17	17	G3	Input	Requests wait state insert bus cycles when an extern state address space is ac

	$\overline{LWR}$	26	26	K4	Output	Low level on this pin indi the MCU is writing to an address space. The lower byte of the dat valid.
	$\overline{AS}/\overline{IOS}$	19	19	G2	Output	Low level on this pin indi the address output on th bus is valid.
	$\overline{CS256}$	17	17	G3	Output	Indicates access to the 2 area of H'F80000 to H'FF
Bus control	$\overline{WR}$	6	6	C1	Output	Low level on this pin indi the MCU is writing to an address space.
	$\overline{HBE}$	22	22	H2	Output	Low level on this pin indi the MCU is accessing an address space. The upper byte of the da valid.
	$\overline{LBE}$	24	24	J3	Output	Low level on this pin indi the MCU is accessing an address space. The lower byte of the dat valid.
	$\overline{AH}$	23	23	J4	Output	Address latch signal for t address-data multiplex b
Boundary scan	$\overline{ETRST}$	91	90	G12	Input	Boundary scan interface
	ETMS	87	86	H12	Input	
	ETDO	88	87	H13	Output	
	ETDI	89	88	H15	Input	
	ETCK	90	89	H14	Input	

communi-cation interface (SCI_1 and SCI_3)	TxD3					
	RxD1, RxD3	136, 44	134, 44	B7, P5	Input	Receive data input pins
	SCK1, SCK3	45, 46	45, 46	R5, M6	Input/ Output	Clock input/output pins.
Serial communi-cation interface with FIFO (SCIF)	TxDF	16	16	G4	Output	Transmit data output pin
	RxDF	15	15	F2	Input	Receive data input pin
	$\overline{\text{CTS}}$	82	81	M15	Input	Transmit grant input pin
	$\overline{\text{RTS}}$	83	82	L13	Output	Transmit request output p
	$\overline{\text{DTR}}$	95	95	F15	Output	Data terminal ready outpu
	$\overline{\text{DSR}}$	96	96	F14	Input	Data set ready input pin
	$\overline{\text{RI}}$	97	97	E13	Input	Ring indicator input pin
	$\overline{\text{DCD}}$	98	98	E15	Input	Data carrier detection inpu
Synchronous serial communi-cation unit (SSU)	SSCK	85	84	L15	Input/ Output	SSU clock I/O pin
	SSI	139	137	A6	Input/ Output	SSU data I/O pin
	SSO	138	136	C6	Input/ Output	SSU data I/O pin
	$\overline{\text{SCS}}$	84	83	L14	Input/ Output	SSU chip select I/O pin
I^2C bus interface (IIC)	SCL0 to SCL5	50, 48, 32, 30, 28, 60	50, 48, 32, 30, 28, 60	M7, R6, M2, L1, K1, N10	Input/ Output	IIC clock input/output pins pins can drive a bus direc the NMOS open drain out

				M12, R13, N12		
	AVCC	76	76	P15	Input	Analog power supply pin the A/D converter is not u these pins should be con the system power supply
	AVref	77	—	N14	Input	Analog reference voltage When the A/D converter used, this pin should be c to the system power sup V).
	AVSS	67	67	P12	Input	Analog ground pins. The should be connected to t power supply (0 V).
	$\overline{\text{ADTRG}}$	43	43	N5	Input	External trigger input pin A/D conversion
LPC Interface (LPC)	LAD3 to LAD0	55 to 58	55 to 58	P8, M9, N9, R9	Input/ Output	Transfer cycle type/addre I/O pins
	$\overline{\text{LFRAME}}$	54	54	R8	Input	Input pin indicating trans start and forced terminat
	$\overline{\text{LRESET}}$	53	53	N8	Input	LPC reset pin. When this a reset state is entered.
	LCLK	52	52	M8	Input	PCI clock input pin
	SERIRQ	51	51	R7	Input/ Output	LPC serialized host inter request signal
	LSCI, $\overline{\text{LSMI}}$, $\overline{\text{PME}}$	66 65 64	66 65 64	M11 P11 R11	Input/ Output	LPC auxiliary output. The functions are general I/O

Ethernet controller (EtherC)	RM_REF-CLK	114	114	A13	Input	Transmit/receive Clock
	RM_TX-EN	115	115	B12	Output	Transmit enable
	RM_TXD1 RM_TXD0	118 119	118 119	C11 B11	Output Output	Transmit data
	RM_CRS-DV	113	113	C12	Input	Carrier detection/receive d
	RM_RXD1 RM_RXD0	116 117	116 117	D11 A12	Input Input	Receive data
	RM_RX-ER	112	112	B13	Input	Receive error
	MDC	92	92	G15	Input	Management data clock
	MDIO	93	93	G14	Input/ Output	Management data I/O
	LNKSTA	34	34	N1	Input	Link status
	EXOUT	33	33	M3	Output	General-purpose external
	WOL	35	35	M4	Output	Wake-on-LAN
USB function module (USB)	VBUS	—	—	J14	Input	USB cable connection mo
	USD+	—	—	K14	Input/ Output	USB data I/O pin
	USD–	—	—	K15	Input/ Output	USB data I/O pin
	DrVcc	—	—	K13	Input	Power supply pin for USB transceiver
	DrVss	—	—	J13	Input	Ground pin for USB built-i transceiver
	PUPDPLS	—	—	J15	Output	USB+ pull-up control pin

				M4, P1, P2, N3, P3, R3		
Retain state output pins	RS14	10	10	E3	Output	Retain state output pins. The outputs on these pin initialized by a system re Pins RS13 to RS10 are s only by the H8S/2472 Gr
	RS13 to RS10	—	—	A4, B4, D4, G13		
	RS9 to RS0	92, 93, 4 to 2, 140, 132 to 129	92,93,4 to 2, 138, 132 to 129	G15, G14, C2, B1, C3, B6, B8, A8, C8, D8		
Debounced input pins	DB7 to DB0	4 to 2, 140, 116 to 119	4 to 2, 138, 116 to 119	C2, B1, C3, B6, D11, A12, C11, B11	Input	Pins with noise eliminatir functions.
	ExDB7 to ExDB0	128 to 121	128 to 121	B9, A9, C9, D9, B10, A10, C10, D10		

			B13, B14, A15		
P27 to P20	95 to 102	95 to 102	F15, F14, E13, E15, E14, E12, D15, D14	Input/ Output	8-bit input/output pins
P37 to P30	128 to 121	128 to 121	B9, A9, C9, D9, B10, A10, C10, D10	Input/ Output	8-bit input/output pins
P47 to P40	4 to 2, 140, 132 to 129	4 to 2, 138, 132 to 129	C2, B1, C3, B6, B8, A8, C8, D8	Input/ Output	8-bit input/output pins
P57 to P50	6, 5, 139, 138, 136, 135, 15, 16	6, 5, 137, 136, 134, 133, 15, 16	C1, D3, A6, C6, B7, A7, F2, G4	Input/ Output	8-bit input/output pins

P77 to P70	75 to 68	75 to 68	R15, P14, R14, P13, M12, R13, N12	Input	8-bit input pins
P87 to P80	43 to 50	43 to 50	N5, P5, R5, M6, N6, R6, P6, M7	Input/ Output	8-bit input/output pins
P97 to P90	17 to 24	17 to 24	G3, G1, G2, H4, H3, H2, J4, J3	Input/ Output	8-bit input/output pins
PA7 to PA0	33 to 35, 37 to 41	33 to 35, 37 to 41	M3, N1, M4, P1, P2, N3, P3, R3	Input/ Output	8-bit input/output pins
PB7 to PB0	112 to 119	112 to 119	B13, C12, A13, B12, D11, A12, C11, B11	Input/ Output	8-bit input/output pins
PC7 to PC0	25 to 32	25 to 32	J2, K4, K3, K1, K2, L1, L4, M2	Input/ Output	8-bit input/output pins

PE7 to PE0	51 to 58	51 to 58	H7, M8, N8, R8, P8, M9, N9, R9	Input/ Output	8-bit input/output pins
PF6	10	10	E3	Input/ Output	7-bit input/output pins. Pins PF5 to PF2 are supp only by the H8S/2472 Gro
PF5 to PF2	—	—	A4, B4, D4, G13		
PF1, PF0	92, 93	92, 93	G15, G14		

- Upward-compatible with H8/300 and H8/300H CPUs
  - Can execute H8/300 and H8/300H CPUs object programs
- General-register architecture
  - Sixteen 16-bit general registers also usable as sixteen 8-bit registers or eight 32-b
- Sixty-nine basic instructions
  - 8/16/32-bit arithmetic and logic instructions
  - Multiply and divide instructions
  - Powerful bit-manipulation instructions
  - Multiply-and-accumulate instruction
- 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: 2 states

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

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 the H 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	2*	12
	MULXU.W Rs, ERd	2*	20
MULXS	MULXS.B Rs, Rd	3*	13
	MULXS.W Rs, ERd	3*	21
CLRMAC	CLRMAC	1*	Not supported
LDMAC	LDMAC ERs,MACH	1*	
	LDMAC ERs,MACL	1*	
STMAC	STMAC MACH,ERd	1*	
	STMAC MACl,ERd	1*	

Note: * This becomes one state greater immediately after a MAC instruction.
In addition, there are differences in address space, CCR and EXR register fun and power-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.
  - A multiply-and-accumulate instruction has 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/2600 CPU has the following enhancement

- More control registers
  - One 8-bit and two 32-bit control registers have been added.
- Enhanced instructions
  - Addressing modes of bit-manipulation instructions have been enhanced.
  - A multiply-and-accumulate instruction has 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.

Linear access to a 64-kbyte maximum address space is provided.

- 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. The exception vector table structure in normal mo
  shown in figure 2.1. For details of the exception vector table, see section 4, Exception
  Handling.

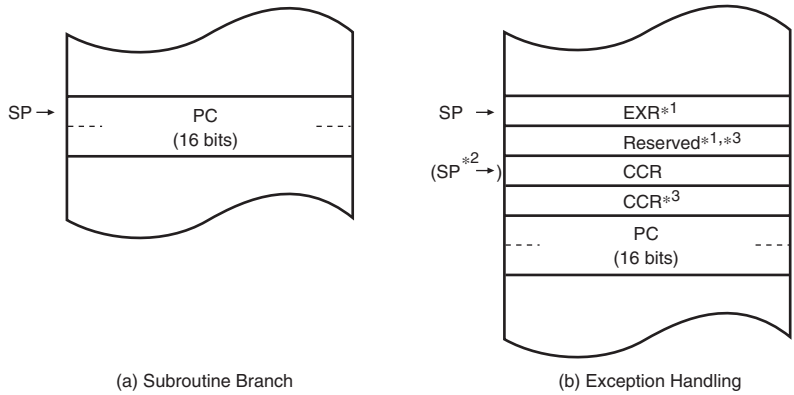
  The memory indirect addressing mode (@@aa:8) employed in the JMP and JSR instr
  uses an 8-bit absolute address included in the instruction code to specify a memory op
  that contains a branch address. In normal mode the operand is a 16-bit word operand,
  providing a 16-bit branch address. Branch addresses can be stored in the area from H'
  H'00FF. Note that the first part of this range 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 t
  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.

Note: Normal mode is not available in this LSI.

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

Exception Vector Table and Memory Indirect Branch Addresses

In advanced mode, the top area starting at H'00000000 is allocated to the exception ve
table in units of 32 bits. In each 32 bits, the upper 8 bits are ignored and a branch addr
stored in the lower 24 bits (figure 2.3). For details of the exception vector table, see se
Exception Handling.

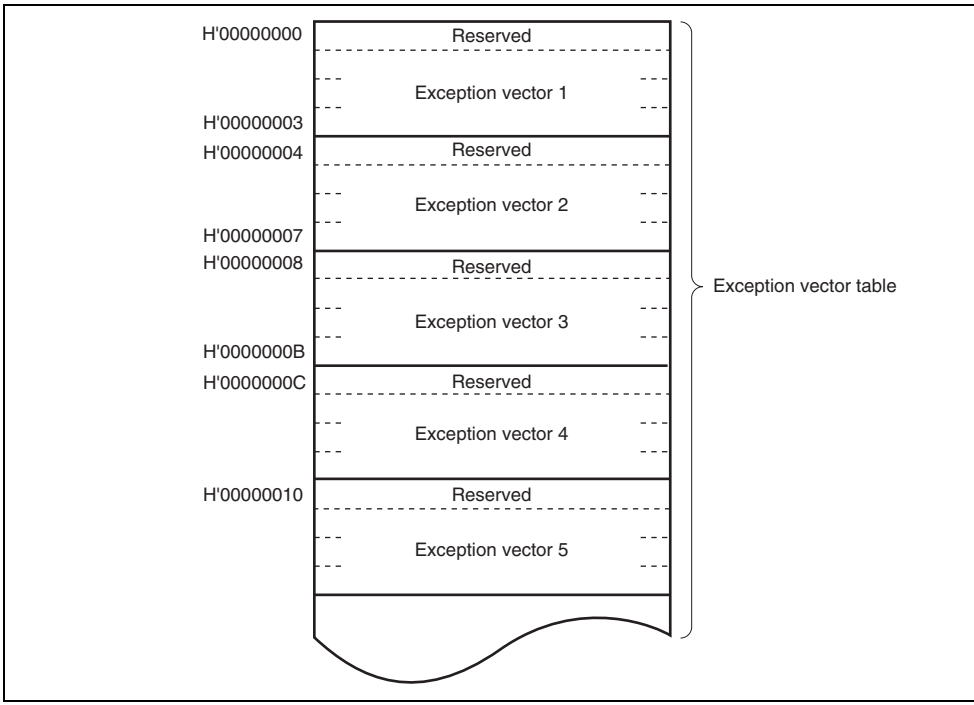


Figure 2.3 Exception Vector Table (Advanced Mode)

Exception Handling.

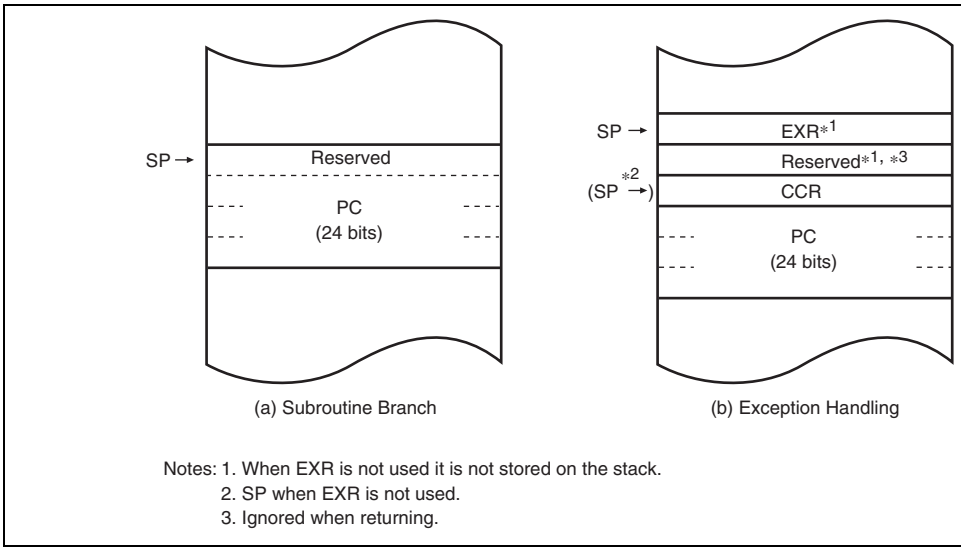


Figure 2.4 Stack Structure in Advanced Mode

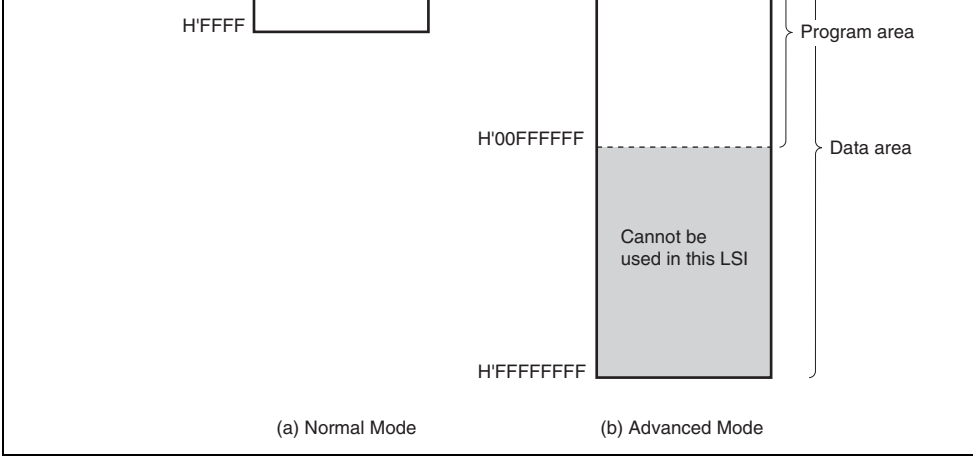
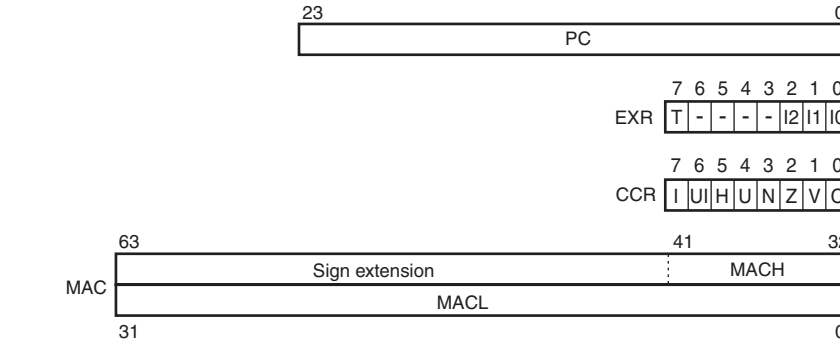


Figure 2.5 Memory Map

ER2	E2	R2H	R2L
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	MAC:	Multiply-accumulate register
UI:	User bit or interrupt mask bit		

Figure 2.6 CPU 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 of s
bit registers.

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.

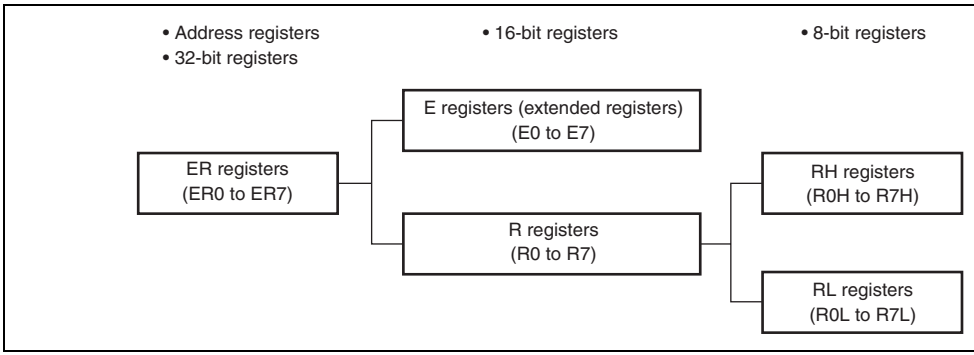


Figure 2.7 Usage of General Registers

Figure 2.8 Stack

2.4.2 Program Counter (PC)

This 24-bit counter indicates the address of the next instruction the CPU will execute. T
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, are executed, all interrupts inclu
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 exception is g each time an instruction is executed. When th cleared to 0, instructions are executed in sequ
6 to 3	—	All 1	—	Reserved These bits are always read as 1.
2	I2	1	R/W	These bits designate the interrupt mask level For details, refer to section 5, Interrupt Contro
1	I1	1	R/W	
0	I0	1	R/W	

7	I	1	R/W	Interrupt Mask Bit Masks interrupts other than NMI when set to 1. accepted regardless of the I bit setting. The I b 1 at the start of an exception-handling sequenc details, refer to section 5, Interrupt Controller.
6	UI	Undefined	R/W	User Bit or Interrupt Mask Bit Can be read or written by software using the LI ANDC, ORC, and XORC instructions. This bit c used as an interrupt mask bit in this LSI.
5	H	Undefined	R/W	Half-Carry Flag When the ADD.B, ADDX.B, SUB.B, SUBX.B, C or NEG.B instruction is executed, this flag is se there is a carry or borrow at bit 3, and cleared t otherwise. When the ADD.W, SUB.W, CMP.W NEG.W instruction is executed, the H flag is se there is a carry or borrow at bit 11, and clearec otherwise. When the ADD.L, SUB.L, CMP.L, o instruction is executed, the H flag is set to 1 if t carry or borrow at bit 27, and cleared to 0 othe
4	U	Undefined	R/W	User Bit Can be read or written by software using the LI ANDC, ORC, and XORC instructions.
3	N	Undefined	R/W	Negative Flag Stores the value of the most significant bit of da sign bit.
2	Z	Undefined	R/W	Zero Flag Set to 1 to indicate zero data, and cleared to 0 indicate non-zero data.

- Shift and rotate instructions, to indicate a

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

2.4.5 Multiply-Accumulate Register (MAC)

This 64-bit register stores the results of multiply-and-accumulate operations. It consists
bit registers denoted MACH and MACL. The lower 10 bits of MACH are valid; the upp
a sign extension.

2.4.6 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.

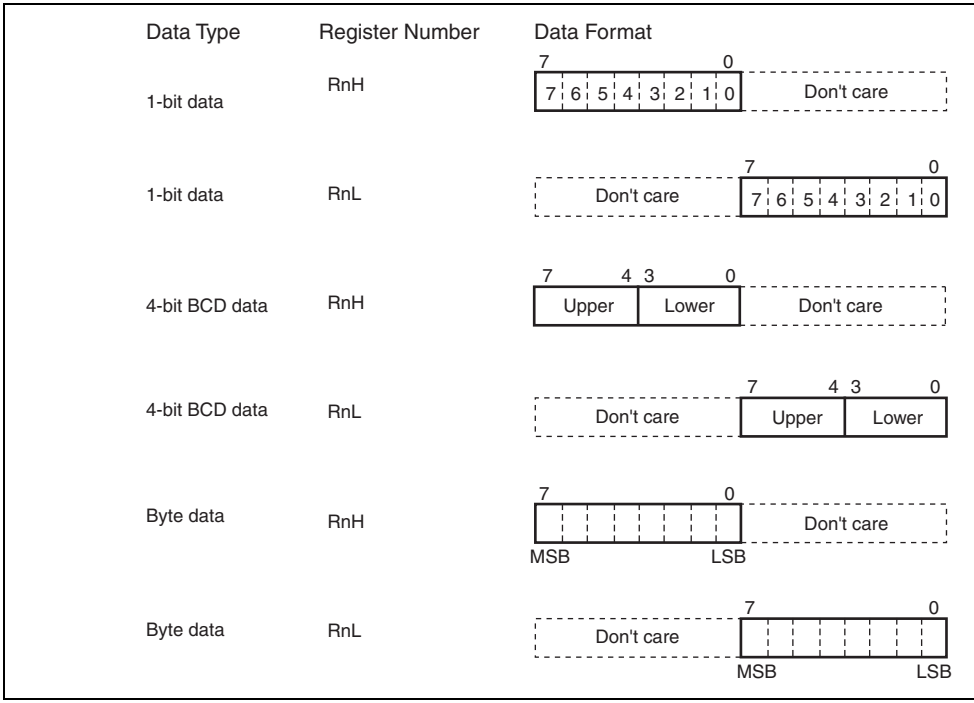
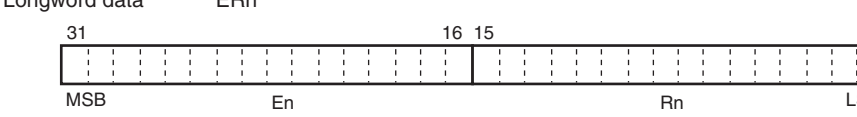


Figure 2.9 General Register Data Formats (1)



[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)

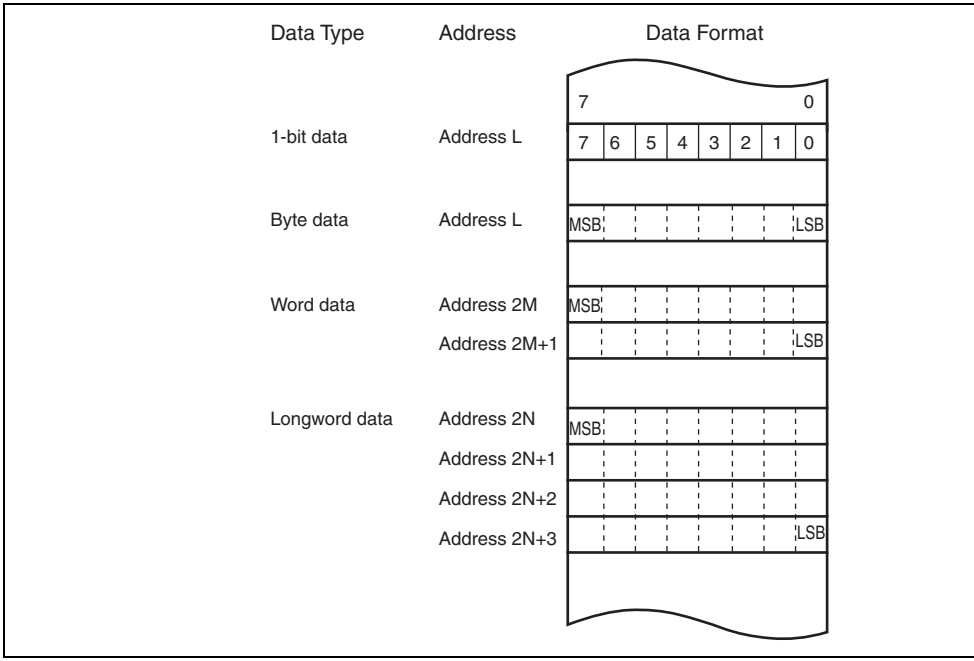


Figure 2.10 Memory Data Formats

Arithmetic operation	ADD, SUB, CMP, NEG	B/W
	ADDX, SUBX, DAA, DAS	B
	INC, DEC	B/W
	ADDS, SUBS	L
	MULXU, DIVXU, MULXS, DIVXS	B/W
	EXTU, EXTS	W/L
	TAS*[4]	B
	MAC, LDMAC, STMAC, CLRMAC	—
Logic operations	AND, OR, XOR, NOT	B/W
Shift	SHAL, SHAR, SHLL, SHLR, ROTL, ROTR, ROTXL, ROTXR	B/W
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	—

Notes: B-byte; W-word; L-longword.

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

ERn	General register (32-bit register)
MAC	Multiply-accumulate 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
→	Move
~	NOT (logical complement)
:8/:16/:24/:32	8-, 16-, 24-, or 32-bit length

Note: * General registers include 8-bit registers (R0H to R7H, R0L to R7L), 16-bit regis
to R7, E0 to E7), and 32-bit registers (ER0 to ER7).

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	L	@SP+ → Rn (register list) Pops two or more general registers from the stack.
STM	L	Rn (register list) → @–SP Pushes two or more general registers onto the stack.

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

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
MAC	—	(EAs) × (EAd) + MAC → MAC Performs signed multiplication on memory contents and adds to the multiply-accumulate register. The following operations c performed: 16 bits × 16 bits + 32 bits → 32 bits, saturating 16 bits × 16 bits + 42 bits → 42 bits, non-saturating
CLRMAC	—	0 → MAC Clears the multiply-accumulate register to zero.
LDMAC STMAC	L	Rs → MAC, MAC → Rd Transfers data between a general register and a multiply-accu register.

Note: 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

		another general register or immediate data.
NOT	B/W/L	~(Rd) → (Rd) Takes the one's complement (logical complement) of general re 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

		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 val
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.

Some instructions have two operation fields.

- Register Field

  Specifies a general register. Address registers are specified by 3 bits, and data register 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.

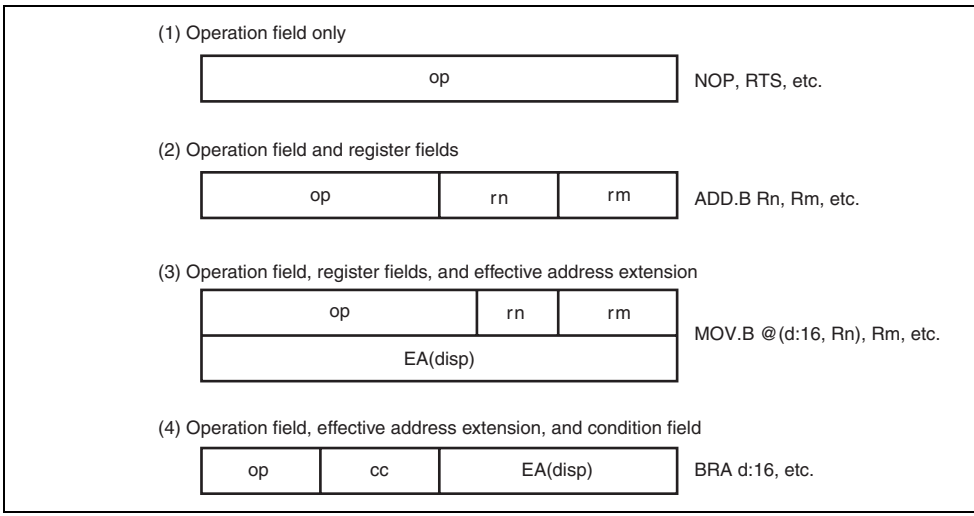


Figure 2.11 Instruction Formats (Examples)

No.	Addressing Mode	Symbol
1	Register direct	Rn
2	Register indirect	@ERn
3	Register indirect with displacement	@(d:16,ERn)/@(d:32,ERn)
4	Register indirect with post-increment Register indirect with pre-decrement	@ERn+ @–ERn
5	Absolute address	@aa:8/@aa:16/@aa:24/@a
6	Immediate	#xx:8/#xx:16/#xx:32
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 lo bits are valid and the upper 8 bits are all assumed to be 0 (H'00).

address register. The value added is 1 for byte access, 2 for word transfer instruction, or 4
longword transfer instruction. For the word or longword transfer instructions, the register
should be even.

Register indirect with pre-decrement—@-ERn: The value 1, 2, or 4 is subtracted from
address register (ERn) specified by the register field in the instruction code, and the resul
address of a memory operand. The result is also stored in the address register. The value
subtracted is 1 for byte access, 2 for word transfer instruction, or 4 for longword transfer
instruction. For the word or longword transfer instructions, the register value should be ev

2.7.5 Absolute Address—@aa:8, @aa:16, @aa:24, or @aa:32

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).

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 dat
operand.

The ADDS, SUBS, INC, and DEC instructions contain immediate data implicitly. Some
manipulation instructions contain 3-bit immediate data in the instruction code, specifyin
number. The TRAPA instruction contains 2-bit immediate data in its instruction code, sp
vector address.

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

This mode is used in the Bcc and BSR instructions. An 8-bit or 16-bit displacement cont
the instruction is sign-extended and added to the 24-bit PC contents to generate a branch
Only the lower 24 bits of this branch address are valid; the upper 8 bits are all assumed t
(H′00). The PC value to which the displacement is added is the address of the first byte
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.

If an odd address is specified in word or longword memory access, or as a branch address 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.

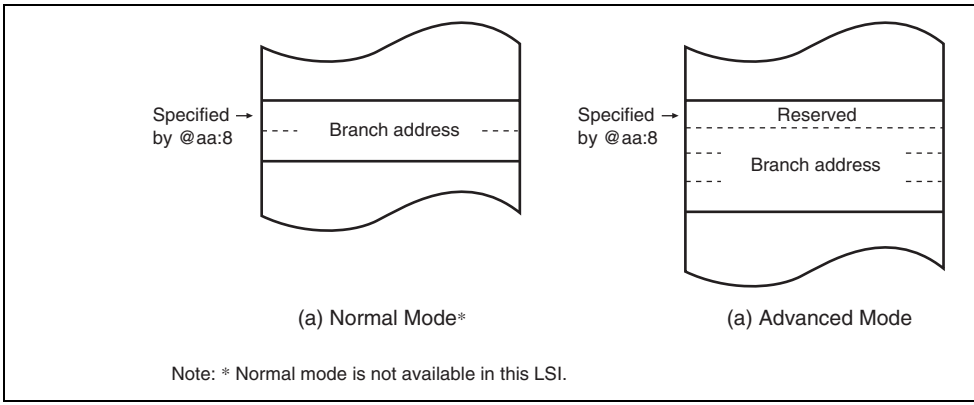
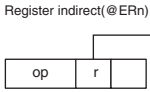
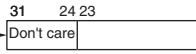
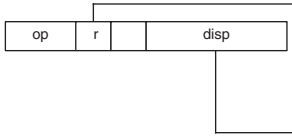
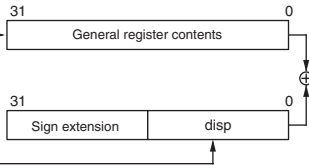
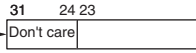
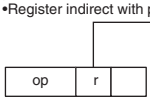
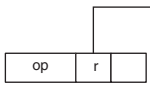
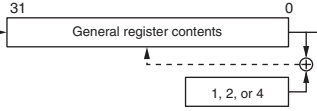
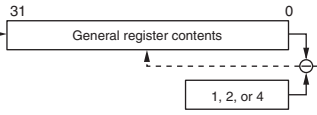
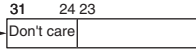


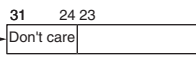
Figure 2.12 Branch Address Specification in Memory Indirect Mode

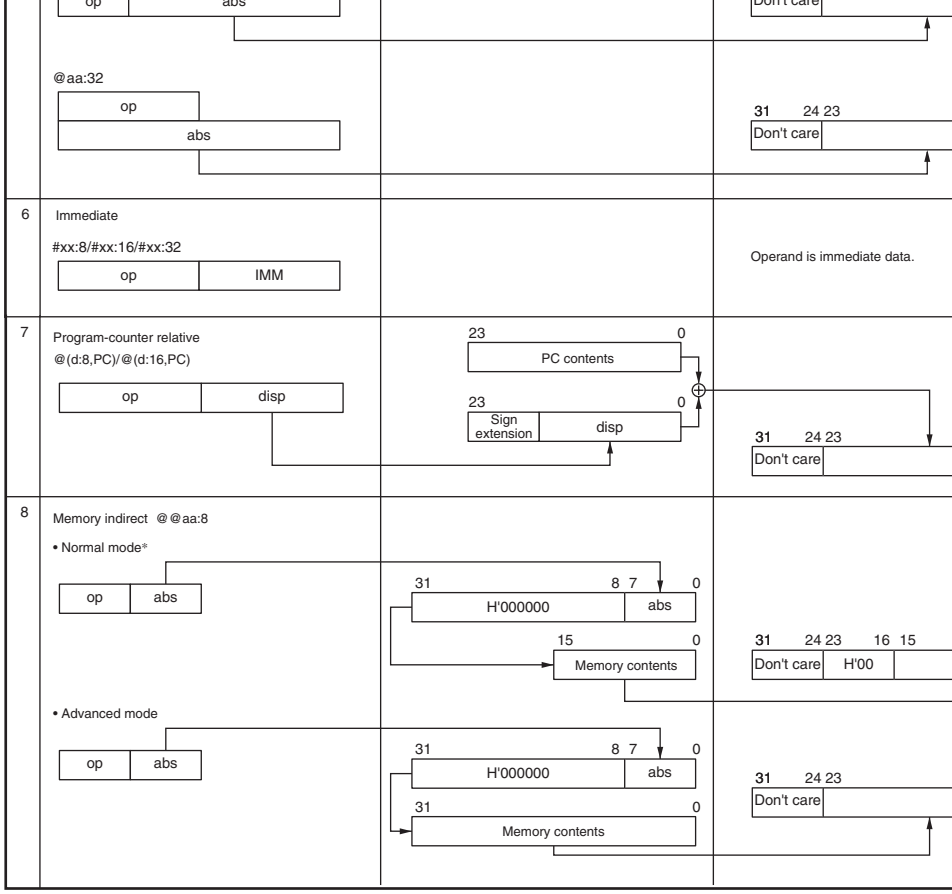
<table>
<tr><td>2</td><td>Register indirect(@ERn)
 
</td><td> 
</td><td> 
</td></tr>
<tr><td>3</td><td>Register indirect with displacement
@(d:16,ERn) or @(d:32,ERn)
 
</td><td> 
</td><td> 
</td></tr>
<tr><td>4</td><td>Register indirect with post-increment or
pre-decrement
•Register indirect with post-increment @ERn+
 

•Register indirect with pre-decrement @-ERn
 

<table><tr><th>Operand Size</th><th>Offset</th></tr><tr><td>Byte</td><td>1</td></tr><tr><td>Word</td><td>2</td></tr><tr><td>Longword</td><td>4</td></tr></table></td><td> 

 
</td><td> 

 
</td></tr>
</table>



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

- Exception-Handling State

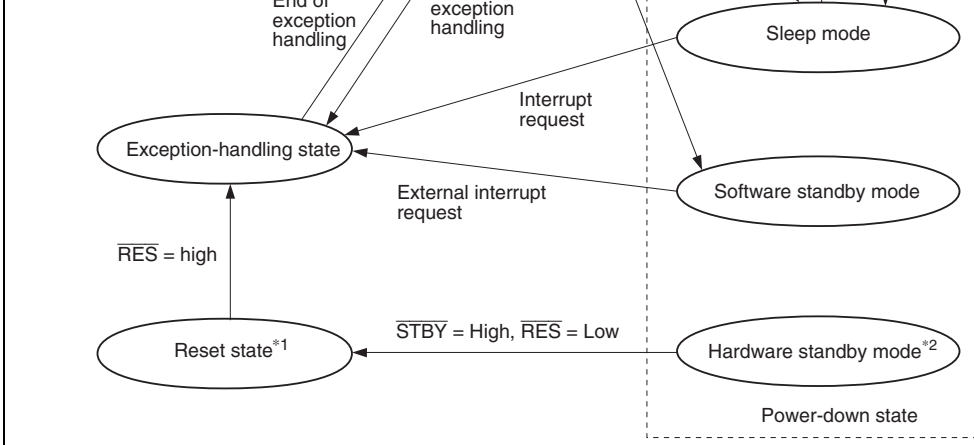
  The exception-handling state is a transient state that occurs when the CPU alters the processing flow due to an exception source, such as a reset, trace, interrupt, or trap i The CPU fetches a start address (vector) from the exception vector table and branche address. For further details, refer to section 4, Exception Handling.

- Program Execution State

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

- Program Stop 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 software standby mode. Fo details, refer to section 28, Power-Down Modes.



Notes: 1. From any state except hardware standby mode, a transition to the reset state occurs whenever 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.

Figure 2.13 State Transitions

beforehand.

2	1	1	Advanced	Extended mode with on-chip ROM Single-chip mode

Mode 2 is single-chip mode after a reset. The CPU can switch to extended mode by setti
EXPE in MDCR to 1.

Modes 0, 1, 3, 5, and 7 are not available in this LSI. Modes 4 and 6 are operating mode
special purpose. Thus, mode pins should be set to enable mode 2 in normal program exe
state. Mode pins should not be changed during operation.

3.2.1 Mode Control Register (MDCR)

MDCR is used to set an operating mode and to monitor the current operating mode.

Bit	Bit Name	Initial Value	R/W	Description
7	EXPE	0	R/W	Extended Mode Enable Specifies extended mode. 0: Single-chip mode 1: Extended mode
6 to 3	—	All 0	R	Reserved
2	MDS2	—*	R	Mode Select 2 and 1
1	MDS1	—*	R	These bits indicate the input levels at mode pins and MD1) (the current operating mode). Bits MD MDS1 correspond to $\overline{MD2}$ and MD1, respectively and MDS1 are read-only bits and they cannot be to. The mode pin ($\overline{MD2}$ and MD1) input levels are into these bits when MDCR is read. These latche canceled by a reset.
0	—	—	R	Reserved

Note: * The initial values are determined by the settings of the $\overline{MD2}$ and MD1 pins.

				0: P97/$\overline{\text{WAIT}}$ pin $\overline{\text{WAIT}}$ pin function is selected by the settings WSCR and WSCR2. 1: $\overline{\text{CS256}}$ pin Outputs low when a 256-kbyte expansion are addresses H'F80000 to H'FBFFFF is accesse
6	IOSE	0	R/W	IOS Enable Enables or disables $\overline{\text{AS}}$/$\overline{\text{IOS}}$ pin function in exte mode. 0: $\overline{\text{AS}}$ pin Outputs low when an external area is access 1: $\overline{\text{IOS}}$ pin Outputs low when an IOS expansion area of addresses H'FFF000 to H'FFF7FF is accesse
5	INTM1	0	R	These bits select the control mode of the interru
4	INTM0	0	R/W	controller. For details on the interrupt control mo section 5.6, Interrupt Control Modes and Interru Operation. 00: Interrupt control mode 0 01: Interrupt control mode 1 10: Setting prohibited 11: Setting prohibited

				Selects the valid edge of the NMI interrupt input. 0: An interrupt is requested at the falling edge of I input 1: An interrupt is requested at the rising edge of N input
1	—	0	R/W	Reserved The initial value should not be changed.
0	RAME	1	R/W	RAM Enable Enables or disables on-chip RAM. The RAME bit initialized when the reset state is released. 0: On-chip RAM is disabled 1: On-chip RAM is enabled

3.2.3 Serial Timer Control Register (STCR)

STCR enables or disables register access, IIC operating mode, and on-chip flash memory selects the input clock of the timer counter.

Bit	Bit Name	Initial Value	R/W	Description
7	IICX2	0	R/W	IIC Transfer Rate Select 2, 1, and 0
6	IICX1	0	R/W	These bits control the IIC operation. These bits s transfer rate in master mode together with bits CI CKS0 in the I^2C bus mode register (ICMR). For d the transfer rate, see table 18.3.
5	IICX0	0	R/W	

				0: Area from H'FFFE88 to H'FFFE8F is reserved Area from H'FFFEA0 to H'FFFEBF is allocate registers of AD, serial multiplexed functions, a ports. Area from H'FFFF80 to H'FFFF87 is allocated registers of power-down states and on-chip p modules. 1: Area from H'FFFE88 to H'FFFE8F is allocate control registers of flash memory. Area from H'FFFEA0 to H'FFFEBF is reserve Area from H'FFFF80 to H'FFFF87 is reserved
2	—	1	R/(W)	Reserved The initial value should not be changed.
1	ICKS1	0	R/W	Internal Clock Source Select 1, 0
0	ICKS0	0	R/W	These bits select a clock to be input to the timer (TCNT) and a count condition together with bits CKS0 in the timer control register (TCR). For de section 11.2.4, Timer Control Register (TCR).

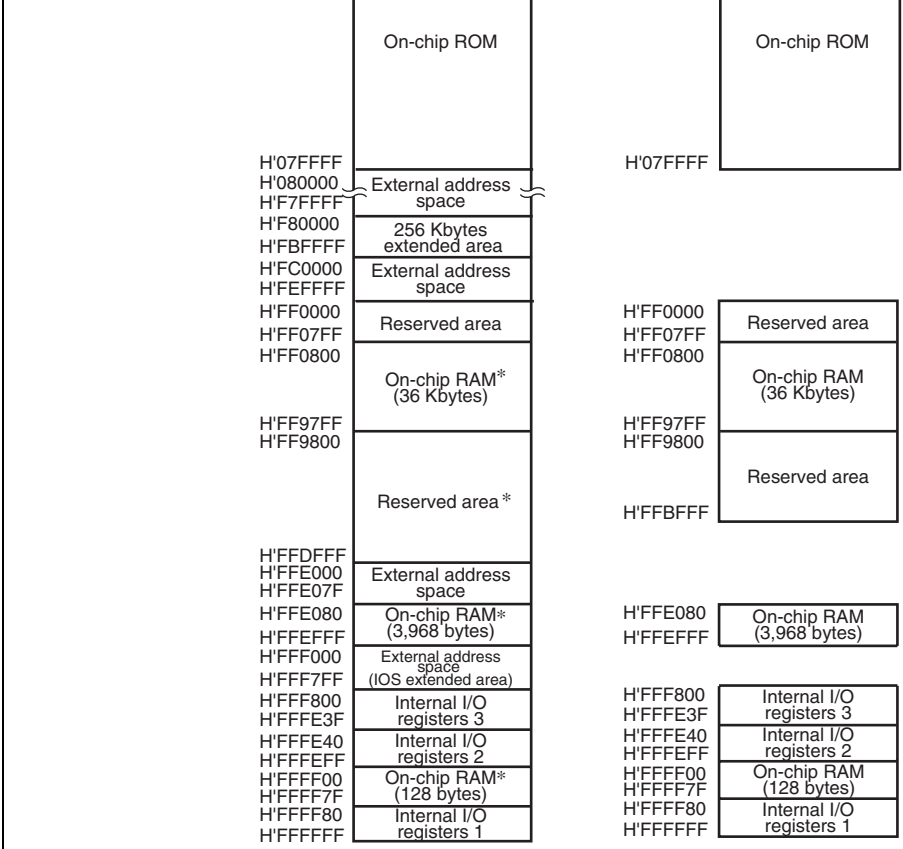
In extended mode, ports 1, 2 (P23 to P20), and 4 (P47 to P44) function as input ports
reset.

Ports 1 and 2 function as an address bus by setting 1 to the corresponding port data di
register (DDR). Port 3 functions as a data bus port, and parts of port 9 and port C carr
control signals. Ports 4 (P43 to P40) and 6 (P63 to P60) function as a data bus port wh
ABW bit in WSCR is cleared to 0.

- Multiplex extended mode

  When 8-bit bus is specified, port 1 functions as the port for address output and data input/output regardless of the setting of the data direction register (DDR). Ports 2 (P2
  and 4 (P47 to P44) can be used as a general port.

  When 16-bit bus is specified, ports 1, 2 (P23 to P20), and 4 (P47 to P44) function as t
  for address output and data input/output regardless of the setting of the data direction
  (DDR).



Notes: * These areas can be used as an external address space by clearing bit RAME in SYSCR to 0.

Figure 3.1 Address Map

Priority	Exception Type	Start of Exception Handling
High ↑	Reset	Starts immediately after a low-to-high transition of pin, or when the watchdog timer overflows.
	Illegal instruction	Started by execution of an undefined code.
	Interrupt	Starts when execution of the current instruction or handling ends, if an interrupt request has been iss Interrupt detection is not performed on completion ORC, XORC, or LDC instruction execution, or on completion of reset exception handling.
	Trap instruction	Started by execution of a trap (TRAPA) instruction instruction exception handling requests are accep times in program execution state.
Low		

Reserved for system use		1 ⏐ 3	H'000004 to H'000007 ⏐ H'00000C to H'00000F
Illegal instruction		4	H'000010 to H'000013
Reserved for system use		5	H'000014 to H'000017
		6	H'000018 to H'00001B
External interrupt (NMI)		7	H'00001C to H'00001F
Trap instruction (four sources)		8	H'000020 to H'000023
		9	H'000024 to H'000027
		10	H'000028 to H'00002B
		11	H'00002C to H'00002F
Reserved for system use		12 ⏐ 15	H'000030 to H'000033 ⏐ H'00003C to H'00003F
External interrupt	IRQ0	16	H'000040 to H'000043
	IRQ1	17	H'000044 to H'000047
	IRQ2	18	H'000048 to H'00004B
	IRQ3	19	H'00004C to H'00004F
	IRQ4	20	H'000050 to H'000053
	IRQ5	21	H'000054 to H'000057
	IRQ6	22	H'000058 to H'00005B
	IRQ7	23	H'00005C to H'00005F
Internal interrupt*		24 ⏐ 29	H'000060 to H'000063 ⏐ H'000074 to H'000077

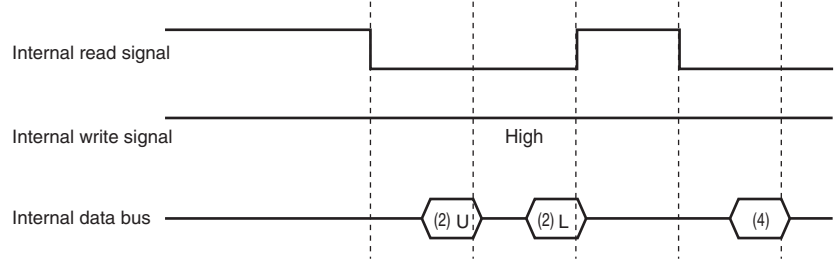
	IRQ10	58	H'0000E8 to H'0000EB
	IRQ11	59	H'0000EC to H'0000EF
	IRQ12	60	H'0000F0 to H'0000F3
	IRQ13	61	H'0000F4 to H'0000F7
	IRQ14	62	H'0000F8 to H'0000FB
	IRQ15	63	H'0000FC to H'0000FF
Internal interrupt*		64 | 119	H'000100 to H'000103 | H'0001DC to H'0001DF

Note: * For details on the internal interrupt vector table, see section 5.5, Interrupt Exc Handling Vector Table.

When the $\overline{\text{RES}}$ pin goes high after being held low for the necessary time, this LSI starts re
exception handling as follows:

1. The internal state of the CPU and the registers of the on-chip peripheral modules are i
   and the I bit in CCR is set 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.

Figure 4.1 shows an example of the reset sequence.



(1) Reset exception handling vector address (1) U = H'000000 (1) L = H'000002
(2) Start address (contents of reset exception handling vector address)
(3) Start address ((3) = (2)U + (2)L)
(4) First program instruction

Figure 4.1 Reset Sequence

4.3.2 Interrupts after Reset

If an interrupt is accepted after a reset and before the stack pointer (SP) is initialized, the CCR will not be saved correctly, leading to a program crash. To prevent this, all interrup including NMI, are disabled immediately after a reset. Since the first instruction of a pro always executed immediately after the reset state ends, make sure that this instruction in the stack pointer (example: MOV.L #xx: 32, SP).

4.3.3 On-Chip Peripheral Modules after Reset is Cancelled

After a reset is cancelled, the module stop control registers (MSTPCR, MSTPCRA, and SUBMSTPB) are initialized, and all modules except the DTC operate in module stop m Therefore, the registers of on-chip peripheral modules cannot be read from or written to from and write to these registers, clear module stop mode.

2. A vector address corresponding to the interrupt source is generated, the start address i
from the vector table to the PC, and program execution begins from that address.

4.5 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) and condition code register (CCR) are saved t
stack.
2. A vector address corresponding to the interrupt source is generated, the start address i
from the vector table to the PC, and program execution starts from that address.

The TRAPA instruction fetches a start address from a vector table entry corresponding to
number from 0 to 3, as specified in the instruction code.

Table 4.3 shows the status of CCR after execution of trap instruction exception handling.

Table 4.3 Status of CCR after Trap Instruction Exception Handling

	CCR	
Interrupt Control Mode	**I**	**UI**
0	Set to 1	Retains value pri
execution
1	Set to 1	Set to 1

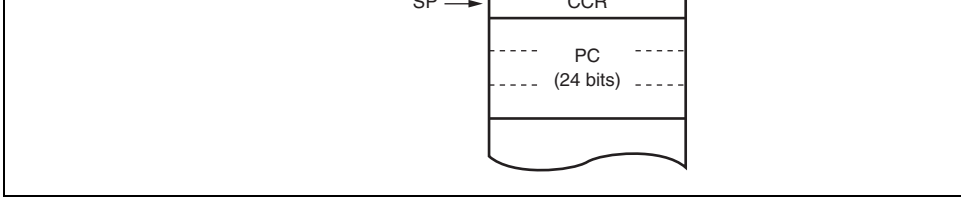


Figure 4.2 Stack Status after Exception Handling

Use the following instructions to restore registers:

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

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

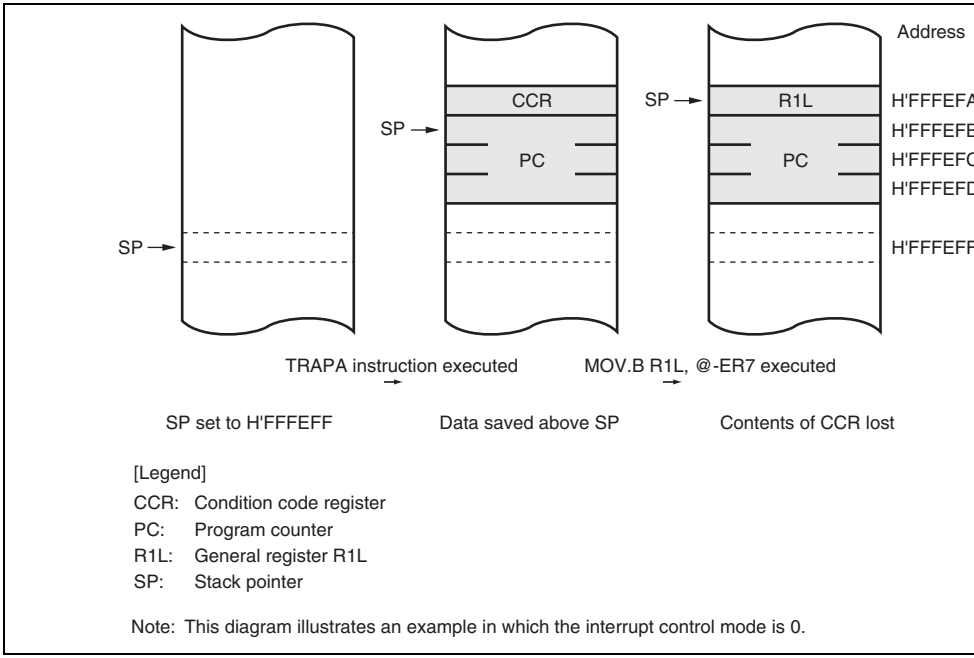


Figure 4.3 Operation When SP Value is Odd

can be set for each module for all interrupts except NMI.

- Three-level interrupt mask control

  By means of the interrupt control mode, I and UI bits in CCR, and ICR, 3-level inter control is performed.

- Independent vector addresses

  All interrupt sources are assigned independent vector addresses, making it unnecessa source to be identified in the interrupt handling routine.

- Thirty-three external interrupts

  NMI is the highest-priority interrupt, and is accepted at all times. Rising edge or fall detection can be selected for NMI. Falling-edge, rising-edge, or both-edge detection, sensing, can be selected for $\overline{\text{IRQn}}$ (n = 15 to 0) and $\overline{\text{ExIRQn}}$ (n = 15 to 0).

- DTC control

  The DTC can be activated by an interrupt request.

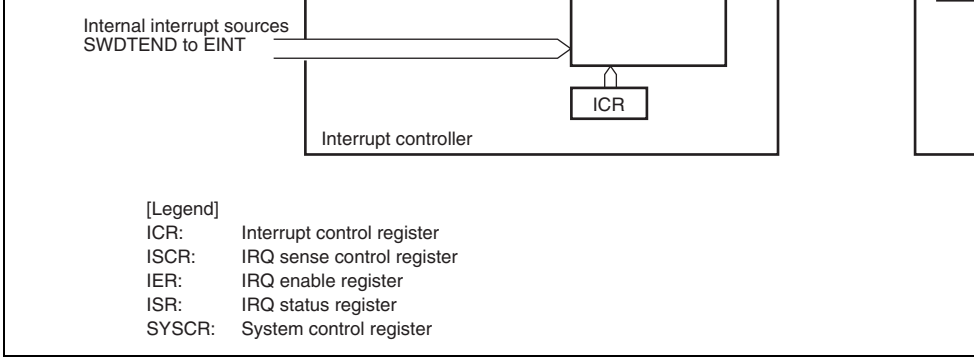


Figure 5.1 Block Diagram of Interrupt Controller

5.2 Input/Output Pins

Table 5.1 summarizes the pins of the interrupt controller.

Table 5.1 Pin Configuration

Symbol	I/O	Function
NMI	Input	Nonmaskable external interrupt Rising edge or falling edge can be selected
$\overline{\text{IRQ15}}$ to $\overline{\text{IRQ0}}$ $\overline{\text{ExIRQ15}}$ to $\overline{\text{ExIRQ0}}$	Input	Maskable external interrupts Rising edge, falling edge, or both edges, or level sensing c selected individually for each pin. Pin of $\overline{\text{IRQn}}$ or $\overline{\text{ExIRQn}}$ to $\overline{\text{IRQn}}$ (n = 15 to 0) interrupt can be selected.

- IRQ sense control registers (ISCR16H, ISCR16L, ISCRH, and ISCRL)
- IRQ enable registers (IER16 and IER)
- IRQ status registers (ISR16 and ISR)

5.3.1 Interrupt Control Registers A to D (ICRA to ICRD)

The ICR registers set interrupt control levels for interrupts other than NMI.

The correspondence between interrupt sources and ICRA to ICRD settings is shown in t

Bit	Bit Name	Initial Value	R/W	Description
7 to 0	ICRn7 to IRCn0	All 0	R/W	Interrupt Control Level 0: Corresponding interrupt source is interrupt co 0 (no priority) 1: Corresponding interrupt source is interrupt co 1 (priority)

[Legend]

n: A to D

0	ICRn0	WDT_1	IIC_4, IIC_5	USB*[1]	—

[Legend]

n: A to D

—: Reserved. The write value should always be 0.

Notes: 1. Supported only by the H8S/2472 Group.

2. Supported only by the H8S/2472 Group and the H8S/2462 Group.

5.3.2 Address Break Control Register (ABRKCR)

ABRKCR controls the address breaks. When both the CMF flag and BIE flag are set to 1 address break is requested.

Bit	Bit Name	Initial Value	R/W	Description
7	CMF	Undefined	R	Condition Match Flag Address break source flag. Indicates that an addr specified by BARA to BARC is prefetched. [Clearing condition] When an exception handling is executed for an a break interrupt. [Setting condition] When an address specified by BARA to BARC is prefetched while the BIE flag is set to 1.
6 to 1	—	All 0	R	Reserved These bits are always read as 0 and cannot be m
0	BIE	0	R/W	Break Interrupt Enable Enables or disables address break. 0: Disabled 1: Enabled

- BARB

Bit	Bit Name	Initial Value	R/W	Description
7 to 0	A15 to A8	All 0	R/W	Addresses 15 to 8 The A15 to A8 bits are compared with A15 to A8 internal address bus.

- BARC

Bit	Bit Name	Initial Value	R/W	Description
7 to 1	A7 to A1	All 0	R/W	Addresses 7 to 1 The A7 to A1 bits are compared with A7 to A1 in internal address bus.
0	—	0	R	Reserved This bit is always read as 0 and cannot be modif

5	IRQ14SCB	0	R/W	00: Interrupt request generated at low level of IRQ ExIRQn input
4	IRQ14SCA	0	R/W	01: Interrupt request generated at falling edge of ExIRQn input
3	IRQ13SCB	0	R/W	10: Interrupt request generated at rising edge of I ExIRQn input
2	IRQ13SCA	0	R/W	11: Interrupt request generated at both falling and edges of IRQn* or ExIRQn input
1	IRQ12SCB	0	R/W	(n = 15 to 12)
0	IRQ12SCA	0	R/W	

Note: * IRQn stands for IRQ15 to IRQ12.

- ISCR16L

Bit	Bit Name	Initial Value	R/W	Description
7	IRQ11SCB	0	R/W	IRQn Sense Control B IRQn Sense Control A
6	IRQ11SCA	0	R/W	00: Interrupt request generated at low level of IRQ ExIRQn input
5	IRQ10SCB	0	R/W	01: Interrupt request generated at falling edge of ExIRQn input
4	IRQ10SCA	0	R/W	10: Interrupt request generated at rising edge of I ExIRQn input
3	IRQ9SCB	0	R/W	11: Interrupt request generated at both falling and edges of IRQn* or ExIRQn input
2	IRQ9SCA	0	R/W	(n = 11 to 8)
1	IRQ8SCB	0	R/W	
0	IRQ8SCA	0	R/W	

Note: * IRQn stands for IRQ11 to IRQ8.

0	IRQ4SCA	0	R/W	11: Interrupt request generated at both falling ar edges of $\overline{\text{IRQn}}$ or $\overline{\text{ExIRQn}}$ input (n = 7 to 4)

Note: * $\overline{\text{IRQn}}$ stands for $\overline{\text{IRQ7}}$ to $\overline{\text{IRQ4}}$.

- ISCRL

Bit	Bit Name	Initial Value	R/W	Description
7	IRQ3SCB	0	R/W	IRQn Sense Control B
6	IRQ3SCA	0	R/W	IRQn Sense Control A
5	IRQ2SCB	0	R/W	00: Interrupt request generated at low level of $\overline{\text{IR}}$ $\overline{\text{ExIRQn}}$ input
4	IRQ2SCA	0	R/W	01: Interrupt request generated at falling edge o $\overline{\text{ExIRQn}}$ input
3	IRQ1SCB	0	R/W	10: Interrupt request generated at rising edge of $\overline{\text{ExIRQn}}$ input
2	IRQ1SCA	0	R/W	11: Interrupt request generated at both falling ar edges of $\overline{\text{IRQn}}$ or $\overline{\text{ExIRQn}}$ input (n = 3 to 0)
1	IRQ0SCB	0	R/W	
0	IRQ0SCA	0	R/W	

Note: * $\overline{\text{IRQn}}$ stands for $\overline{\text{IRQ3}}$ to $\overline{\text{IRQ0}}$.

- IER

Bit	Bit Name	Initial Value	R/W	Description
7 to 0	IRQ7E to IRQ0E	All 0	R/W	IRQn Enable (n = 7 to 0) The IRQn interrupt request is enabled when this b

				[Clearing conditions] • When reading 1, then writing 0 • When interrupt exception handling is execut low-level detection is set and $\overline{\text{IRQn}}$* or $\overline{\text{ExIR}}$ high • When IRQn interrupt exception handling is e when falling-edge, rising-edge, or both-edge is set (n = 15 to 8)

Note: * $\overline{\text{IRQn}}$ stands for $\overline{\text{IRQ15}}$ to $\overline{\text{IRQ8}}$.

- ISR

Bit	Bit Name	Initial Value	R/W	Description
7 to 0	IRQ7F to IRQ0F	All 0	R/W	[Setting condition] • When the interrupt source selected by the IS registers occurs [Clearing conditions] • When reading 1, then writing 0 • When interrupt exception handling is execut low-level detection is set and $\overline{\text{IRQn}}$* or $\overline{\text{ExIR}}$ high • When IRQn interrupt exception handling is e when falling-edge, rising-edge, or both-edge is set (n = 7 to 0)

Note: * $\overline{\text{IRQn}}$ stands for $\overline{\text{IRQ7}}$ to $\overline{\text{IRQ0}}$.

edge on the NMI pin.

IRQ15 to IRQ0 Interrupts: Interrupts IRQ15 to IRQ0 are requested by an input signal a
$\overline{\text{IRQ15}}$ to $\overline{\text{IRQ0}}$ or pins $\overline{\text{ExIRQ15}}$ to $\overline{\text{ExIRQ0}}$. Interrupts IRQ15 to IRQ0 have the followi
features:

- The interrupt exception handling for interrupt requests IRQ15 to IRQ0 can be started
  independent vector address.
- Using ISCR, it is possible to select whether an interrupt is generated by a low level, fa
  edge, rising edge, or both edges, at pins $\overline{\text{IRQ15}}$ to $\overline{\text{IRQ0}}$ or pins $\overline{\text{ExIRQ15}}$ to $\overline{\text{ExIRQ0}}$.
- Enabling or disabling of interrupt requests IRQ15 to IRQ0 can be selected with IER.
- The status of interrupt requests IRQ15 to IRQ0 is indicated in ISR. ISR flags can be c
  0 by software.

The detection of IRQ15 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, cle
corresponding port DDR to 0 so that it is not used as an I/O pin for another function.

A block diagram of interrupts IRQ15 to IRQ0 is shown in figure 5.2.

Figure 5.2 Block Diagram of Interrupts IRQ15 to IRQ0

5.4.2 Internal Interrupts

Internal interrupts issued from the on-chip peripheral modules have the following featur

- For each on-chip peripheral module there are flags that indicate the interrupt request and enable bits that individually select enabling or disabling of these interrupts. Whe enable bit for a particular interrupt source is set to 1, an interrupt request is sent to th controller.
- The control level for each interrupt can be set by ICR.
- The DTC can be activated by an interrupt request from an on-chip peripheral module
- An interrupt request that activates the DTC is not affected by the interrupt control m status of the CPU interrupt mask bits.

Table 5.3 Interrupt Sources, Vector Addresses, and Interrupt Priorities

Origin of Interrupt Source	Name	Vector Number	Vector Address: Advanced Mode	ICR	Pr
External pin	NMI	7	H'00001C	—	Hi
	IRQ0	16	H'000040	ICRA7	
	IRQ1	17	H'000044	ICRA6	
	IRQ2	18	H'000048	ICRA5	
	IRQ3	19	H'00004C		
	IRQ4	20	H'000050	ICRA4	
	IRQ5	21	H'000054		
	IRQ6	22	H'000058	ICRA3	
	IRQ7	23	H'00005C		
DTC	SWDTEND (Software activation data transfer end)	24	H'000060	ICRA2	
WDT_0	WOVI0 (Interval timer)	25	H'000064	ICRA1	
WDT_1	WOVI1 (Interval timer)	26	H'000068	ICRA0	
—	Address break	27	H'00006C	—	
A/D converter	ADI (A/D conversion end)	28	H'000070	ICRB7	
EVC	EVENTI	29	H'000074	—	
TMR_X	CMIAX (Compare match A)	44	H'0000B0	ICRB4	
	CMIBX (Compare match B)	45	H'0000B4		
	OVIX (Overflow)	46	H'0000B8		
FRT	OCIA (Output compare A)	52	H'0000D0	ICRB6	
	OCIB (Output compare B)	53	H'0000D4		
	FOVI (Overflow)	54	H'0000D8		Lo

	CMIB0 (Compare match B) OVI0 (Overflow)	65 66	H'000104 H'000108	
TMR_1	CMIA1 (Compare match A) CMIB1 (Compare match B) OVI1 (Overflow)	68 69 70	H'000110 H'000114 H'000118	ICRB2
TMR_Y	CMIAY (Compare match A) CMIBY (Compare match B) OVIY (Overflow)	72 73 74	H'000120 H'000124 H'000128	ICRB1
IIC_2	IICI2	76	H'000130	ICRC2
IIC_3	IICI3	78	H'000138	
SCI_3	ERI3 (Reception error 3) RXI3 (Reception completion 3) TXI3 (Transmission data empty 3) TEI3 (Transmission end 3)	80 81 82 83	H'000140 H'000144 H'000148 H'00014C	ICRC7
SCI_1	ERI1 (Reception error 1) RXI1 (Reception completion 1) TXI1 (Transmission data empty 1) TEI1 (Transmission end 1)	84 85 86 87	H'000150 H'000154 H'000158 H'00015C	ICRC6
SSU	ERIS (Reception error S) RXIS (Reception completion S) TXIS (Transmission data empty S)	88 89 90	H'000160 H'000164 H'000168	ICRC5
SCIF	SCIFI	92	H'000170	ICRD1
IIC_0	IICI0	94	H'000178	ICRC4
IIC_1	IICI1	98	H'000188	ICRC3
IIC_4	IICI4	100	H'000190	ICRB0
IIC_5	IICI5	102	H'000198	ICRB0
LPC	ERR1(transfer error, etc.) IBFI1 (IDR1 reception completion) IBFI2 (IDR2 reception completion) IBFI3 (IDR3 reception completion)	104 105 106 107	H'0001A0 H'0001A4 H'0001A8 H'0001AC	ICRC1

Notes: 1. Supported only by the H8S/2472 Group.
2. Supported only by the H8S/2472 Group and the H8S/2462 Group.

0	0	0	ICR	I	Interrupt mask control is perfo the I bit. Priority levels can be ICR.
1		1	ICR	I, UI	3-level interrupt mask control performed by the I and UI bits levels can be set with ICR.

Figure 5.3 shows a block diagram of the priority decision circuit.

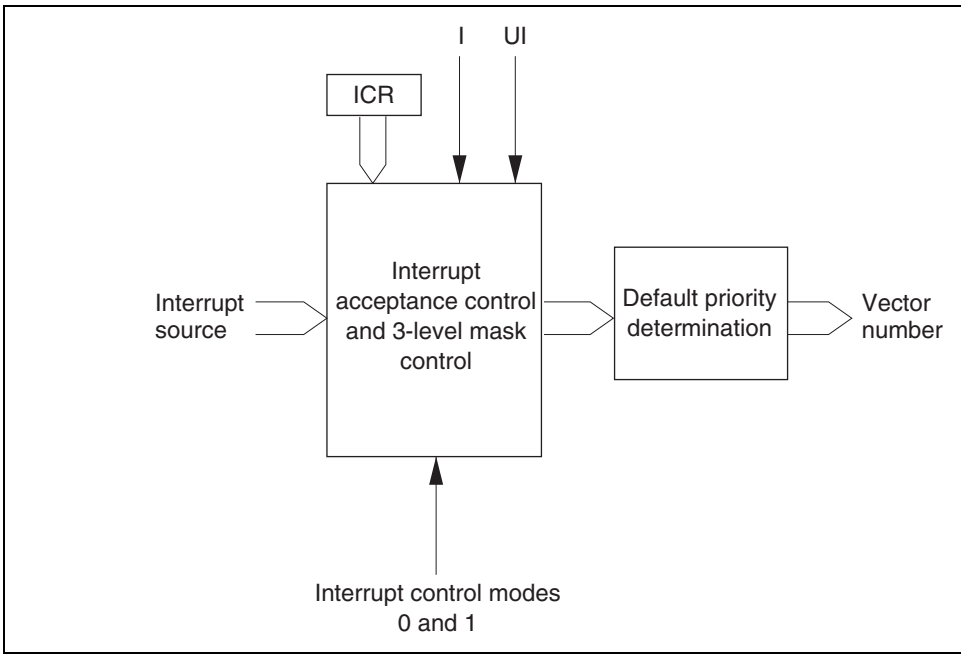


Figure 5.3 Block Diagram of Interrupt Control Operation

			priority)
	1	*	NMI and address break interrupts
1	0	*	All interrupts (interrupt control level 1 ha priority)
	1	0	NMI, address break, and interrupt contr interrupts
		1	NMI and address break interrupts

[Legend]

* Don't care

Default Priority Determination: The priority is determined for the selected interrupt, an vector number is generated.

If the same value is set for ICR, 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 interrupt source are held pending

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

PR: Sets priority
—: Not used

5.6.1 Interrupt Control Mode 0

In interrupt control mode 0, interrupts other than NMI are masked by ICR and the I bit o in the CPU. Figure 5.4 shows a flowchart of the interrupt acceptance operation.

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. According to the interrupt control level specified in ICR, the interrupt controller acc interrupt request with interrupt control level 1 (priority), and holds pending an interr with interrupt control level 0 (no priority). If several interrupt requests are issued, an request with the highest priority is accepted according to the priority order, an interru handling is requested to the CPU, and other interrupt requests are held pending.
3. If the I bit in CCR is set to 1, only NMI and address break interrupt requests are acce the interrupt controller, and other interrupt requests are held pending. If the I bit is cl any interrupt request is accepted. KIN, WUE, and EVENTI interrupts are enabled or by the I bit.
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 for NMI and addre interrupts.

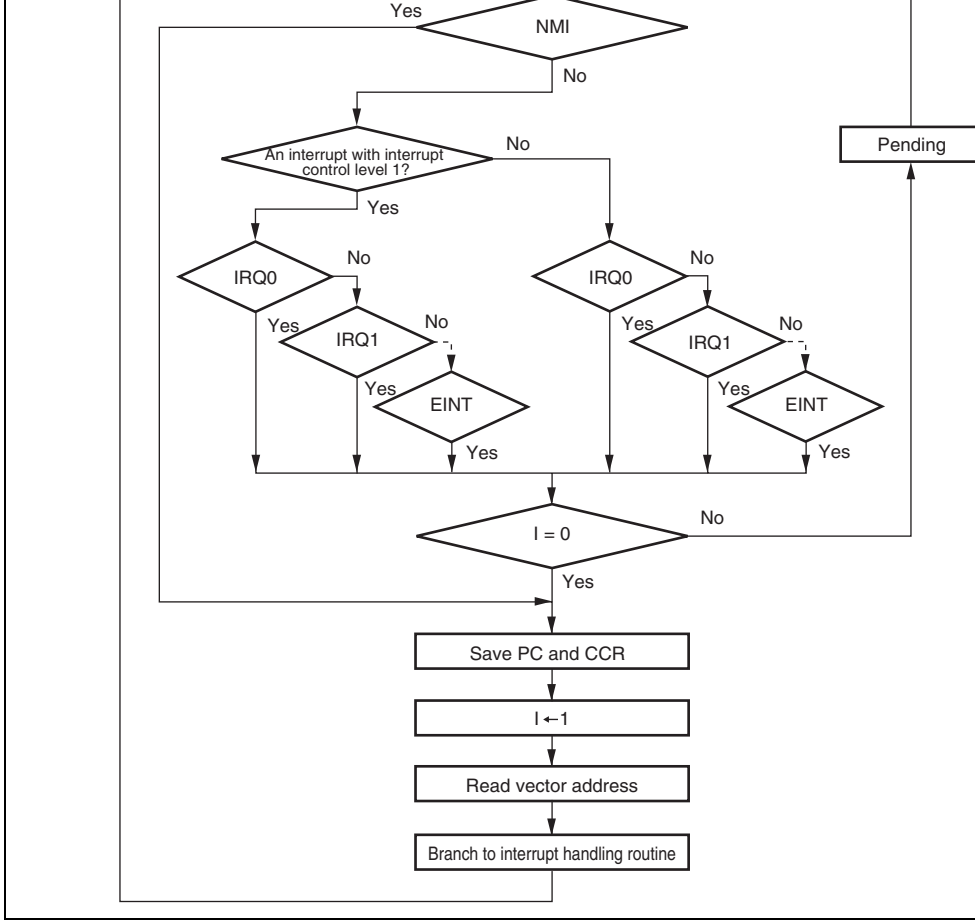


Figure 5.4 Flowchart of Procedure up to Interrupt Acceptance in Interrupt Contro

For instance, the state when the interrupt enable bit corresponding to each interrupt is se
ICRA to ICRD are set to H'20, H'00, H'00, and H'00, respectively (IRQ2 and IRQ3 inter
set to interrupt control level 1, and other interrupts are set to interrupt control level 0) is
below. Figure 5.6 shows a state transition diagram.

- All interrupt requests are accepted when I = 0. (Priority order: NMI > IRQ2 > IRQ3
  IRQ1 > address break …)
- Only NMI, IRQ2, IRQ3, and address break interrupt requests are accepted when I =
  0.
- Only NMI and address break interrupt requests are accepted when I = 1 and UI = 1.

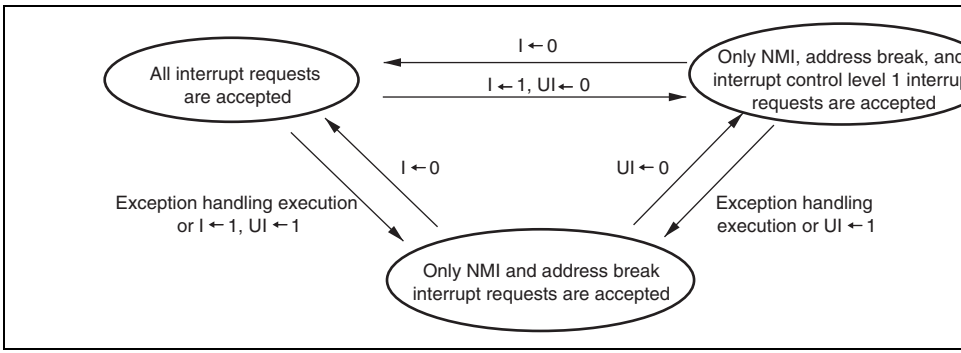


Figure 5.5 State Transition in Interrupt Control Mode 1

An interrupt request with interrupt control level 0 is accepted when the I bit is cleared

When both the I and UI bits are set to 1, only NMI and address break interrupt reques accepted, and other interrupts are held pending.

When the I bit is cleared to 0, the UI bit is not affected.

4. When the CPU accepts an interrupt request, it starts interrupt exception handling after 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. The I and UI bits in CCR are set to 1. This masks all interrupts except for NMI and ad break interrupts.
7. The CPU generates a vector address for the accepted interrupt and starts execution of interrupt handling routine at the address indicated by the contents of the vector addres vector table.

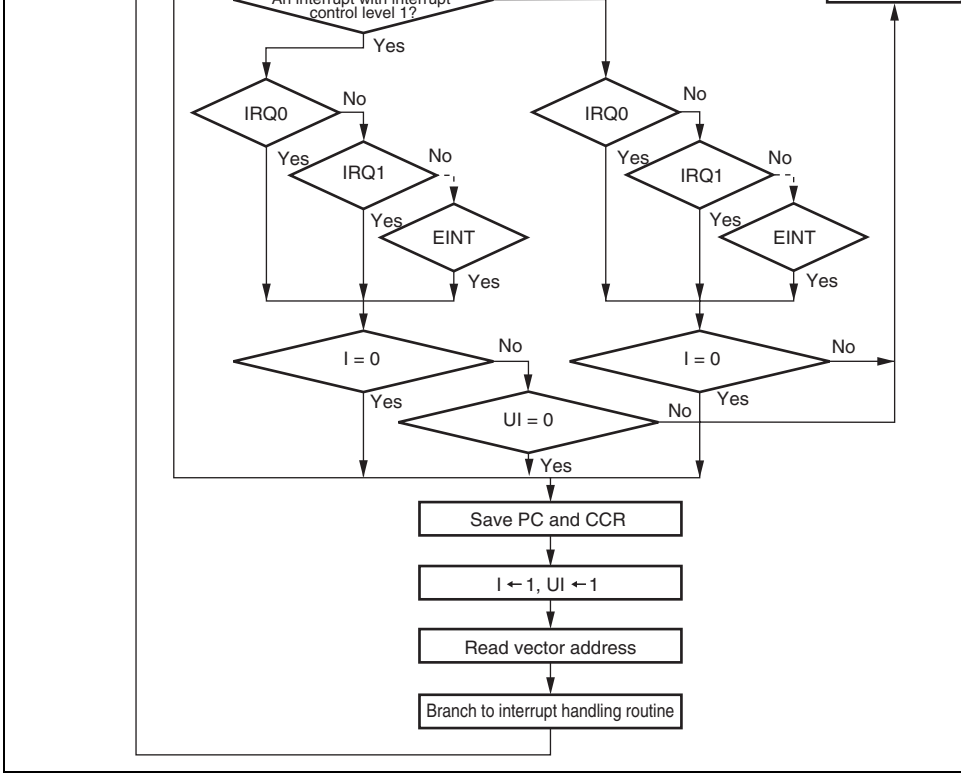


Figure 5.6 Flowchart of Procedure Up to Interrupt Acceptance in Interru Control Mode 1

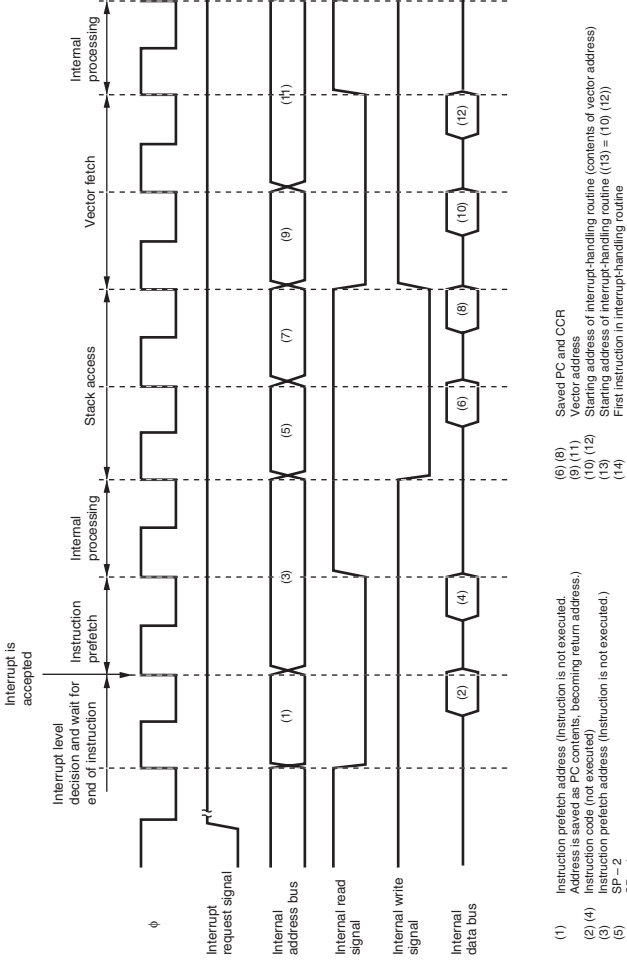


Figure 5.7 Interrupt Exception Handling

3	PC, CCR stack save	$2 \cdot S_K$
4	Vector fetch	$2 \cdot S_I$
5	Instruction fetch*[3]	$2 \cdot S_I$
6	Internal processing*[4]	2
	Total (using on-chip memory)	12 to 32

Notes: 1. Two states in case of internal interrupt.
2. Refers to MULXS and DIVXS instructions.
3. Prefetch after interrupt acceptance and prefetch of interrupt handling routine.
4. Internal processing after interrupt acceptance and internal processing after ve

Table 5.8 Number of States in Interrupt Handling Routine Execution Status

	Object of Access				
		External Device			
		8-Bit Bus		16-Bit	
Symbol	Internal Memory	2-State Access	3-State Access	2-State Access	3
Instruction fetch S_I	1	4	6 + 2m	2	3
Branch address read S_J					
Stack manipulation S_K					

[Legend]
m: Number of wait states in external device access.

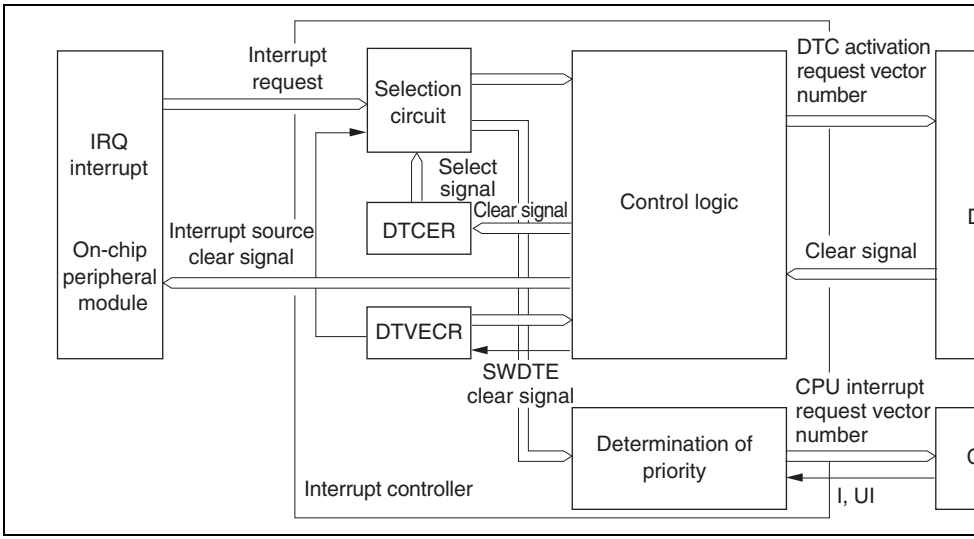


Figure 5.8 Interrupt Control for DTC

The interrupt controller has three main functions in DTC control.

(1) Selection of Interrupt Source

It is possible to select DTC activation request or CPU interrupt request with the DTCE bi DTCERA to DTCERF in the DTC. After a DTC data transfer, the DTCE bit can be clear and an interrupt request sent to the CPU in accordance with the specification of the DISE MRB in the DTC. When the DTC performs the specified number of data transfers and the counter reaches 0, following the DTC data transfer the DTCE bit is cleared to 0 and an in request is sent to the CPU.

the settings of the DTCE bit of DTCERA to DTCERF in the DTC and the DISEL bit of the DTC.

Table 5.9 Interrupt Source Selection and Clearing Control

Settings			
DTC		Interrupt Source Selection/Clearing	
DTCE	DISEL	DTC	CPU
0	X	×	Δ
1	0	Δ	×
	1	○	Δ

[Legend]

Δ: The relevant interrupt is used. Interrupt source clearing is performed.
(The CPU should clear the source flag in the interrupt handling routine.)

○: The relevant interrupt is used. The interrupt source is not cleared.

×: The relevant interrupt cannot be used.

X: Don't care

handling will be executed for the higher-priority interrupt, and the lower-priority interrup
ignored. The same rule is also applied when an interrupt source flag is cleared to 0. Figur
shows an example in which the CMIEA bit in the TMR's TCR register is cleared to 0.

The above conflict will not occur if an enable bit or interrupt source flag is cleared to 0 w
interrupt is masked.

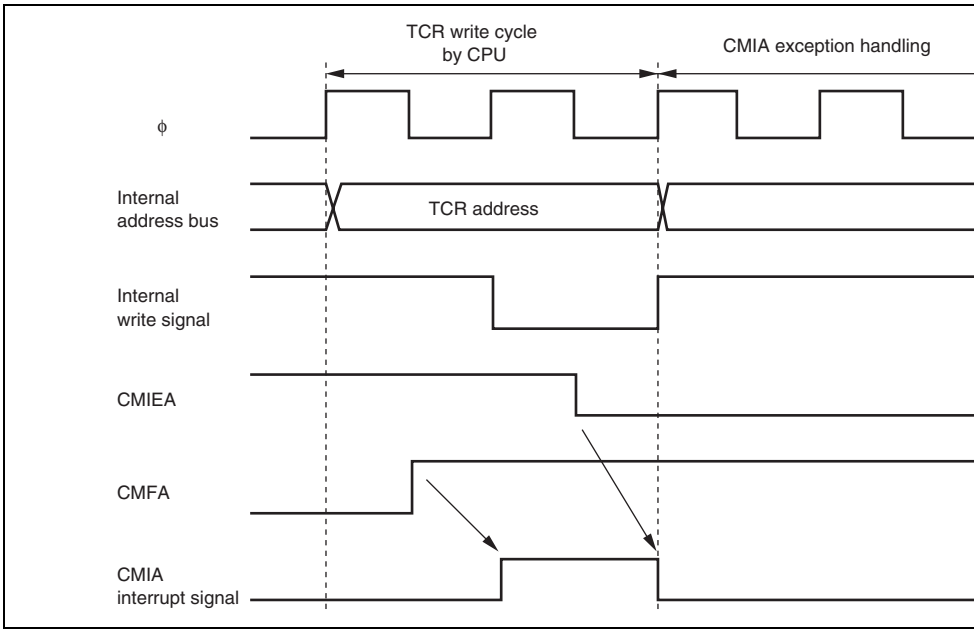


Figure 5.9 Conflict between Interrupt Generation and Disabling

With the EEPMOV.B instruction, an interrupt request (including NMI) issued during the
is not accepted until the move is completed.

With the EEPMOV.W instruction, if an interrupt request is issued during the transfer, in
exception handling starts at a break in the transfer cycle. The PC value saved on the stac
case is the address of the next instruction. Therefore, if an interrupt is generated during e
of an EEPMOV.W instruction, the following coding should be used.

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

5.7.4 IRQ Status Registers (ISR16, ISR)

Since IRQnF may be set to 1 according to the pin status after a reset, the ISR16 and the
should be read after a reset, and then write 0 in IRQnF (n = 15 to 0).

Two modes for external extension

Normal extended mode: Normal extension
(when ADMXE = 0 in SYSCR2 and OBE = 0 in PTCNT0)
Glueless extension
(when ADMXE = 0 in SYSCR2 and OBE = 1 in PTCNT0)

Address-data multiplex extended mode: Multiplex extension (when ADMXE = 1 in

- Extended area division

  Possible in normal extended mode

  The external address space can be accessed as basic extended areas.

  A 256-Kbyte extended area can be set and controlled independently of basic extende

- Address pin reduction

  In normal extended mode:

  A 256-Kbyte extended area from H'F80000 to H'FBFFFF can be selected using 18 a pins and the $\overline{\text{CS256}}$ signal.

  A 2-Kbyte area from H'FFF000 to H'FFF7FF can be selected using six to eleven add and the $\overline{\text{IOS}}$ signal.

  In address-data multiplex extended mode:

  The external address space can be accessed as the following two extended areas.

H'F80000 to H'F8FFFF	64 Kbytes	256-Kbyte extended area
H'FFF000 to H'FFF7FF	2 Kbytes	IOS extended area

  These areas can be selected using 8 pins or 16 pins, which is a total of address pins a input/output pins.

- Control address hold signal and area select signal polarity

  The output polarity of $\overline{\text{IOS}}$, $\overline{\text{CS256}}$, and $\overline{\text{AH}}$ can be inverted by the PNCCS and PNC LPWRCR

Program wait states can be inserted for each area.

- Burst ROM interface

  In normal extended mode

  A burst ROM interface can be set for basic extended areas.

  1-state access or 2-state access can be selected for burst access.

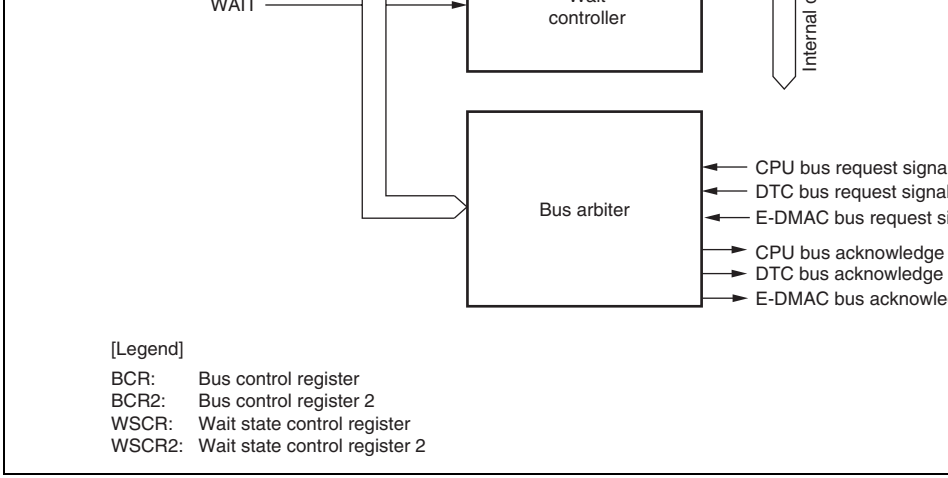
- Idle cycle insertion

  In normal extended mode

  An idle cycle can be inserted for external write cycles immediately after external read

- Bus arbitration function

  Includes a bus arbiter that arbitrates bus mastership between the CPU, DTC, and E-D



[Legend]

BCR:	Bus control register
BCR2:	Bus control register 2
WSCR:	Wait state control register
WSCR2:	Wait state control register 2

Figure 6.1 Block Diagram of Bus Controller

$\overline{\text{IOS}}$	Output	Chip select signal indicating that the IOS extended a being accessed (when the IOSE bit in SYSCR is 1).
$\overline{\text{CS256}}$	Output	Chip select signal indicating that the 256-Kbyte exte area is being accessed (when the CS256E bit in SY 1).
$\overline{\text{RD}}$	Output	Strobe signal indicating that the external address spa being read.
$\overline{\text{HWR}}$	Output	Strobe signal indicating that the external address spa being written to, and the upper half (D15 to D8, AD1 AD8) of the data bus is valid.
$\overline{\text{LWR}}$	Output	Strobe signal indicating that the external address spa being written to, and the lower half (D7 to D0, AD7 to the data bus is valid.
$\overline{\text{WAIT}}$	Input	Wait request signal when accessing the external spa
$\overline{\text{WR}}$	Output	Strobe signal indicating that the external address spa being written to.
$\overline{\text{HBE}}$	Output	Strobe signal indicating that the external address spa being accessed, and the upper half (D15 to D8) of th bus is valid.
$\overline{\text{LBE}}$	Output	Strobe signal indicating that the external address spa being accessed, and the lower half (D7 to D0) of the bus is valid.
$\overline{\text{AH}}$	Output	Signal indicating address fetch timing when the bus address-data multiplex bus state.
AD15 to AD0	Input/Output	Address output and data input/output pins for addres multiplex extension.

- System control register 2 (SYSCR2)

6.3.1 Bus Control Register (BCR)

BCR is used to specify the access mode for the external address space and the I/O area r
the $\overline{AS}$/$\overline{IOS}$ pin is specified as an I/O strobe pin.

Bit	Bit Name	Initial Value	R/W	Description
7	—	1	R/W	Reserved The initial value should not be changed.
6	ICIS	1	R/W	Idle Cycle Insertion Selects whether or not to insert 1-state of the idl between successive external read and external cycles. 0: Idle cycle not inserted 1: 1-state idle cycle inserted
5	BRSTRM	0	R/W	Valid only in the normal extended mode. Burst ROM Enable Selects the bus interface for the external addres 0: Basic bus interface 1: Burst ROM interface When the CS256E bit in SYSCR is set to 1, burs interface cannot be selected for the 256-Kbyte e area.

				Selects the number of words that can be accesse burst access via the burst ROM interface. 0: Max, 4 words 1: Max, 8 words
2	—	0	R/W	Reserved The initial value should not be changed.
1	IOS1	1	R/W	IOS Select 1 and 0
0	IOS0	1	R/W	Select the address range where the $\overline{\text{IOS}}$ signal is See table 6.12.

3	ADFULLE	0	R/W	Address Output Full Enable Controls the address output, A23 to A21, in acce extended area. See section 8, I/O Ports. This is supported while ADMXE = 1.
2	EXCKS	0	R/W	External Extension Clock Select Selects the operating clock used in external exte area access. 0: Medium-speed clock is selected as the opera 1: System clock (ϕ) is selected as the operating The operating clock is switched in the bus cycle external extended area access.
1	—	1	R/W	Reserved The initial value should not be changed.
0	—	0	R/W	Reserved The initial value should not be changed.

				extended area when the CS256E bit in SYSCR is 0: 16-bit bus 1: 8-bit bus
6	AST256	1	R/W	256-Kbyte Extended Area Access State Control Selects the number of states for access to the 25 extended area when the CS256E bit in SYSCR is This bit also enables or disables wait-state inserti [ADMXE = 0] Normal extension 0: 2-state access space. Wait state insertion disa 1: 3-state access space. Wait state insertion enal [ADMXE = 1] Address-data multiplex extension 0: 2-state data access space. Wait state insertion 1: 3-state data access space. Wait state insertion
5	ABW	1	R/W	Basic Extended Area Bus Width Control Selects the bus width for access to the basic exte area. 0: 16-bit bus 1: 8-bit bus When the CS256E bit in SYSCR is set to 1, this b is ignored in access to the 256-Kbyte extended a

				0: 2-state data access space. Wait state insertio 1: 3-state data access space. Wait state insertio enabled When the CS256E bit in SYSCR is set to 1, this is ignored in access to the 256-Kbyte extended
3	WMS1	0	R/W	Basic Extended Area Wait Mode Select 1 and 0
2	WMS0	0	R/W	Selects the wait mode for access to the basic ex area when the AST bit is set to 1. 00: Program wait mode 01: Wait disabled mode 10: Pin wait mode 11: Pin auto-wait mode When the CS256E bit in SYSCR is set to 1, this is ignored in access to the 256-Kbyte extended
1	WC1	1	R/W	Basic Extended Area Wait Count 1 and 0
0	WC0	1	R/W	Selects the number of program wait states to be when the basic extended area is accessed when bit is set to 1. The program wait state is only ins data cycles. 00: Program wait state is not inserted 01: 1 program wait state is inserted 10: 2 program wait states are inserted 11: 3 program wait states are inserted When the CS256E bit in SYSCR is set to 1, this is ignored in access to the 256-Kbyte extended

				0: Program wait mode 1: Wait disabled mode
6	WC11	1	R/W	256-Kbyte Extended Area Wait Count 1 and 0
5	WC10	1	R/W	Selects the number of program wait states to be i into the data cycle for access to the 256-Kbyte ex area when the CS256E bit in SYSCR and the AS in WSCR are set to 1. 00: Program wait state is not inserted 01: 1 program wait state is inserted 10: 2 program wait states are inserted 11: 3 program wait states are inserted
4	—	All 0	R/W	Reserved
3				

				Selects the number of program wait states to be into the address cycle for access to the address-multiplex extended area. 0: Program wait state is not inserted 1: 1 program wait state is inserted in the address
1, 0	—	All 1	R/W	Reserved

6.3.5 System Control Register 2 (SYSCR2)

SYSCR2 controls the address-data multiplex operation.

Bit	Bit Name	Initial Value	R/W	Description
7 to 4	—	All 0	R/W	Reserved The initial value should not be changed.
3	ADMXE	0	R/W	Address-Data Multiplex Bus Interface Enabl 0: Normal extended bus interface 1: Address data multiplex extended bus inte
2 to 0	—	All 0	R/W	Reserved The initial value should not be changed.

(a) Bus Width

A bus width of 8 or 16 bits can be selected via the ABW and ABW256 bits in WSCR.

(b) Number of Access States

Two or three access states can be selected via the AST and AST256 bits in WSCR. When state access space is designated, wait-state insertion is disabled.

In the burst ROM interface, the number of access states for the basic extended area is det regardless of the AST bit setting.

(c) Wait Mode and Number of Program Wait States

When the basic extended area is specified as a 3-state access space by the AST bit in WS wait mode and the number of program wait states to be inserted automatically is selected WMS1, WMS0, WC1, and WC0 bits in WSCR. From 0 to 3 program wait states can be s

When the 256-Kbyte extended area is specified as a 3-state access space by the AST256 b WSCR, the wait mode and the number of program wait states to be inserted automatically selected by the WMS10, WC11, and WC10 bits in WSCR2. From 0 to 3 program wait sta be selected.

The wait function for external extension is effective for connecting low-speed devices to external address space. However, this wait function may cause some problems when the of bus masters other than the CPU, such as the DTC are to be delayed.

Tables 6.2 to 6.5 show each bit setting and external address space division in the address the external address space, and the bus specifications for the basic bus interface of each a

H'F80000 to H'FBFFFF (256 Kbytes) 256-Kbyte extended area	Δ: When CS256E = 0, used as basic extended area.	When WAIT pin functi selected while CS256 CS256 is output and a pins A17 to A0 are us
H'FC0000 to H'FEFFFF (192 Kbytes)	○: No condition	—
H'FF0800 to H'FFBFFF (46 Kbytes)	Δ: When RAME = 0, used as basic extended area.	—
H'FFC000 to H'FFDFFF (8 Kbytes)	○: No condition	—
H'FFE000 to H'FFE07F (128 bytes)	○: No condition.	—
H'FFE080 to H'FFEFFF (3968 bytes)	Δ: When RAME = 0, used as basic extended area.	—
H'FFF000 to H'FFF7FF (2 Kbytes)	○ No condition When IOSE = 1, IOS is output and address pins A10 to A0 are used.	—
H'FFFF00 to H'FFFF7F (128 bytes)	Δ When RAME = 0, used as basic extended area.	—

[Legend]

○: This address range is unconditionally accessed as the basic extended area.

Δ: Condition for making this address range accessed as the basic extended area.

—: This address range cannot be used as part of a 256-Kbyte extended area.

Note: * In the burst ROM interface, the bus width is specified by the ABW bit in WSCH number of full access states (wait can be inserted) is specified by the AST bit i and the number of access cycles in burst access is specified regardless of the setting.

Table 6.4 Bus Specifications for Basic Extended Area/Basic Bus Interface

						Bus Specifications		
ABW	**AST**	**WMS1**	**WMS0**	**WC1**	**WC0**	**Bus Width**	**Number of Access States**	**Nu Pro Wa Sta**
0	0	X	X	X	X	16	2	0
	1	0	1	X	X	16	3	0
		Other than WMS1 = 0 and WMS0 = 1		0	0		3	0
					1			1
				1	0			2
					1			3
1	0	X	X	X	X	8	2	0
	1	0	1	X	X	8	3	0
		Other than WMS1 = 0 and WMS0 = 1		0	0		3	0
					1			1
				1	0			2
					1			3

[Legend]
X: Don't care

			1	0			2
				1			3
1	0	X	X	X	8	2	0
	1	1	X	X	8	3	0
		0	0	0		3	0
				1			1
			1	0			2
				1			3

[Legend]

X: Don't care

- IOS Extended Area

  When the IOS extended area is specified as a 3-state access space by the AST bit in W
  the wait mode and the number of program wait states to be inserted automatically is s
  by the WMS1, WMS0, WC1, and WC0 bits in WSCR. Zero or one program wait stat
  inserted into address cycle. From zero to three program wait states can be selected for
  cycle.

- 256-Kbyte Extended Area

  When the 256-Kbyte extended area is specified as a 3-state access space by the AST2
  WSCR, the wait mode and the number of program wait states to be inserted automatic
  selected by the WMS10, WC11, and WC10 bits in WSCR2. Zero or one program wai
  can be inserted into address cycle. From zero to three program wait states can be selec
  data cycle.

The wait function for external extension is effective for connecting low-speed devices to
external address space. However, this wait function may cause some problems when the
of bus masters other than the CPU, such as the DTC, are to be delayed.

Tables 6.6 to 6.11 show address-data multiplex address space and the bus specifications f
basic bus interface of each area.

256-Kbyte extended area H'FA0000 to H'FAFFFF (64 Kbytes)	—	No condition
256-Kbyte extended area H'FB0000 to H'FBFFFF (64 Kbytes)	—	No condition
H'FC0000 to H'FFBFFF (240 Kbytes)	—	No condition
H'FFC000 to H'FFDFFF (8 Kbytes)	—	No condition
H'FFE000 to H'FFEFFF (4 Kbytes)	—	No condition
IOS extended area H'FFF000 to H'FFF7FF (2 Kbytes)	O	When IOSE = 1, $\overline{\text{IOS}}$ is output and address pins AD0 or AD7 to AD0 are used.
H'FFFF00 to H'FFFF7F (128 bytes)	—	No condition

[Legend]

—: This address range cannot be used as the address-data multiplex address space

O: Condition for making this address range accessed as the address-data multiplex space.

Table 6.8 Bus Specifications for IOS Extended Area/Multiplex Bus Interface (Ad Cycle)

AST	WMS1	WMS0	WC22	WC1	WC0	Number of Access States	Num Prog Wai
—	—	—	0	—	—	2	0
			1	—	—		1

Table 6.9 Bus Specifications for IOS Extended Area/Multiplex Bus Interface (Dat

AST	WMS1	WMS0	WC1	WC0	Number of Access States	Nur Pro Wai
0	—	—	—	—	2	0
1	0	1	—	—	3	0
	Other than WMS1 = 0 and WMS0 = 1		0	0	3	0
				1		1
			1	0		2
				1		3

AST256	WMS1	WC1	WC0	Number of Access States	Num Prog State
0	—	—	—	2	0
1	1	—	—	3	0
	0	0	0	3	0
			1		1
		1	0		2
			1		3

6.4.2 Advanced Mode

The external address space (H'FFF000 to H'FFF7FF) can be accessed by specifying the
pin as an I/O strobe pin. The 256-Kbyte extended area (H'F80000 to H'FBFFFF) can be
by the $\overline{\text{CS256}}$ pin function.

The external address space is initialized as the basic bus interface and a 3-state access sp
mode 2, the address space other than on-chip ROM, on-chip RAM, internal I/O registers
reserved areas is specified as the external address space. The on-chip RAM and its reser
are enabled when the RAME bit in SYSCR is set to 1, and disabled when the RAME bit
to 0. Addresses H'FF0800 to H'FFBFFF, H'FFE080 to H'FFEFFF, and H'FFFF00 to H'F
the on-chip RAM area and its reserved area are always specified as the external address

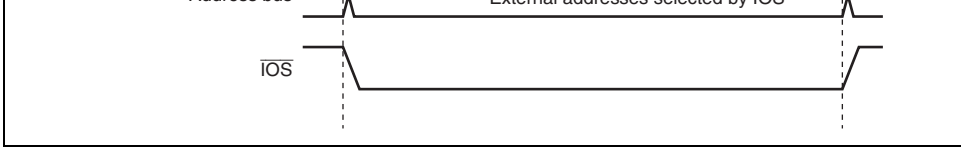


Figure 6.2 $\overline{\text{IOS}}$ Signal Output Timing

Enabling or disabling $\overline{\text{IOS}}$ signal output is performed by the IOSE bit in SYSCR. In the e mode, the $\overline{\text{IOS}}$ pin functions as an $\overline{\text{AS}}$ pin by a reset. To use this pin as an $\overline{\text{IOS}}$ pin, set the bit to 1. For details, see section 8, I/O Ports.

The address ranges of the $\overline{\text{IOS}}$ signal output can be specified by the IOS1 and IOS0 bits i as shown in table 6.12.

Table 6.12 Address Range for $\overline{\text{IOS}}$ Signal Output

IOS1	IOS0	$\overline{\text{IOS}}$ Signal Output Range	
0	0	H'FFF000 to H'FFF03F	
	1	H'FFF000 to H'FFF0FF	
1	0	H'FFF000 to H'FFF3FF	
	1	H'FFF000 to H'FFF7FF	(Initi

6.3.1 Data Size and Data Alignment

Data sizes for the CPU and other internal bus masters are byte, word, and longword. The
a data alignment function, and controls whether the upper data bus (D15 to D8/AD15 to
lower data bus (D7 to D0/AD7 to AD0) is used when the external address space is acces
according to the bus specifications for the area being accessed (8-bit access space or 16-
space) and the data size.

(1) 8-Bit Access Space

Figure 6.3 illustrates data alignment control for the 8-bit access space. With the 8-bit acc
the upper data bus (D15 to D8) is always used for accesses. The amount of data that can
accessed at one time is one byte: a word access is performed as two byte accesses, and a
access, as four byte accesses.

The lower data bus (AD7 to AD0) is used in address-data multiplex extended mode.

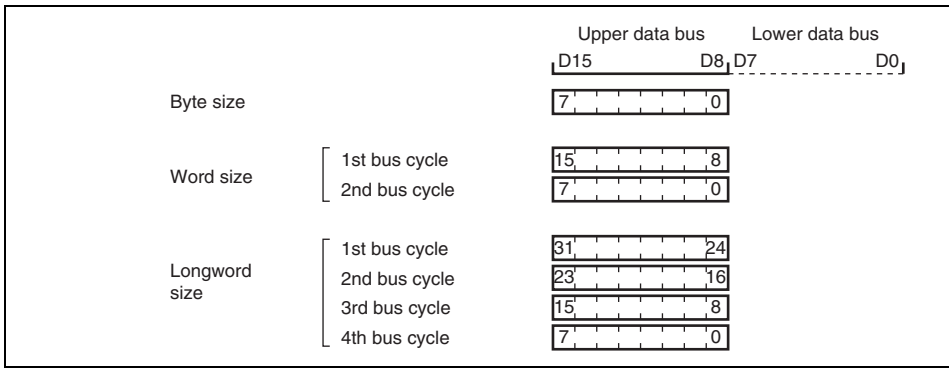


Figure 6.3 Access Sizes and Data Alignment Control (8-bit Access Space

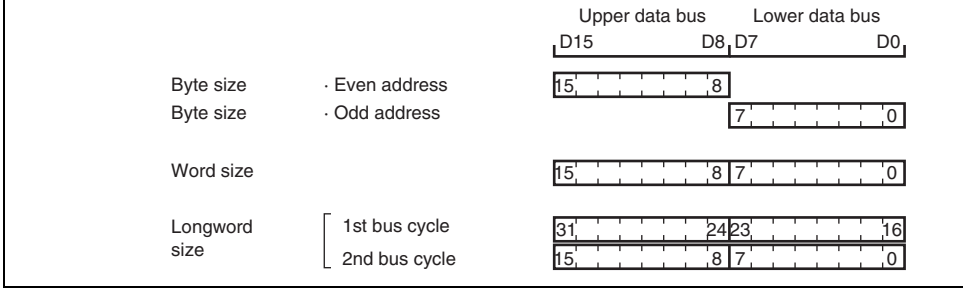


Figure 6.4 Access Sizes and Data Alignment Control (16-bit Access Space)

Area	Size	Write	Address	Strobe	AD15 to AD8)	AD0)
8-bit access space	Byte	Read	—	$\overline{RD}$	Valid	Ports
		Write	—	$\overline{HWR}$		
8-bit access space (in address-data multiplex extended mode)	Byte	Read	—	$\overline{RD}$	Ports or others	Valid
		Write	—	$\overline{HWR}$		
16-bit access space	Byte	Read	Even	$\overline{RD}$	Valid	Invali
			Odd		Invalid	Valid
		Write	Even	$\overline{HWR}$	Valid	Unde
			Odd	$\overline{LWR}$	Undefined	Valid
	Word	Read	—	$\overline{RD}$	Valid	Valid
		Write	—	$\overline{HWR}$, $\overline{LWR}$		

[Legend]

Undefined: Undefined data is output.

Invalid: Input state with the input value ignored.

Ports or others: Used as ports or I/O pins for on-chip peripheral modules, and are not u data bus.

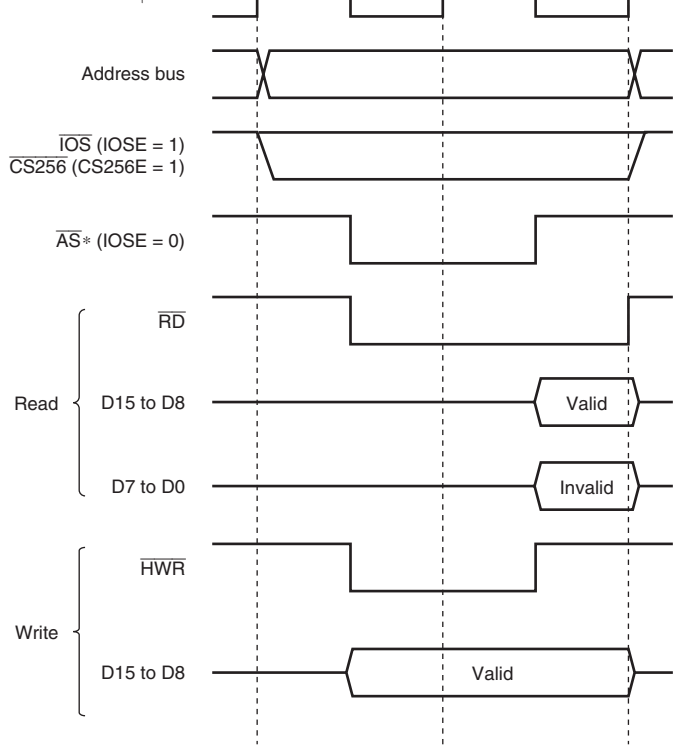
8-bit access space	Byte	Read	—	RD	Valid	Ports o
		Write	—	WR		
16-bit access space	Byte	Read	Even	RD, HBE	Valid	Invalid
			Odd	RD, LBE	Invalid	Valid
		Write	Even	WR, HBE	Valid	Undefi
			Odd	WR, LBE	Undefined	Valid
	Word	Read	—	RD, HBE, LBE	Valid	Valid
		Write	—	WR, HBE, LBE		

[Legend]

Undefined: Undefined data is output.

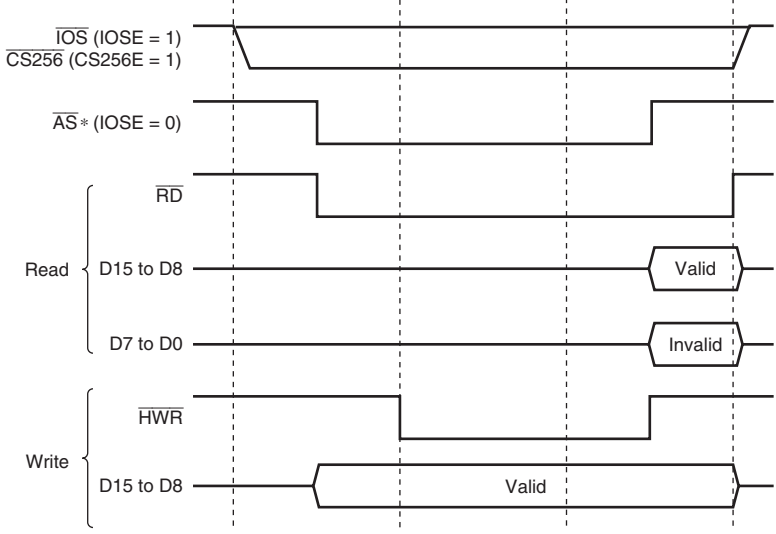
Invalid: Input state with the input value ignored.

Ports or others: Used as ports or I/O pins for on-chip peripheral modules, and are not us data bus.



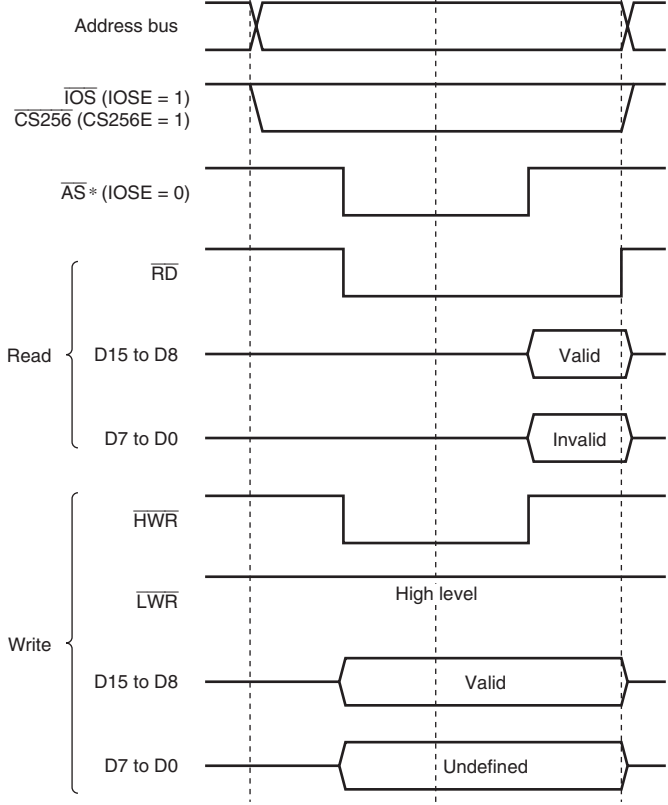
Note: * For external address space access, this signal is not output when the 256-Kbyte extended are
is accessed with CS256E = 1.

Figure 6.5 Bus Timing for 8-Bit, 2-State Access Space



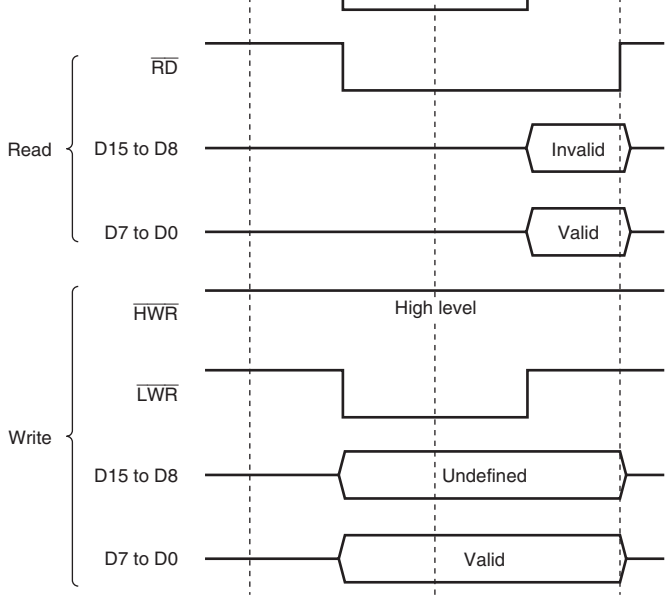
Note: * For external address space access, this signal is not output when the 256-Kbyte extended area is accessed with CS256E = 1.

Figure 6.6 Bus Timing for 8-Bit, 3-State Access Space



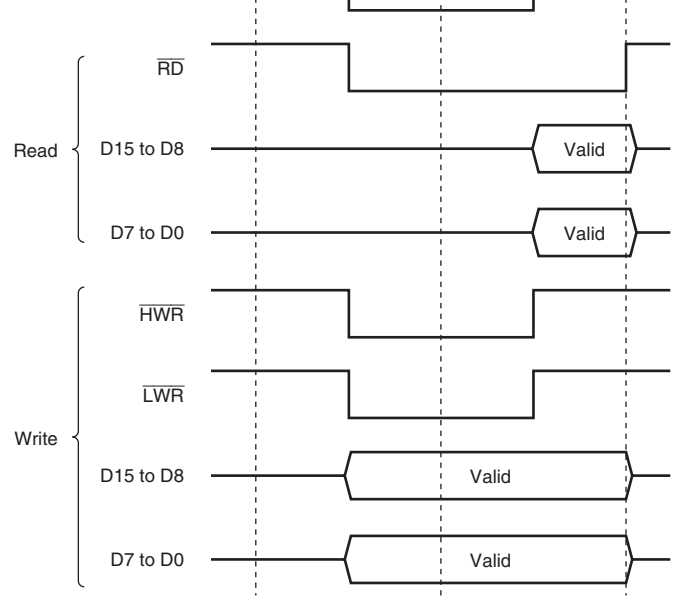
Note: * For external address space access, this signal is not output when the 256-Kbyte extended are is accessed with CS256E = 1.

Figure 6.7 Bus Timing for 16-Bit, 2-State Access Space (Even Byte Acces



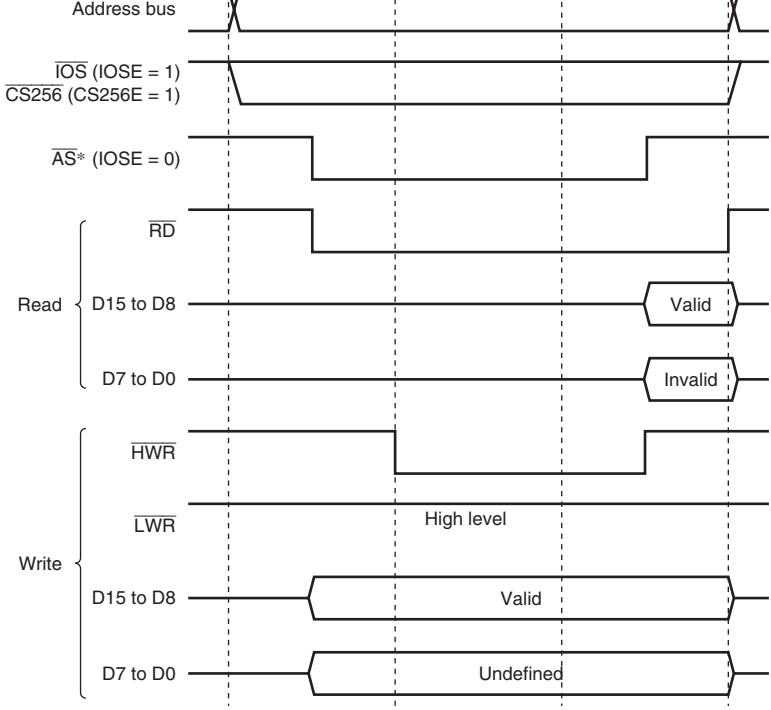
Note: * For external address space access, this signal is not output when the 256-Kbyte extended area is accessed with CS256E = 1.

Figure 6.8 Bus Timing for 16-Bit, 2-State Access Space (Odd Byte Access)



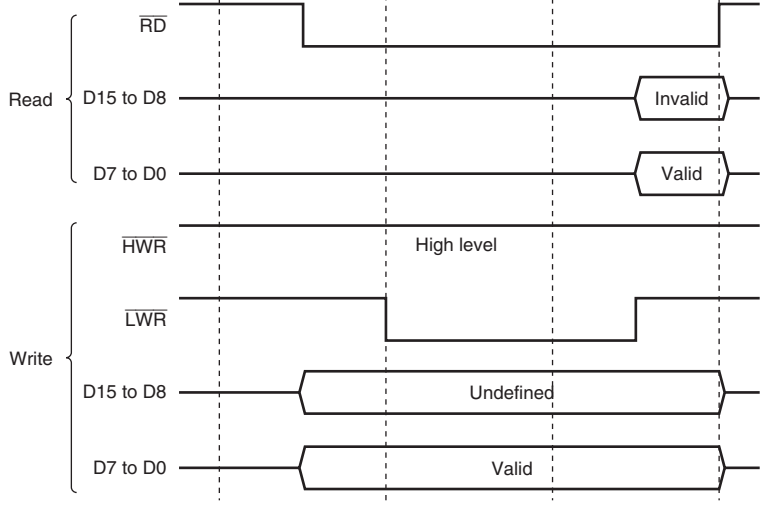
Note: * For external address space access, this signal is not output when the 256-Kbyte extended area is accessed with CS256E = 1.

Figure 6.9 Bus Timing for 16-Bit, 2-State Access Space (Word Access)



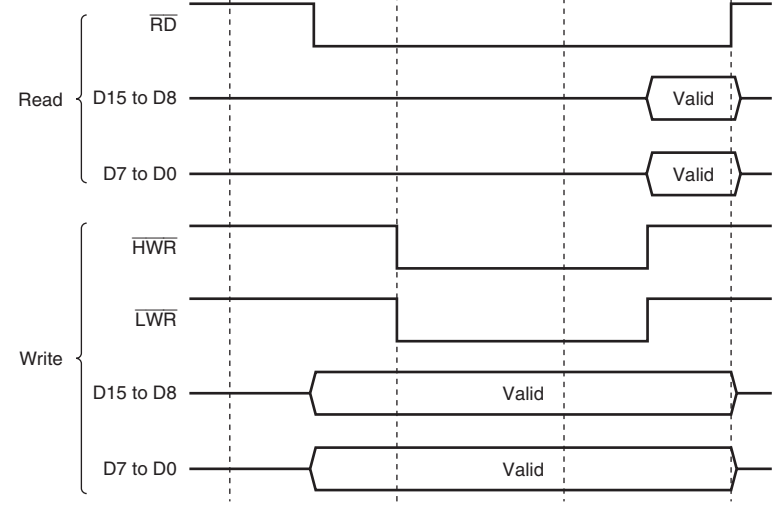
Note: * For external address space access, this signal is not output when the 256-Kbyte extended area is accessed with CS256E = 1.

Figure 6.10 Bus Timing for 16-Bit, 3-State Access Space (Even Byte Access



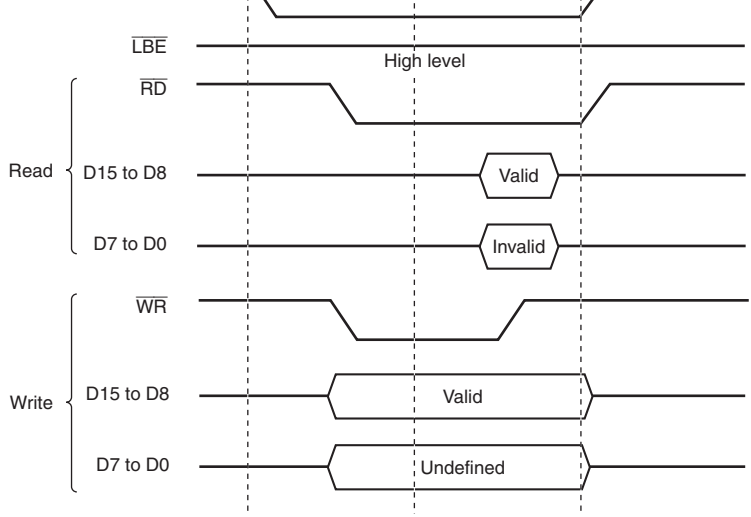
Note: * For external address space access, this signal is not output when the 256-Kbyte extended area is accessed with CS256E = 1.

Figure 6.11 Bus Timing for 16-Bit, 3-State Access Space (Odd Byte Acces



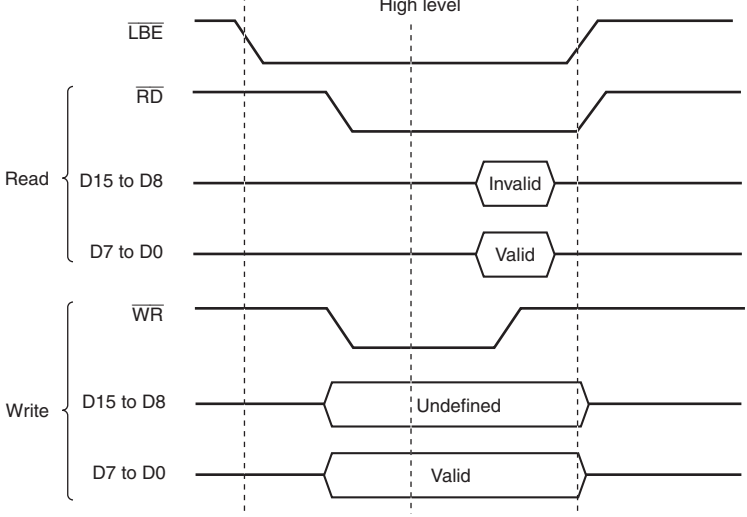
Note: * For external address space access, this signal is not output when the 256-Kbyte extended area is accessed with CS256E = 1.

Figure 6.12 Bus Timing for 16-Bit, 3-State Access Space (Word Access)



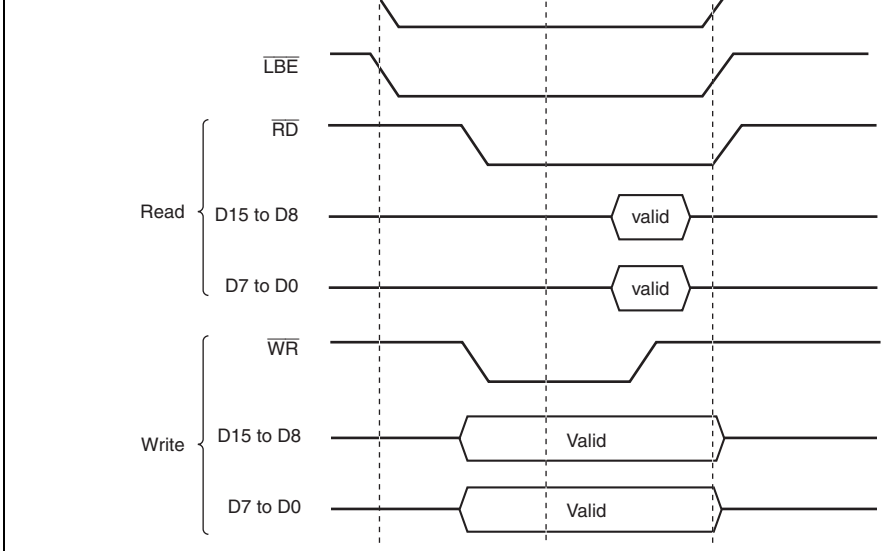
Note: * For external address space access, this signal is not output when the 256-Kbyte extended are is accessed with CS256E = 1.

Figure 6.13 Glueless Extension Even Byte Access (ADMXE = 0)



Note: * For external address space access, this signal is not output when the 256-Kbyte extended area is accessed with CS256E = 1.

Figure 6.14 Glueless Extension Odd Byte Access (ADMXE = 0)



Note: * For external address space access, this signal is not output when the 256-Kbyte extended a
is accessed with CS256E = 1.

Figure 6.15 Glueless Extension Word Access (ADMXE = 0)

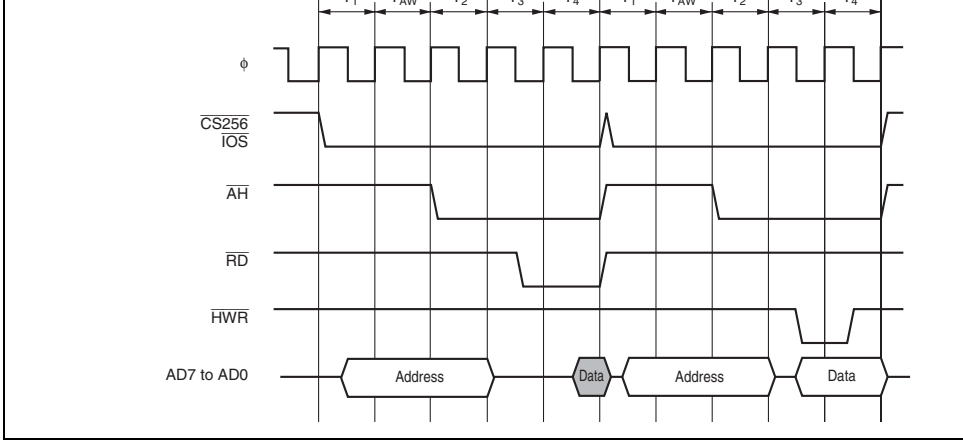


Figure 6.16 Bus Timing for 8-Bit, 2-State Access Space

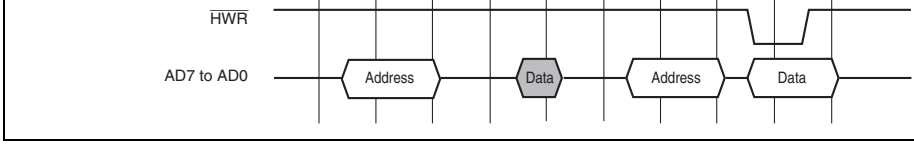


Figure 6.17 Bus Timing for 8-Bit, 2-State Access Space

(2) 8-Bit, 3-State Data Access Space

Figure 6.18 shows the bus timing for an 8-bit, 3-state access space. When an 8-bit acces
accessed, the lower half (AD7 to AD0) of the data bus is used. Wait states can be inserte

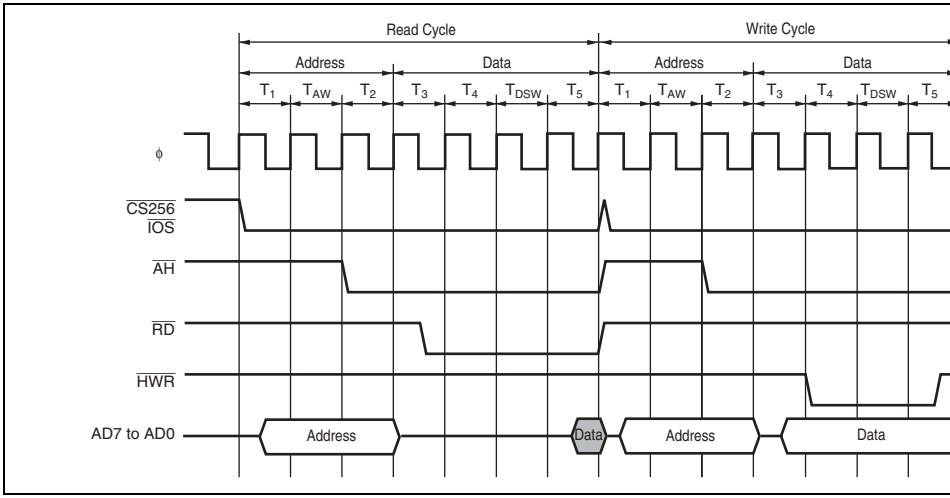


Figure 6.18 Bus Timing for 8-Bit, 3-State Access Space

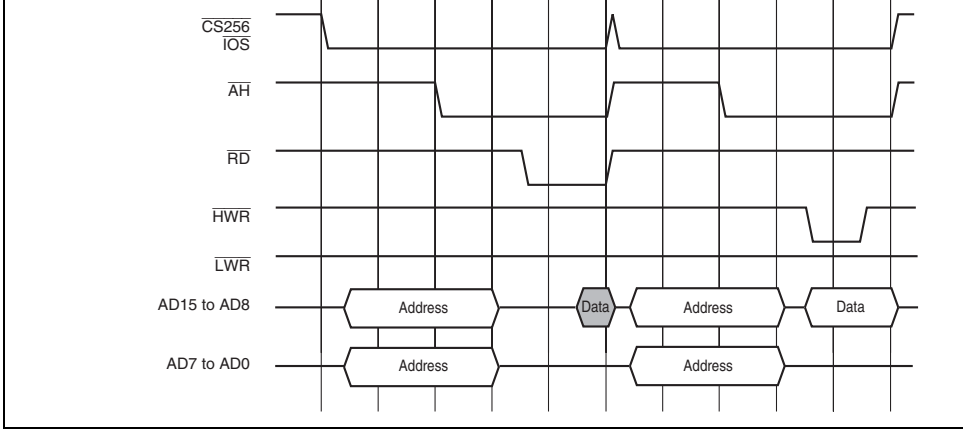


Figure 6.19 Bus Timing for 16-Bit, 2-State Access Space (1) (Even Byte Acce

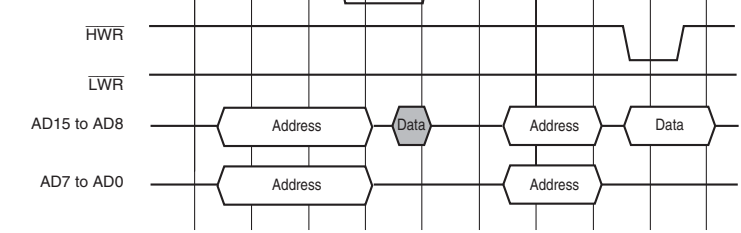


Figure 6.20 Bus Timing for 16-Bit, 2-State Access Space (2) (Even Byte Acc

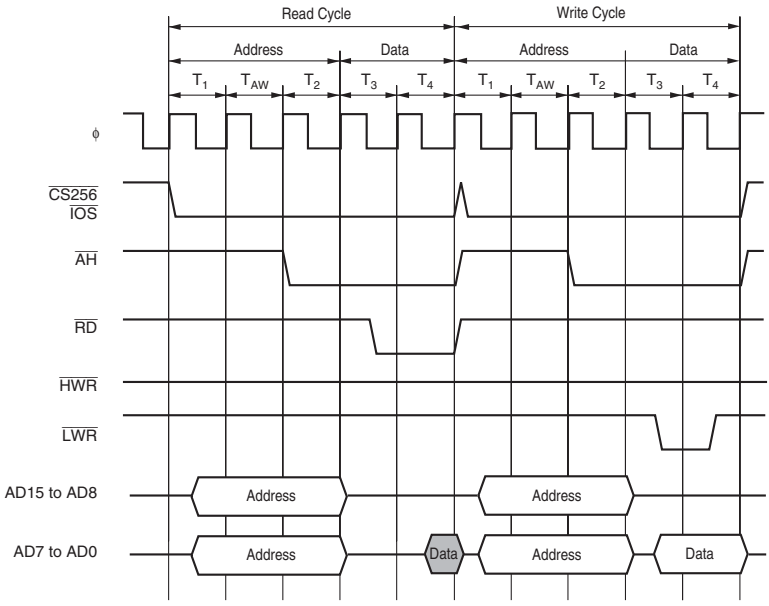


Figure 6.21 Bus Timing for 16-Bit, 2-State Access Space (3) (Odd Byte Acc

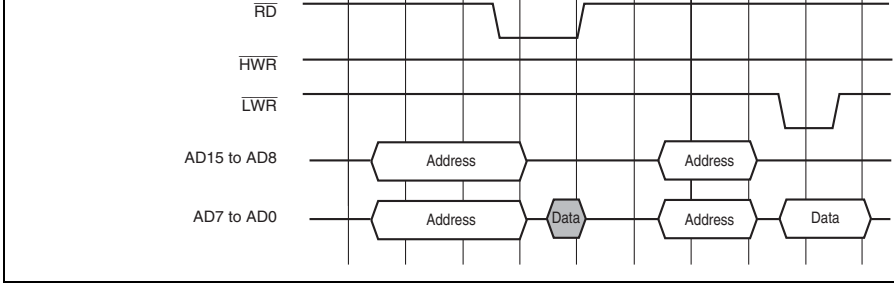


Figure 6.22 Bus Timing for 16-Bit, 2-State Access Space (4) (Odd Byte Acces

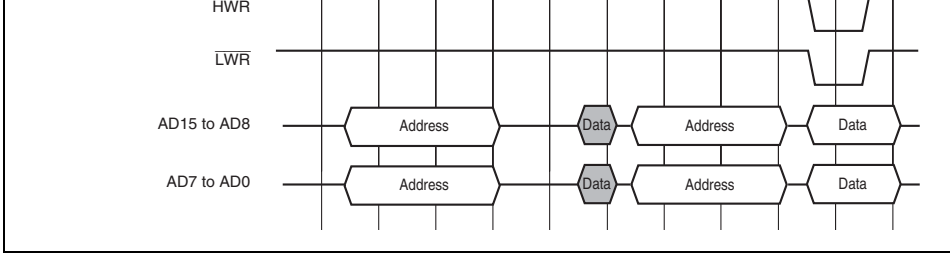


Figure 6.23 Bus Timing for 16-Bit, 2-State Access Space (5) (Word Acces

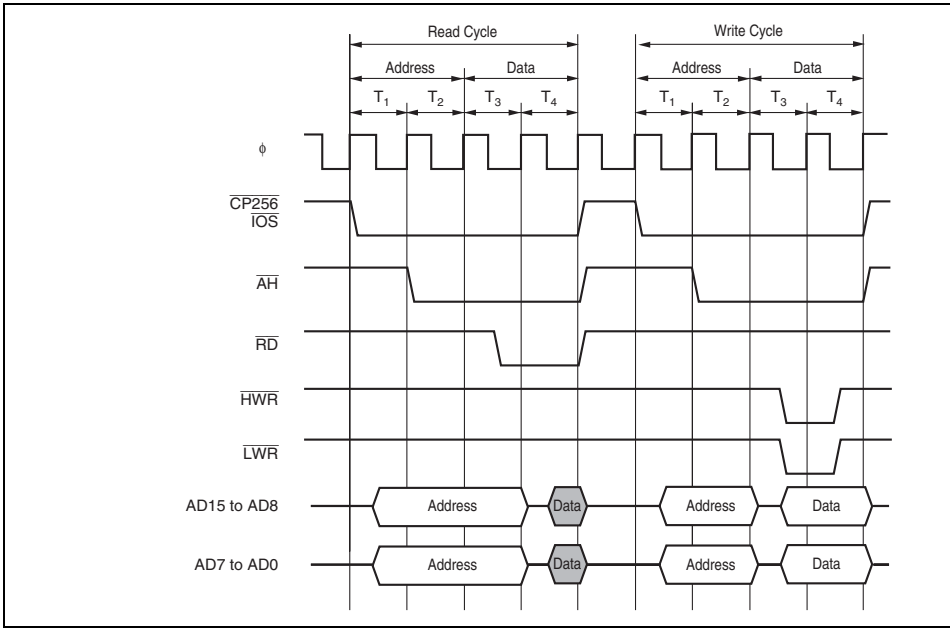


Figure 6.24 Bus Timing for 16-Bit, 2-State Access Space (6) (Word Acces

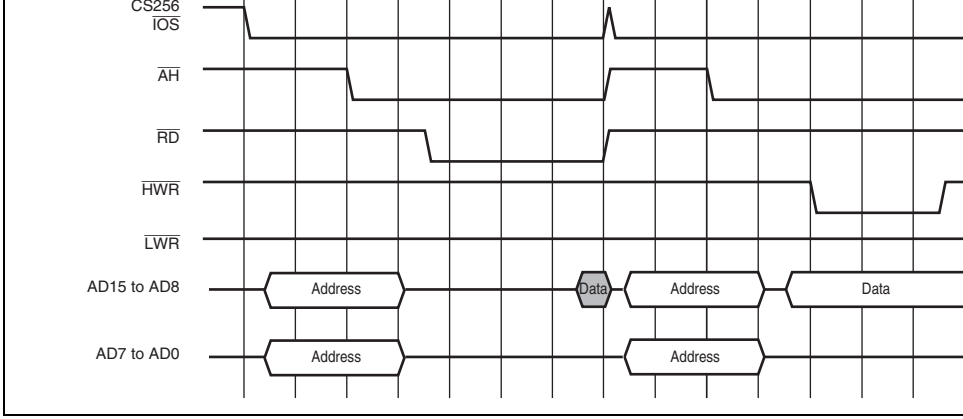


Figure 6.25 Bus Timing for 16-Bit, 3-State Access Space (1) (Even Byte Acce

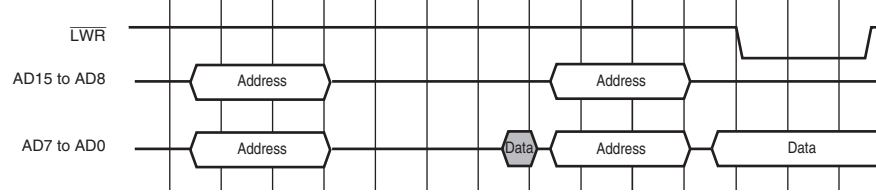


Figure 6.26 Bus Timing for 16-Bit, 3-State Access Space (2) (Odd Byte Acc

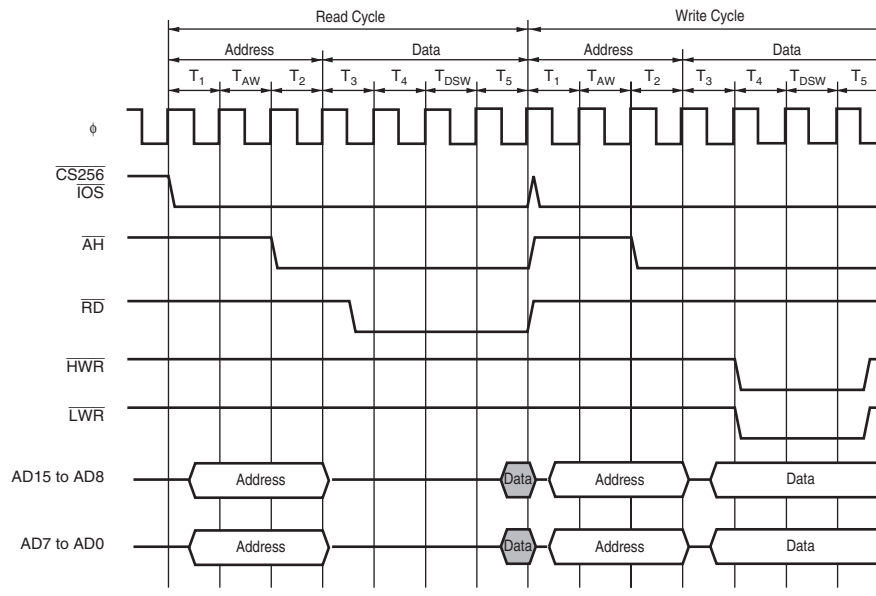


Figure 6.27 Bus Timing for 16-Bit, 3-State Access Space (3) (Word Acces

the WC1 and WC0 bits in WSCR (the WC11 and WC10 bits in WSCR2 for the 256-Kby extended area).

(b) Pin Wait Mode

A specified number of wait states T_W are always inserted between the T_2 state and T_3 state accessing the external address space. The number of wait states T_W is specified by the sett the WC1 and WC0 bits. If the $\overline{\text{WAIT}}$ pin is low at the falling edge of ϕ in the last T_2 or T_W another T_W state is inserted. If the $\overline{\text{WAIT}}$ pin is held low, T_W states are inserted until it goe

Pin wait mode is useful when inserting four or more T_W states, or when changing the num states to be inserted for each external device.

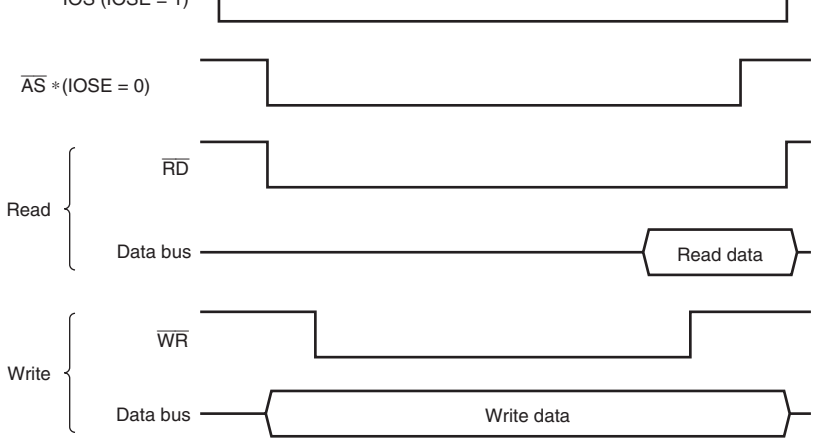
(c) Pin Auto-Wait Mode

A specified number of wait states T_W are inserted between the T_2 state and T_3 state when a the external address space if the $\overline{\text{WAIT}}$ pin is low at the falling edge of ϕ in the last T_2 sta number of wait states T_W is specified by the settings of the WC1 and WC0 bits. Even if th pin is held low, T_W states are inserted only up to the specified number of states.

Pin auto-wait mode enables the low-speed memory interface only by inputting the chip se signal to the $\overline{\text{WAIT}}$ pin.

Figure 6.28 shows an example of wait state insertion timing in pin wait mode.

The settings after a reset are: 3-state access, 3 program wait insertion, and $\overline{\text{WAIT}}$ pin inpu disabled.



Note: ↓ shown in ϕ clock indicates the WAIT pin sampling timing.

* For external address space access, this signal is not output when the 256-kbyte extended area is accessed with CS256E = 1.

Figure 6.28 Example of Wait State Insertion Timing (Pin Wait Mode)

When accessing the external address space, a specified number of wait states T_{DSW} can be between the T_4 state and T_5 state of data state. The number of wait states T_{DSW} is specified settings of the WC1 and WC0 bits. If the $\overline{WAIT}$ pin is low at the falling edge of ϕ in the l T_{DSW}, or T_{DOW} state, another T_{DOW} state is inserted. If the $\overline{WAIT}$ pin is held low, T_{DOW} states inserted until it goes high.

Pin wait mode is useful when inserting four or more T_{DOW} states, or when changing the nu T_{DOW} states to be inserted for each external device.

(c) Pin Auto-Wait Mode

A specified number of wait states T_{DOW} are inserted between the T_4 state and T_5 state when accessing the external address space if the $\overline{WAIT}$ pin is low at the falling edge of ϕ in the state. The number of wait states T_{DOW} is specified by the settings of the WC1 and WC0 bi if the $\overline{WAIT}$ pin is held low, T_{DOW} states are inserted only up to the specified number of st

Pin auto-wait mode enables the low-speed memory interface only by inputting the chip se signal to the $\overline{WAIT}$ pin.

Figure 6.29 shows an example of wait state insertion timing in pin wait mode.

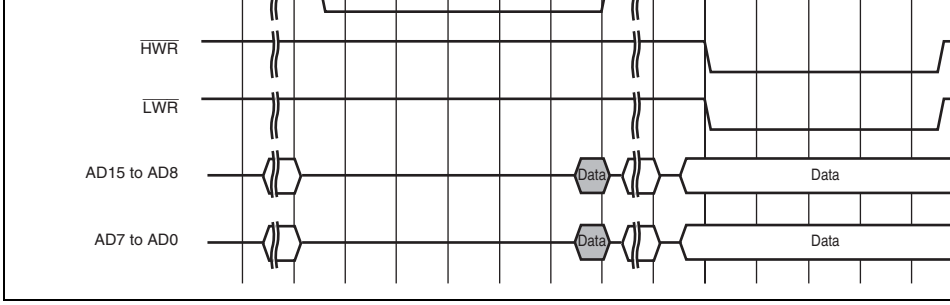


Figure 6.29 Example of Wait State Insertion Timing

determined by the AST bit in WSCR. When the AST bit is set to 1, wait states can be inse
or 2 states can be selected for burst access according to the setting of the BRSTS1 bit in B
Wait states cannot be inserted in a burst cycle. Burst accesses of a maximum four words i
performed when the BRSTS0 bit in BCR is cleared to 0, and burst accesses of a maximum
words is performed when the BRSTS0 bit in BCR is set to 1.

The basic access timing for the burst ROM space is shown in figures 6.30 and 6.31.

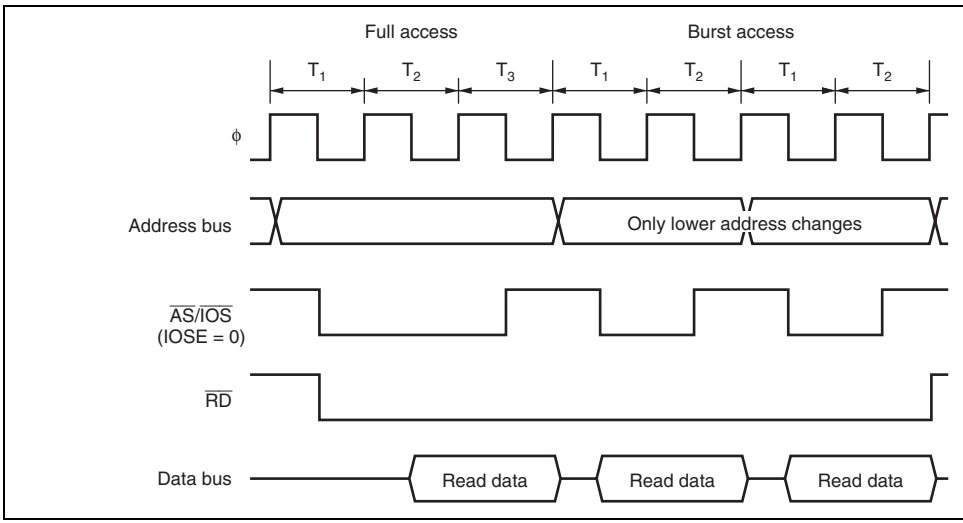


Figure 6.30 Access Timing Example in Burst ROM Space (AST = BRSTS1 =

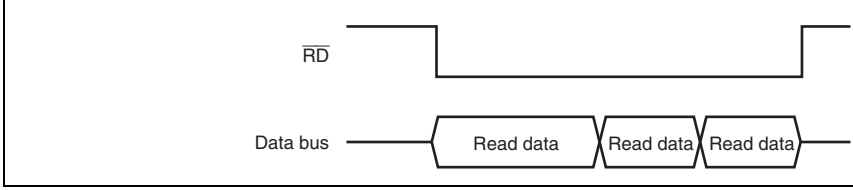


Figure 6.31 Access Timing Example in Burst ROM Space (AST = BRSTS1

6.6.2 Wait Control

As with the basic bus interface, program wait insertion or pin wait insertion using the W possible in the initial cycle (full access) of the burst ROM interface. For details, see sect Wait Control. Wait states cannot be inserted in a burst cycle.

for ROM with a long output floating time, and bus cycle B is a CPU write cycle. In figure
with no idle cycle inserted, a collision occurs in bus cycle B between the read data from R
the CPU write data. In figure 6.32 (b), an idle cycle is inserted, thus preventing data collis

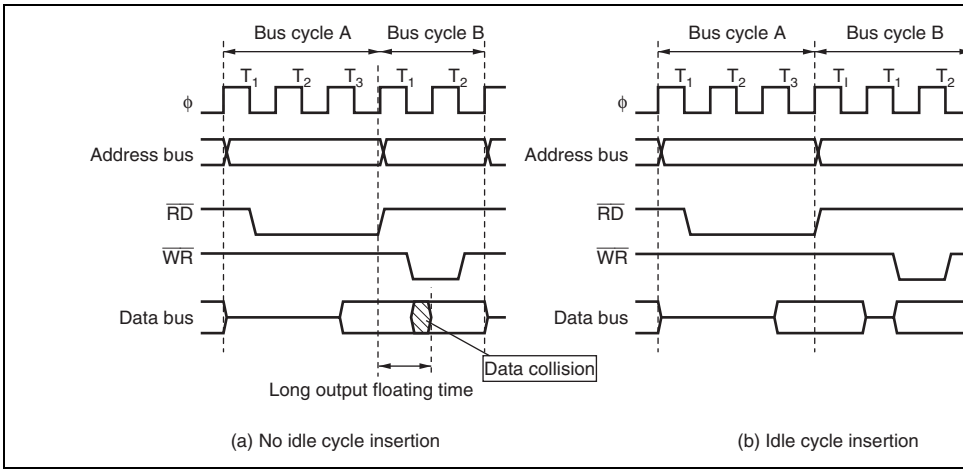


Figure 6.32 Examples of Idle Cycle Operation

6.8 Bus Arbitration

6.8.1 Overview

The BSC has a bus arbiter that arbitrates bus master operations. There are three bus mas
CPU, DTC, and E-DMAC – that perform read/write operations while they have bus mas

6.8.2 Operation

Each bus master requests the bus mastership by means of a bus mastership request signa
arbiter detects the bus mastership request signal from the bus masters, and if a bus reque
it sends a bus mastership request acknowledge signal to the bus master that made the re
designated timing. If there are bus requests from more than one bus master, the bus mas
request acknowledge signal is sent to the one with the highest priority. When a bus mast
the bus mastership request acknowledge signal, it takes the bus mastership until that sig
canceled. The order of bus master priority is as follows:

(High) E-DMAC > DTC > CPU (Low)

timing for transferring the bus mastership is as follows:

- Timing for transferring the bus mastership to the DTC

1. Bus mastership is transferred at a break between bus cycles. However, if bus cycle is
in discrete operations, as in the case of a longword size access, the bus is not transfer
break between the operations. For details, see section 2.7, Bus States During Instructi
Execution in the H8S/2600 Series, H8S/2000 Series Software Manual.
2. If the CPU is in sleep mode, it transfers the bus mastership immediately.

- Timing for transferring the bus mastership to the E-DMAC

1. Bus mastership is transferred at a break between bus cycles. Even if bus cycle is exec
discrete operations, as in the case of a longword size access, the bus can be transferre
break between bus cycles. For details, see section 21, Ethernet Controller Direct Mem
Access Controller (E-DMAC).
2. If the CPU is in sleep mode, it transfers the bus mastership immediately.

in discrete operations, as in the case of a longword size access, the bus is not transfer
break between the operations. In addition, in the case of a 32-bit access by the DTC,
not transferred at a break between the operations. For details, see section 21, Etherne
Controller Direct Memory Access Controller (E-DMAC).

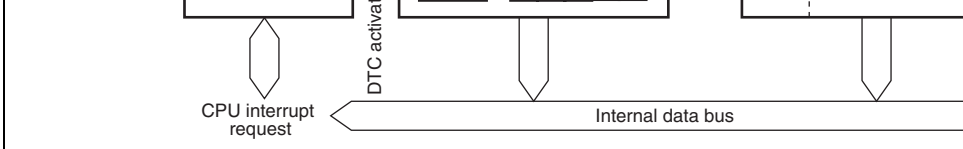
2. If the CPU is in sleep mode, it transfers the bus mastership immediately.

(3) E-DMAC

The E-DMAC is the highest-priority bus master, and sends the bus arbiter a request for t
when an activation request is generated. The E-DMAC does not release the bus until the
consecutive transfer cycles have completed. For details, see section 21, Ethernet Contro
Memory Access Controller (E-DMAC).

7.1 Features

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



[Legend]

MRA, MRB:	DTC mode register A, B
CRA, CRB:	DTC transfer count register A, B
SAR:	DTC source address register
DAR:	DTC destination address register
DTCERA to DTCERF:	DTC enable registers A to F
DTVECR:	DTC vector register

Figure 7.1 Block Diagram of DTC

These six registers cannot be directly accessed from the CPU. When a DTC activation in
source occurs, the DTC reads a set of register information that is stored in on-chip RAM
corresponding DTC registers and transfers data. After the data transfer, it writes a set of
register information back to on-chip RAM.

- DTC enable registers (DTCER)
- DTC vector register (DTVECR)
- Keyboard comparator control register (KBCOMP)
- Event counter control register (ECCR)
- Event counter status register (ECS)

				(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			These bits specify a DAR operation after a dat transfer.
				0*: 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
2	MD0			These bits specify the DTC transfer mode.
				00: Normal mode
				01: Repeat mode
				10: Block transfer mode
				11: Setting prohibited
1	DTS	Undefined	—	DTC Transfer Mode Select
				Specifies whether the source side or the destin side is set to be a repeat area or block area in mode or block transfer mode.
				0: Destination side is repeat area or block area
				1: Source side is repeat area or block area
0	Sz	Undefined	—	DTC Data Transfer Size
				Specifies the size of data to be transferred.
				0: Byte-size transfer
				1: Word-size transfer

Note: * Don't care

				the end of the specified number of data transf clearing of the interrupt source flag, and clear DTCER are not performed.
6	DISEL	Undefined	—	DTC Interrupt Select When this bit is set to 1, a CPU interrupt requ generated every time data transfer ends. Whe is cleared to 0, a CPU interrupt request is ger only when the specified number of data transf
5 to 0	—	Undefined	—	Reserved These bits have no effect on DTC operation. T value should always be 0.

7.2.3 DTC Source Address Register (SAR)

SAR is a 24-bit register that designates the source address of data to be transferred by th For word-size transfer, specify an even source address.

7.2.4 DTC Destination Address Register (DAR)

DAR is a 24-bit register that designates the destination address of data to be transferred DTC. For word-size transfer, specify an even destination address.

7.2.6 DTC Transfer Count Register B (CRB)

CRB is a 16-bit register that designates the number of times data is to be transferred by th
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.

7.2.7 DTC Enable Registers (DTCER)

DTCER specifies DTC activation interrupt sources. DTCER is comprised of five register
DTCERA to DTCERF. The correspondence between interrupt sources and DTCE bits is
tables 7.1 and 7.4. For DTCE bit setting, use bit manipulation instructions such as BSET
BCLR. Multiple DTC activation sources can be set at one time (only at the initial setting)
masking all interrupts and writing data after executing a dummy read on the relevant regi

Bit	Bit Name	Initial Value	R/W	Description
7 to 0	DTCE7 to DTCE0	All 0	R/W	DTC Activation Enable Setting this bit to 1 specifies a relevant interrup as a DTC activation source. [Clearing conditions] • When data transfer has ended with the DIS MRB set to 1 • When the specified number of transfers ha These bits are not cleared when the DISEL bit the specified number of transfers have not bee completed

1	DTCEn1	—	—	(82)TXI3	—	(106)IBFI2	—
0	DTCEn0	—	—	(85)RXI1	—	(107)IBFI3	—

[Legend]

n: A to F

( ): Vector number

—: Reserved. The write value should always be 0.

*: Only in the H8S/2472

7.2.8 DTC Vector Register (DTVECR)

DTVECR enables or disables DTC activation by software, and sets a vector number for software activation interrupt.

Bit	Bit Name	Initial Value	R/W	Description
7	SWDTE	0	R/W	DTC Software Activation Enable Setting this bit to 1 activates DTC. Only 1 can be this bit. [Clearing conditions] • When the DISEL bit is 0 and the specified nu transfers have not ended • When 0 is written to the DISEL bit after a sof activated data transfer end interrupt (SWDTI request has been sent to the CPU. This bit will not be cleared when the DISEL bit is data transfer has ended or when the specified n transfers has ended.

7.2.9 Keyboard Comparator Control Register (KBCOMP)

KBCOMP enables or disables the comparator scan function of event counter.

Bit	Bit Name	Initial Value	R/W	Description
7	EVENTE	0	R/W	Event Count Enable 0: Disables event count function 1: Enables event count function
6, 5	—	All 0	R	Reserved These bits are always read as 0 and cannot be m
4 to 0	—	All 0	R/W	Reserved The initial value should not be changed.

6 to 4	—	All 0	R	Reserved These bits are always read as 0 and cannot be
3 to 0	ECSB3 to ECSB0	All 0	R/W	Event Counter Channel Select 3 to 0 These bits select pins for event counter input. A of pins are selected starting from EVENT0. Whe PAnDDR is set to 1, inputting events to EVENT EVENT7 is ignored. 0000: EVENT0 is used 0001: EVENT0 to EVENT1 are used 0010: EVENT0 to EVENT2 are used 0011: EVENT0 to EVENT3 are used 0100: EVENT0 to EVENT4 are used 0101: EVENT0 to EVENT5 are used 0110: EVENT0 to EVENT6 are used 0111: EVENT0 to EVENT7 are used 1000: EVENT0 to EVENT8 are used 1001: EVENT0 to EVENT9 are used 1010: EVENT0 to EVENT10 are used 1011: EVENT0 to EVENT11 are used 1100: EVENT0 to EVENT12 are used 1101: EVENT0 to EVENT13 are used 1110: EVENT0 to EVENT14 are used 1111: EVENT0 to EVENT15 are used

0: The corresponding event is already processed

1: The corresponding event is not yet processed

	1	DTS	0: Destination is repeat area
	0	Sz	1: Word size transfer
MRB	7	CHNE	0: Chain transfer is disabled
	6	DISEL	0: Interrupt request is generated when data is transfe the number of specified times
	5 to 0	—	B'000000
SAR	23 to 0	—	Identical optional RAM address. Its lower five bits are
DAR	23 to 0	—	The start address of 16 words is this address. They a incremented every time an event is detected in EVEI EVENT15.
CRAH	7 to 0	—	H'FF
CRAL	7 to 0	—	H'FF
CRBH	7 to 0	—	H'FF
CRBL	7 to 0	—	H'FF
DTCERC	4	DTCEC4	1: DTC function of the event counter is enabled
KBCOMP	7	EVENTE	1: Event counter enable
RAM	—	—	(SAR, DAR) : Result of EVENT0 count (SAR, DAR) + 2: Result of EVENT 1 count (SAR, DAR) + 4: Result of EVENT 2 count ↓ (SAR, DAR) + 30: Result of EVENT 15 count

The corresponding flag to ECS input pin is set to 1 when the event pins that are specifie
ECSB3 to ECSB0 in ECCR detect the edge events specified by the EDSB in ECCR. For
state, status/address codes are generated.

An EVENTI interrupt request is generated even if only one bit in ECS is set to 1.

													1	0	
												1	0	0	
											1	0	0	0	
										1	0	0	0	0	
									1	0	0	0	0	0	
								1	0	0	0	0	0	0	
							1	0	0	0	0	0	0	0	
						1	0	0	0	0	0	0	0	0	
					1	0	0	0	0	0	0	0	0	0	
				1	0	0	0	0	0	0	0	0	0	0	
			1	0	0	0	0	0	0	0	0	0	0	0	
		1	0	0	0	0	0	0	0	0	0	0	0	0	
	1	0	0	0	0	0	0	0	0	0	0	0	0	0	
1	0	0	0	0	0	0	0	0	0	0	0	0	0	0	

1. Continuous events that are input from the same pin and out of DTC handling are ig
because the count up is operated by means of the DTC.
2. If some events are generated in short intervals, the priority of event counter handlin
ordered and events are not handled in order of arrival.
3. If the counter overflows, this event counter counts from H'0000 without generating
interrupt.

diagram of DTC activation source control. For details on the interrupt controller, see sect
Interrupt Controller.

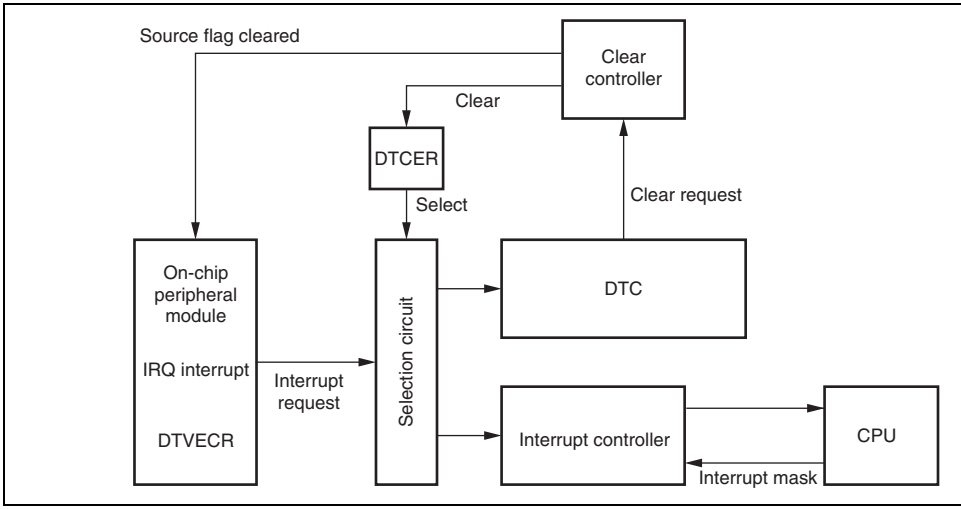


Figure 7.2 Block Diagram of DTC Activation Source Control

then reads the register 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 a 2-byte unit. Specify the lower two bytes of t information start address.

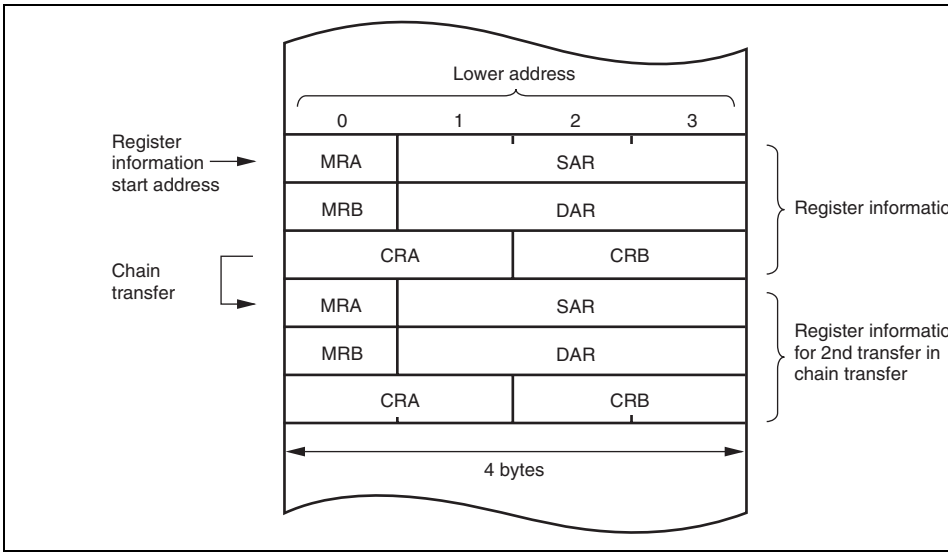


Figure 7.3 DTC Register Information Location in Address Space

EVC	EVENTI	29	H'043A	DTCEC4
IIC_2	IICI2	76	H'0498	DTCEB6
IIC_3	IICI3	78	H'049C	DTCED4
SCI_3	RXI3	81	H'04A2	DTCEC2
	TXI3	82	H'04A4	DTCEC1
SCI_1	RXI1	85	H'04AA	DTCEC0
	TXI1	86	H'04AC	DTCED7
SSU	RXIS	89	H'04B2	DTCED6
	TXIS	90	H'04B4	DTCED5
IIC_0	IICI0	94	H'04BC	DTCEB5
IIC_1	IICI1	98	H'04C4	DTCED3
LPC	ERRI	104	H'04D0	DTCEE3
	IBFI1	105	H'04D2	DTCEE2
	IBFI2	106	H'04D4	DTCEE1
	IBFI3	107	H'04D6	DTCEE0
USB (only in the H8S/2472)	USBINT0	115	H'04E6	DTCEF7
	USBINT1	118	H'04EC	DTCEF6

Note: * DTCE bits with no corresponding interrupt are reserved, and the write value sh
always be 0.

decremented, or left fixed depending on its register information.

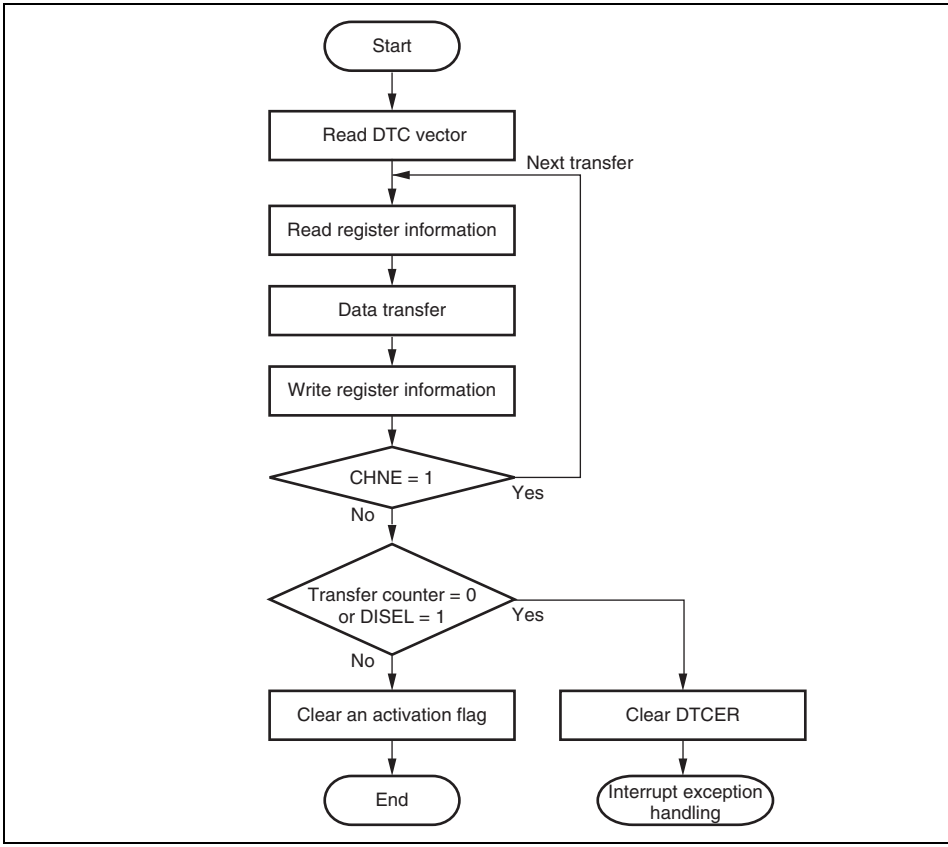


Figure 7.4 DTC Operation Flowchart

DTC transfer count register A	CRA	Transfer counter
DTC transfer count register B	CRB	Not used

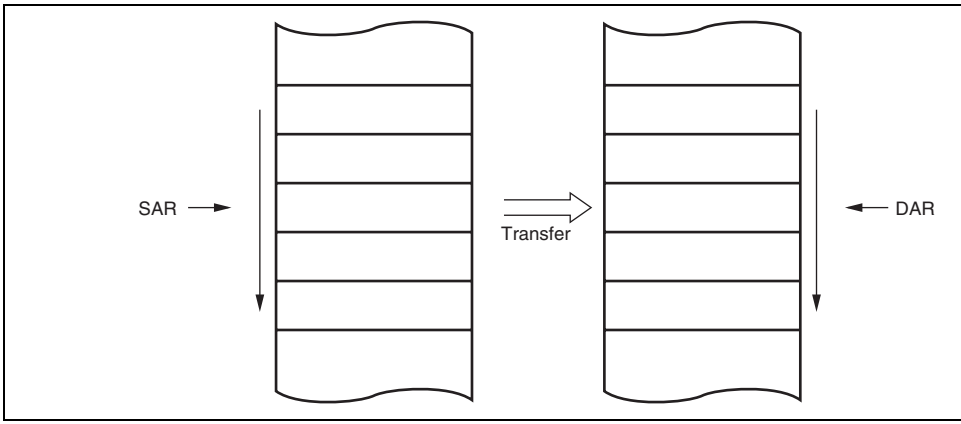


Figure 7.5 Memory Mapping in Normal Mode

Name	Abbreviation	Function
DTC source address register	SAR	Transfer source address
DTC destination address register	DAR	Transfer destination addres
DTC transfer count register AH	CRAH	Holds number of transfers
DTC transfer count register AL	CRAL	Transfer Count
DTC transfer count register B	CRB	Not used

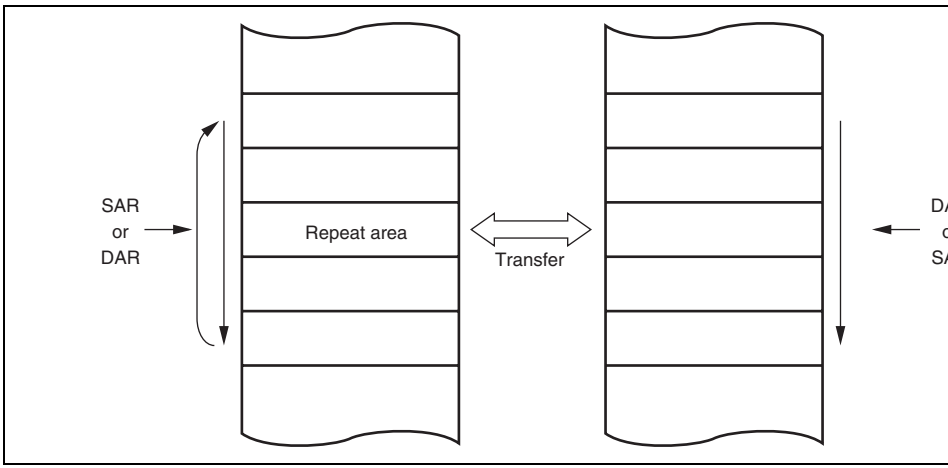


Figure 7.6 Memory Mapping in Repeat Mode

Name	Abbreviation	Function
DTC source address register	SAR	Transfer source address
DTC destination address register	DAR	Transfer destination address
DTC transfer count register AH	CRAH	Holds block size
DTC transfer count register AL	CRAL	Block size counter
DTC transfer count register B	CRB	Transfer counter

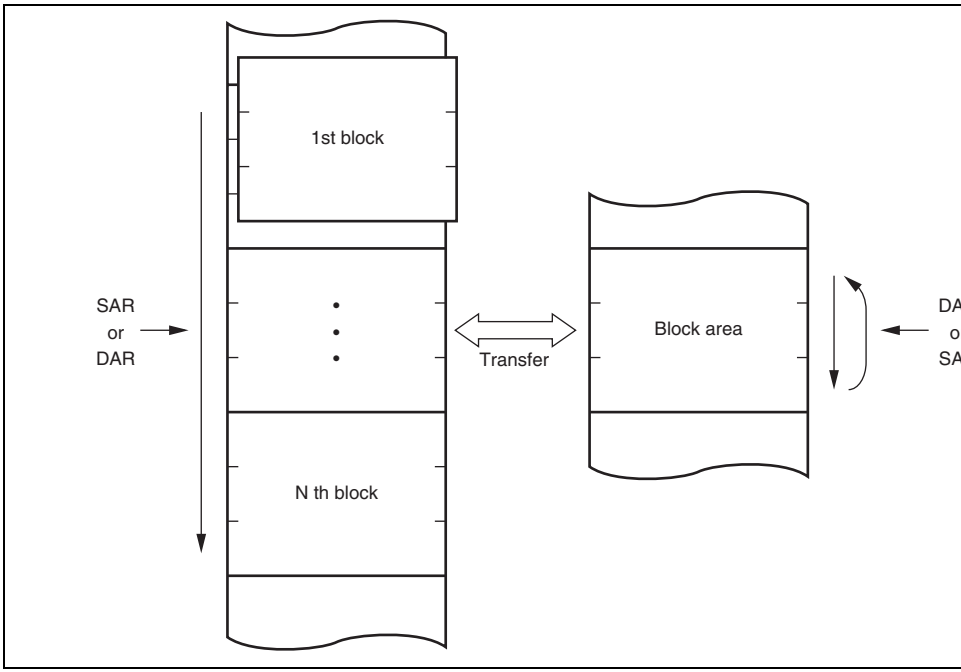


Figure 7.7 Memory Mapping in Block Transfer Mode

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

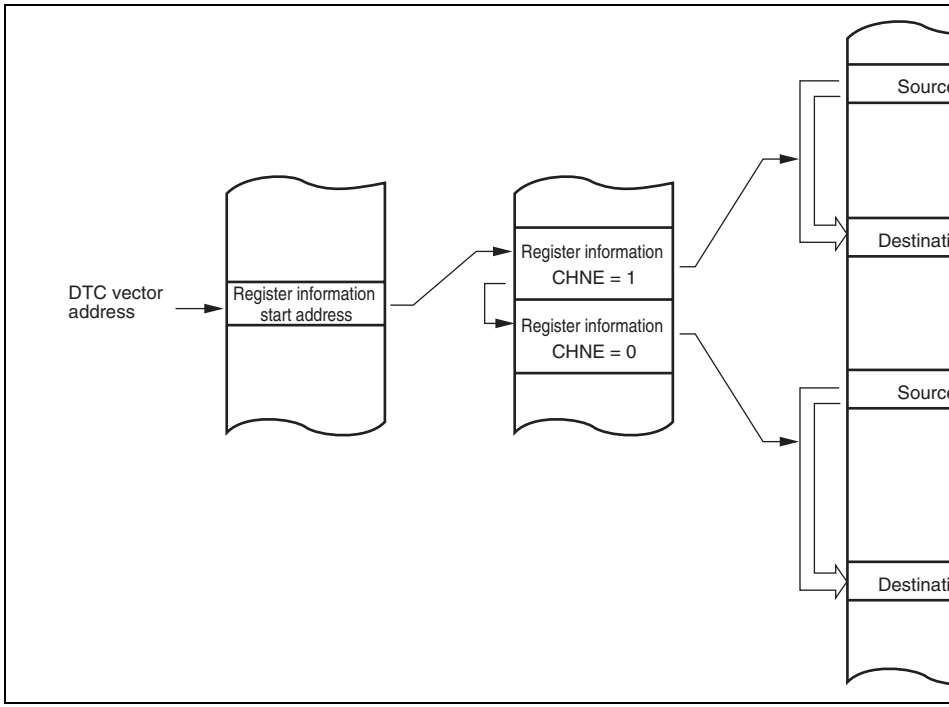


Figure 7.8 Chain Transfer Operation

transfers have been completed, after data transfer ends, the SWDTE bit is held at 1 and an
SWDTEND interrupt is generated. The interrupt handling routine will then clear the SWI
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.

7.6.6 Operation Timing

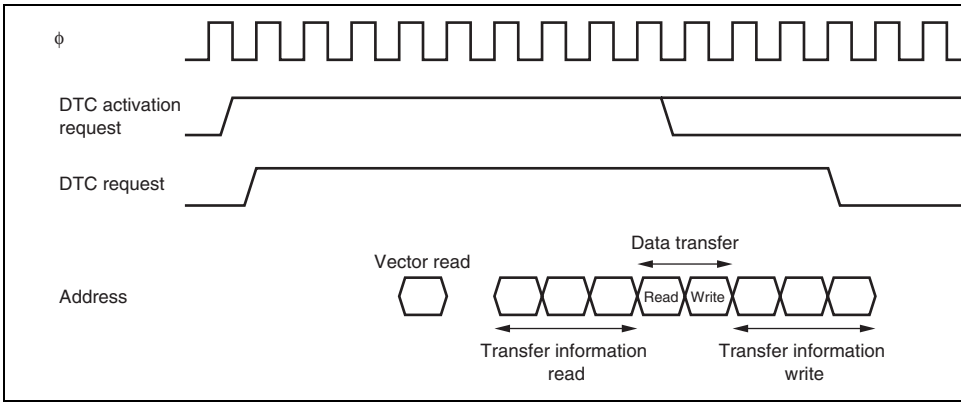


Figure 7.9 DTC Operation Timing (Example in Normal Mode or Repeat Mo

Figure 7.10 DTC Operation Timing (Example of Block Transfer Mode, with Block Size of 2)

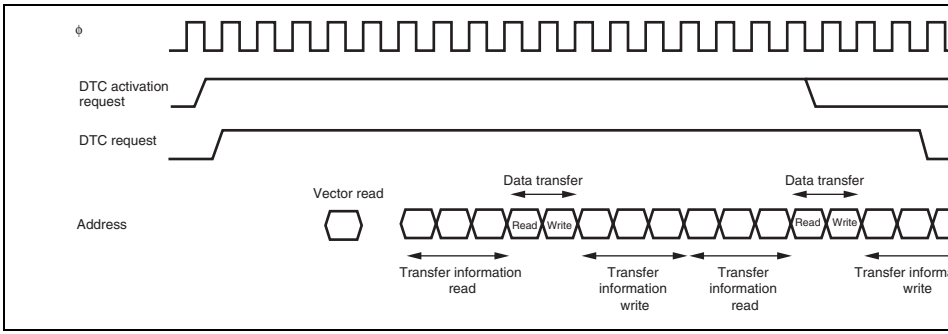


Figure 7.11 DTC Operation Timing (Example of Chain Transfer)

Normal	1	6	1	1	3
Repeat	1	6	1	1	3
Block transfer	1	6	N	N	3

[Legend]

N: Block size (initial setting of CRAH and CRAL)

Table 7.9 Number of States Required for Each Execution Status

Object to be Accessed		On-Chip RAM (H'FFEC00 to H'FFEFFF)	On-Chip RAM (On-chip RAM area other than H'FFEC00 to H'FFEFFF)	On-Chip ROM	On-Chip I/O Registers		External Devices		
Bus width		32	16	16	8	16	8	8	16
Access states		1	1	1	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	1	2	2	2	3 + m	2
	Word data read S_K	1	1	1	4	2	4	6 + 2m	2
	Byte data write S_L	1	1	1	2	2	2	3 + m	2
	Word data write S_L	1	1	1	4	2	4	6 + 2m	2
	Internal operation S_M	1	1	1	1	1	1	1	1

7.7.1 Activation by Interrupt

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

1. Set the MRA, MRB, SAR, DAR, CRA, and CRB register information in on-chip RA
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. T 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 tran been completed, the DTCE bit is cleared to 0 and a CPU interrupt is requested. If the continue transferring data, set the DTCE bit to 1.

7.7.2 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 RA
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 the 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 interrup requested, the SWDTE bit is cleared to 0. If the DTC is to continue transferring data SWDTE bit to 1. When the DISEL bit is 1 or after the specified number of data trans been completed, the SWDTE bit is held at 1 and a CPU interrupt is requested.

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

2. Set the start address of the register information at 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 a receive error interrupts.
5. Each time the reception of one byte of data has been completed on the SCI, the RDRF SSR is set to 1, an RXI interrupt is generated, and the DTC is activated. The receive d 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 128 data transfers have been completed, the RDRF flag i 1, the DTCE bit is cleared to 0, and an RXI interrupt request is sent to the CPU. The i handling routine will perform wrap-up processing.

2. Set the start address of the register information at the DTC vector address (H'04C0).
3. Check that the SWDTE bit in DTVECR is 0. Check that there is currently no transfe by software.
4. Write 1 to the SWDTE bit and the vector number (H'60) to DTVECR. The write dat
5. Read DTVECR again and check that it is set to the vector number (H'60). If it is not, indicates that the write failed. This is presumably because an interrupt occurred betw 3 and 4 and led to a different software activation. To activate this transfer, go back to
6. If the write was successful, the DTC is activated and a block of 128 bytes of data is transferred.
7. After the transfer, an SWDTEND interrupt occurs. The interrupt handling routine sh the SWDTE bit to 0 and perform wrap-up processing.

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

7.9.3 DTCE Bit Setting

For DTCE bit setting, use bit manipulation instructions such as BSET and BCLR, for rea
writing. Multiple DTC activation sources can be set at one time (only at the initial setting
masking all interrupts and writing data after executing a dummy read on the relevant regi

7.9.4 DTC Activation by Interrupt Sources of SCI, IIC, or A/D Converter

Interrupt sources of the SCI, IIC, or A/D converter which activate the DTC are cleared wi
reads from or writes to the respective registers, and they cannot be cleared by the DISEL

port A pins and D0 to D5 pins, the on/off status of the input pull-up MOS is controlled b
respective DDR and the output data register (ODR). Ports 1 to 4, and 6 have an input pu
control register (PCR), in addition to DDR and DR, to control the on/off status of the inp
MOSs.

Port 3 pins and pins 47 to 44 and B3 to B0 have built-in de-bouncers (DBn) that elimina
in the input signals.

Ports 4 and F are designed for retain state outputs (RSn), which retain the output values
pins even if a reset is generated when the watchdog timer has overflowed.

Ports 1 to 6, and 8 to E can drive a single TTL load and 30 pF capacitive load. All the I/
can drive a Darlington transistor in output mode. Port pins 80 to 83, C0 to C5, D6, and D
NMOS push-pull output.

Port 2	General I/O port multiplexed with SCIF control I/O	P27/$\overline{\text{DTR}}$ P26/$\overline{\text{DSR}}$ P25/$\overline{\text{RI}}$ P24/$\overline{\text{DCD}}$	Same as left	Buil pull-
	General I/O port multiplexed with address output and address-data multiplex I/O	P23 P22 P21 P20	P23/A11/AD11 P22/A10/AD10 P21/A9/AD9 P20/A8/AD8	
Port 3	General I/O port multiplexed with de-bounced input and bidirectional data bus I/O	P37/ExDB7 P36/ExDB6 P35/ExDB5 P34/ExDB4 P33/ExDB3 P32/ExDB2 P31/ExDB1 P30/ExDB0	P37/ExDB7/D15 P36/ExDB6/D14 P35/ExDB5/D13 P34/ExDB4/D12 P33/ExDB3/D11 P32/ExDB2/D10 P31/ExDB1/D9 P30/ExDB0/D8	Buil pull-
Port 4	General I/O port multiplexed with interrupt input, de-bounced input, address output, and address-data multiplex I/O	P47/$\overline{\text{IRQ7}}$/RS7/DB7/HC7 P46/$\overline{\text{IRQ6}}$/RS6/DB6/HC6 P45/$\overline{\text{IRQ5}}$/RS5/DB5/HC5 P44/$\overline{\text{IRQ4}}$/RS4/DB4/HC4	P47/A15/AD15 P46/A14/AD14 P45/A13/AD13 P44/A12/AD12	Buil pull- LED cap (sin 12 r
	General I/O port multiplexed with interrupt input and bidirectional data bus* I/O	P43/$\overline{\text{IRQ3}}$/RS3/HC3 P42/$\overline{\text{IRQ2}}$/RS2/HC2 P41/$\overline{\text{IRQ1}}$/RS1/HC1 P40/$\overline{\text{IRQ0}}$/RS0/HC0	P43/$\overline{\text{IRQ3}}$/RS3/HC3/D7* P42/$\overline{\text{IRQ2}}$/RS2/HC2/D6* P41/$\overline{\text{IRQ1}}$/RS1/HC1/D5* P40/$\overline{\text{IRQ0}}$/RS0/HC0/D4*	

	and SCI_1 I/O	P50/IRQ8/TxDF		
Port 6	General I/O port multiplexed with interrupt input, SCIF control I/O and SSU control I/O	P67/ExIRQ8/SSCK P66/ExIRQ9/SCS P65/ExIRQ10/RTS P64/ExIRQ11/CTS	Same as left	Bu pu
	General I/O port multiplexed with interrupt input, PWMX output, and bidirectional data bus* I/O	P63/PWX3 P62/PWX2 P61/IRQ15/PWX1 P60/IRQ14/PWX0	P63/PWX3/D3* P62/PWX2/D2* P61/IRQ15/PWX1/D1* P60/IRQ14/PWX0/D0*	
Port 7	General input port multiplexed with A/D converter analog input	P77/AN7 P76/AN6 P75/AN5 P74/AN4 P73/AN3 P72/AN2 P71/AN1 P70/AN0	Same as left	
Port 8	General I/O port multiplexed with interrupt input, A/D converter external trigger input, and SCI_1 and SCI_3 I/O	P87/ExIRQ15/TxD3/ADTRG P86/ExIRQ14/RxD3 P85/ExIRQ13/SCK1 P84/ExIRQ12/SCK3	Same as left	
	General I/O port multiplexed with IIC_0 and IIC_1 I/O	P83/SDA1 P82/SCL1 P81/SDA0 P80/SCL0	Same as left	NM pu ou

Port A	General I/O port multiplexed with interrupt input, DTC event counter input, EtherC control I/O, and address output	PA7/ExIRQ7/EVENT7/EXOUT PA6/$\overline{\text{ExIRQ6}}$/EVENT6/LNKSTA PA5/$\overline{\text{ExIRQ5}}$/EVENT5/WOL PA4/$\overline{\text{ExIRQ4}}$/EVENT4 PA3/$\overline{\text{ExIRQ3}}$/EVENT3 PA2/$\overline{\text{ExIRQ2}}$/EVENT2 PA1/$\overline{\text{ExIRQ1}}$/EVENT1 PA0/$\overline{\text{ExIRQ0}}$/EVENT0	PA7/ExIRQ7/EVENT7/A23 PA6/$\overline{\text{ExIRQ6}}$/EVENT6/A22/LNKSTA PA5/$\overline{\text{ExIRQ5}}$/EVENT5/A21/WOL PA4/$\overline{\text{ExIRQ4}}$/EVENT4/A20 PA3/$\overline{\text{ExIRQ3}}$/EVENT3/A19 PA2/$\overline{\text{ExIRQ2}}$/EVENT2/A18 PA1/$\overline{\text{ExIRQ1}}$/EVENT1/A17 PA0/$\overline{\text{ExIRQ0}}$/EVENT0/A16	Bui pull
Port B	General I/O port multiplexed with DTC event counter input and EtherC control I/O	PB7/EVENT15/RM_RX-ER PB6/EVENT14/RM_CRS-DV PB5/EVENT13/ RM_REF-CLK PB4/EVENT12/RM_TX-EN	Same as left	
	General I/O port multiplexed with de-bounced input, DTC event counter input, and EtherC control I/O	PB3/DB3/EVENT11/RM_RXD1 PB2/DB2/EVENT10/RM_RXD0 PB1/DB1/EVENT9/RM_TXD1 PB0/DB0/EVENT8/RM_TXD0	Same as left	
Port C	General I/O port multiplexed with bus control output	PC7 PC6	$\overline{\text{RD}}$ PC6/$\overline{\text{LWR}}$	
	General I/O port multiplexed with IIC_2 to IIC_4 I/O	PC5/SDA4 PC4/SCL4 PC3/SDA3 PC2/SCL3 PC1/SDA2 PC0/SCL2	Same as left	NM pus outp

		PE7/SERIRQ	Same as left
	multiplexed with LPC I/O	PE6/LCLK PE5/$\overline{\text{LRESET}}$ PE4/$\overline{\text{LFRAME}}$ PE3/LAD3 PE2/LAD2 PE1/LAD1 PE0/LAD0	
Port F	General I/O port multiplexed with PWMX output and EtherC control I/O	PF6/ExPWX2/RS14 PF5/RS13 PF4/RS12 PF3/ExPWX3/RS11 PF2/RS10 PF1/RS9/MDC PF0/RS8/MDIO	Same as left

Note: * Available when configured for 16-bit data bus.

The individual bits of P1DDR specify input or output for the pins of port 1.

Bit	Bit Name	Initial Value	R/W	Description
7	P17DDR	0	W	• Normal extended mode (ADMXE = 0) When set to 1, the corresponding pins fur address output pins; when cleared to 0, fu as input port pins. • Address-data multiplex extended mode (A 1) These bits correspond to the AD7 to AD0 the address-data multiplex bus. • Single-chip mode When set to 1, the corresponding pins fur output port pins; when cleared to 0, functi input port pins.
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	

(3) Port 1 Pull-Up MOS Control Register (P1PCR)

P1PCR controls the port 1 built-in input pull-up MOSs.

Bit	Bit Name	Initial Value	R/W	Description
7	P17PCR	0	R/W	When the pins are in the input state, the corresponding input pull-up MOS is turned a P1PCR bit is set to 1. Do not change the initial value when using address-data multiplex extended bus mod
6	P16PCR	0	R/W	
5	P15PCR	0	R/W	
4	P14PCR	0	R/W	
3	P13PCR	0	R/W	
2	P12PCR	0	R/W	
1	P11PCR	0	R/W	
0	P10PCR	0	R/W	

Pin function	P1n input pin	ADn input/output pin	P1n input pin	An output pin	Setting prohibited	P1

[Legend] n = 7 to 0, X: Don't care.

(b) Single-Chip Mode (EXPE = 0)

The pin function is switched as shown below according to the P1nDDR bit.

P1nDDR	0	1
Pin function	P1n input pin	P1n output pin

[Legend] n = 7 to 0

(5) Port 1 Input Pull-Up MOS

Port 1 has built-in input pull-up MOSs that can be controlled by software. The input pull-
can be used regardless of the operating mode. Table 8.2 summarizes the input pull-up MO

Table 8.2 Port 1 Input Pull-Up MOS States

Reset	Hardware Standby Mode	Software Standby Mode	In Other Operations
Off	Off	On/Off	On/Off

[Legend]

Off: Always off.

On/Off: On when P1DDR = 0 and P1PCR = 1; otherwise off.

The individual bits of P2DDR specify input or output for the pins of port 2.

Bit	Bit Name	Initial Value	R/W	Description
7	P27DDR	0	W	When set to 1, the corresponding pins fu output port pins; when cleared to 0, funct input port pins.
6	P26DDR	0	W	
5	P25DDR	0	W	
4	P24DDR	0	W	
3	P23DDR	0	W	• Normal extended mode (ADMXE = 0 When set to 1, the corresponding pin as address output pins; when cleared function as input port pins. The address output pins used are in with the settings of the IOSE and CS of SYSCR. • Address-data multiplex extended mo (ADMXE = 1) These bits correspond to the AD11 to of the address-data multiplex bus. • Single-chip mode When set to 1, the corresponding pin as output port pins; when cleared to as input port pins.
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	

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

P2PCR controls the port 2 built-in input pull-up MOSs.

Bit	Bit Name	Initial Value	R/W	Description
7	P27PCR	0	R/W	When the pins are in the input state, the corresponding input pull-up MOS is turned when a P2PCR bit is set to 1.
6	P26PCR	0	R/W	
5	P25PCR	0	R/W	
4	P24PCR	0	R/W	
3	P23PCR	0	R/W	When the pins are in the input state, the corresponding input pull-up MOS is turned when a P2PCR bit is set to 1. Do not change the initial value when using address-data multiplex extended bus mod
2	P22PCR	0	R/W	
1	P21PCR	0	R/W	
0	P20PCR	0	R/W	

11 in the table below is expressed by the following logical expression.

Address 11 = 1: $\overline{\text{ADFULLE}} \bullet \overline{\text{CS256E}} \bullet \text{IOSE}$

P23DDR	0		1		
ADMXE	0	1	0		
Address 11	X	X	0	1	
Pin function	P23 input pin	AD11 input/output pin	A11 output pin	P23 output pin	AD11 in

- P22 to P20

P2nDDR	0		1	
ADMXE	0	1	0	
Pin function	P2n input pin	ADm input/output pin	Am output pin	ADm inp p

[Legend] m = 10 to 8, n = 2 to 0, X: Don't care.

Pin function	P27 input pin	P27 output pin	$\overline{\text{DTR}}$ output pin	P27 input pin	P27 output pin	$\overline{\text{D}}$

[Legend] X: Don't care.

- P26/$\overline{\text{DSR}}$, P25/$\overline{\text{RI}}$, P24/$\overline{\text{DCD}}$

  The pin function is switched as shown below according to the P2nDDR bit.

P2nDDR	0	1
Pin function	P2n input pin $\overline{\text{DSR}}$/$\overline{\text{RI}}$/$\overline{\text{DCD}}$ input pin	P2n output pin

[Legend] n = 6 to 4

- P23 to P20

  The pin function is switched as shown below according to the P2nDDR bit.

P2nDDR	0	1
Pin function	P2n input pin	P2n output pin

[Legend] n = 3 to 0

Off: Always off

On/Off: On when P2DDR = 0 and P2PCR = 1; otherwise off.

- Noise canceler mode control register (P3NCMC)
- Noise cancel cycle setting register (NCCS)

(1) Port 3 Data Direction Register (P3DDR)

The individual bits of P3DDR specify input or output for the port 3 pins.

Bit	Bit Name	Initial Value	R/W	Description
7	P37DDR	0	W	• Normal extended mode (ADMXE = 0) The pins function as bidirectional data bus • Other modes When set to 1, the corresponding pins func output port pins; when cleared to 0, functio port pins.
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	cleared to 0, 1 is read.
1	P31DR	0	R/W	• Other modes
0	P30DR	0	R/W	P3DR stores output data for the port 3 pin used as the general output port. If this register is read, the P3DR values ar the bits with the corresponding P3DDR bi For the bits with the corresponding P3DD cleared to 0, the pin states are read.

(3) Port 3 Pull-Up MOS Control Register (P3PCR)

P3PCR controls the port 3 built-in input pull-up MOSs.

Bit	Bit Name	Initial Value	R/W	Description
7	P37PCR	0	R/W	• Normal extended mode (ADMXE = 0)
6	P36PCR	0	R/W	This register has no effect on operation.
5	P35PCR	0	R/W	• Other modes
4	P34PCR	0	R/W	When the pins are in the input state, the corresponding input pull-up MOS is turned
3	P33PCR	0	R/W	a P3PCR bit is set to 1.
2	P32PCR	0	R/W	
1	P31PCR	0	R/W	
0	P30PCR	0	R/W	

2	P32NCE	0	R/W	P3DR at the sampling cycle set by NCCS.
1	P31NCE	0	R/W	The operation changes according to the oth
0	P30NCE	0	R/W	control bits.

(5) Noise Canceler Mode Control Register (P3NCMC)

When the noise canceler is enabled, P3NCMC controls whether 1 or 0 is expected for the signal to port 3 in bit units.

Bit	Bit Name	Initial Value	R/W	Description
7	P37NCMC	1	R/W	• Normal extended mode (ADMXE = 0)
6	P36NCMC	1	R/W	This register has no effect on operation.
5	P35NCMC	1	R/W	• Other modes
4	P34NCMC	1	R/W	1 expected: 1 is stored in the port data regi while 1 is input stably.
3	P33NCMC	1	R/W	
2	P32NCMC	1	R/W	0 expected: 0 is stored in the port data regi while 0 is input stably.
1	P31NCMC	1	R/W	
0	P30NCMC	1	R/W	

001: 0.94 µs	ϕ/32	101: 1.9 ms	ϕ/6
010: 15.1 µs	ϕ/512	110: 3.9 ms	ϕ/1
011: 240.9 µs	ϕ/8192	111: 7.7 ms	ϕ/2

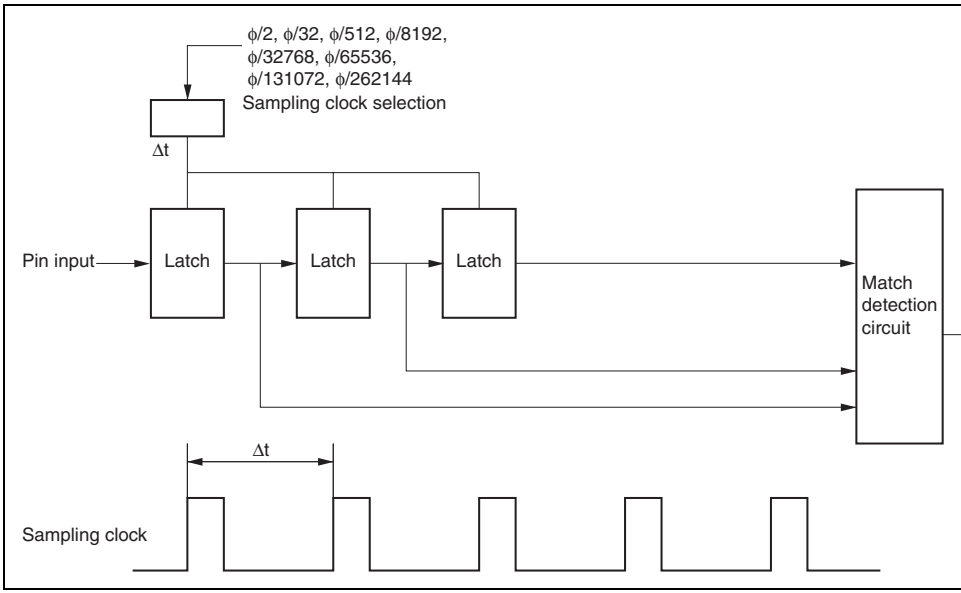


Figure 8.1 Noise Canceler Circuit

(n = 7 to 0)

Figure 8.2 Noise Canceler Operation

(7) Pin Functions

(a) Normal Extended Mode

Port 3 pins are automatically set to function as bidirectional data bus pins.

(b) Address-Data Multiplex Extended Mode

The operation is the same as that in single-chip mode.

(c) Single-Chip Mode

The pin function is switched as shown below according to the combination of the P3nDD
the P3nNCE bit.

P3nDDR	0		1
P3nNCE	0	1	X
Pin function	P3n input pin	ExDBn input	P3n output pin

[Legend] n = 7 to 0, X: Don't care.

Single-chip mode (EXPE = 0)	Off	Off	On/Off	On/C
Address-data multiplex extended mode (EXPE = 1, ADMXE = 1)				

[Legend]

Off: Always off.

On/Off: On when input state and P3PCR = 1; otherwise off.

- Noise canceler mode control register (P4BNCMC)
- Noise cancel cycle setting register (NCCS)

(1) Port 4 Data Direction Register (P4DDR)

The individual bits of P4DDR specify input or output for the port 4 pins. P4DDR is initia only by a system reset, and retains the value even if an internal reset signal of the WDT is generated.

Bit	Bit Name	Initial Value	R/W	Description
7	P47DDR	0	W	• Normal extended mode (ADMXE = 0) When set to 1, the corresponding pins func address output pins; when cleared to 0, fur input port pins. The address output pins used are in accord settings of the IOSE and CS256E bits of SY • Address-data multiplex extended mode (AD 1) These bits correspond to the AD15 to AD12 the address-data multiplex bus. • Single-chip mode When set to 1, the corresponding pins func output port pins; when cleared to 0, functio port pins.
6	P46DDR	0	W	
5	P45DDR	0	W	
4	P44DDR	0	W	

P4DR stores output data for the port 4 pins. P4DR is initialized only by a system reset, a the value even if an internal reset signal of the WDT is generated.

Bit	Bit Name	Initial Value	R/W	Description
7	P47DR	0	R/W	These bits store output data for the port 4 pin used as the general output port. If this register is read, the P4DR values are re bits with the corresponding P4DDR bits set to bits with the corresponding P4DDR bits cleare the pin states are read.
6	P46DR	0	R/W	
5	P45DR	0	R/W	
4	P44DR	0	R/W	
3	P43DR	0	R/W	• Normal extended mode (16-bit data bus) Since the corresponding pins function as bidirectional data bus pins, the value in th has no effect on operation. If this register is read, the P4DR values ar the bits with the corresponding P4DDR bi For the bits with the corresponding P4DD cleared to 0, 1 is read. • Other modes These bits store output data for the port 4 are used as the general output port. If this register is read, the P4DR values ar the bits with the corresponding P4DDR bi For the bits with the corresponding P4DD cleared to 0, the pin states are read.
2	P42DR	0	R/W	
1	P41DR	0	R/W	
0	P40DR	0	R/W	

2	P42PCR	0	R/W	
1	P41PCR	0	R/W	
0	P40PCR	0	R/W	

(4) Noise Canceler Enable Register (P4BNCE)

The individual bits of P4BNCE enable or disable the noise canceler circuits for ports 4 an

Bit	Bit Name	Initial Value	R/W	Description
7	P47NCE	0	R/W	Enables the noise canceler circuit for the corre
6	P46NCE	0	R/W	pin and the pin state is fetched into P4DR at th
5	P45NCE	0	R/W	sampling cycle set by NCCS.
4	P44NCE	0	R/W	The operation changes according to the other bits.
3 to 0	PB3NCE to PB0NCE	All 0	R/W	Bits for port B setting

	to PB0NCMC			

(6) Noise Canceler Cycle Setting Register (NCCS)

NCCS controls the sampling cycle of the noise cancelers.

Bit	Bit Name	Initial Value	R/W	Description
7 to 3	—	Undefined	R/W	Reserved Undefined value is read from these bits.
2	NCCK2	0	R/W	These bits set the sampling cycle of the nois cancelers. • When ϕ = 34 MHz 000: 0.06 µs ϕ/2 100: 963.8 µs 001: 0.94 µs ϕ/32 101: 1.9 ms 010: 15.1 µs ϕ/512 110: 3.9 ms 011: 240.9 µs ϕ/8192 111: 7.7 ms
1	NCCK1	0	R/W	
0	NCCK0	0	R/W	

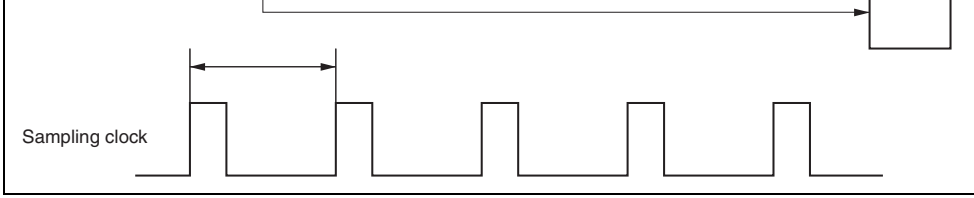


Figure 8.3 Noise Canceler Circuit

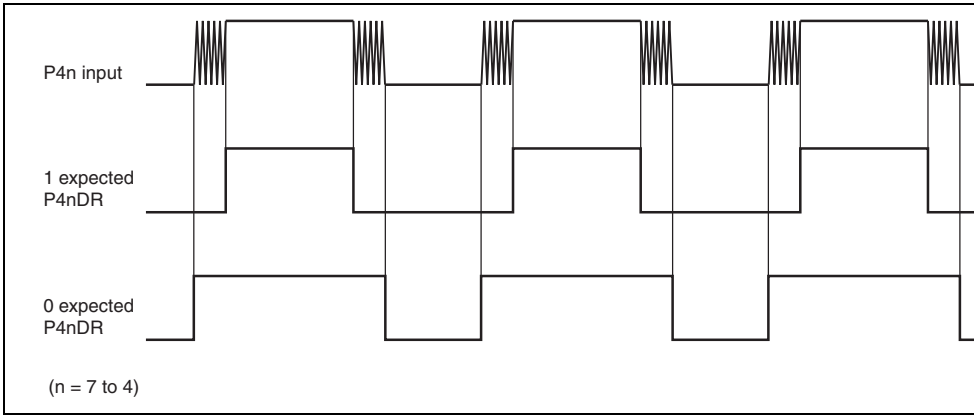


Figure 8.4 Noise Canceler Operation

P4nDDR	0	1	
Address 13	X	0	1
Pin function	P4n input pin	Am output pin	P4n out

[Legend] m = 15 to 12, n = 7 to 4
X: Don't care.

- P43 to P40

Port pins 43 to 40 function as bidirectional data bus pins in 16-bit bus extension, and ca as general I/O port in 8-bit bus extension.

(b) Address-Data Multiplex Extended Mode

Port pins 47 to 44 are automatically set to function as address bus pins. Port pins 43 to 4 used as general I/O port pins.

P4nNCE	0	1	X
Pin function	P4n input	DBn input	P4n output
	$\overline{\text{IRQn}}$ input	$\overline{\text{IRQn}}$ input (with the noise canceler)	

[Legend] n = 7 to 4
X: Don't care.

The pin function is switched as shown below according to the P4nDDR bit. When the ISS ISSR is cleared to 0 and the IRQnE bit in IER of the interrupt controller is set to 1, the pi used as the $\overline{\text{IRQn}}$ input pin. To use as the $\overline{\text{IRQn}}$ input pin, clear the P4nDDR bit to 0.

- P43 to P40

P4nDDR	0	1
Pin function	P4n input pin	P4n output pin
	$\overline{\text{IRQn}}$ input pin	

[Legend] n = 3 to 0

Single-chip mode (EXPE = 0)	Off	Off	On/Off	On/C
Address-data multiplex extended mode (EXPE = 1, ADMXE = 1)				

[Legend]

Off: Always off.

On/Off: On when input state and P4PCR = 1; otherwise off.

The individual bits of P5DDR specify input or output for the port 5 pins.

Bit	Bit Name	Initial Value	R/W	Description
7	P57DDR	0	W	If port 5 pins are specified for use as the gen port, the corresponding pins function as outp when the P5DDR bits are set to 1, and as inp when cleared to 0.
6	P56DDR	0	W	The corresponding port 5 pin functions as the clock output pin (φ) when this bit is set to 1, a the general I/O port when cleared to 0.
5	P55DDR	0	W	If port 5 pins are specified for use as the gen port, the corresponding pins function as outp when the P5DDR bits are set to 1, and as inp when cleared to 0.
4	P54DDR	0	W	
3	P53DDR	0	W	
2	P52DDR	0	W	
1	P51DDR	0	W	
0	P50DDR	0	W	

2	P52DR	0	R/W
1	P51DR	0	R/W
0	P50DR	0	R/W

Note: * The initial value is determined in accordance with the pin state of P56.

(3) Pin Functions

(a) Normal Extended Mode and Address-Data Multiplex Extended Mode

Port pin 57 is automatically set to function as a bus control output pin. The functions of
56 to 50 are the same as those in single-chip mode.

(b) Single-Chip Mode

Port 5 pins can operate as the SCIF, SCI_1, and SSU input/output, noise canceler input,
I/O port pins. The relationship between register setting values and pin functions are as fo

- P57

  The pin function is switched as shown below according to the P57DDR bit.

P57DDR	0	1
Pin function	P57 input pin	P57 output pin

The pin function is switched as shown below according to the RE bit in SSER of the S
the P55DDR bit. When the ISS13 bit in ISSR16 is cleared to 0 and the IRQ13E bit in
the interrupt controller is set to 1, this pin can be used as the $\overline{\text{IRQ13}}$ input pin. To use
$\overline{\text{IRQ13}}$ input pin, clear the P55DDR bit to 0.

RE	0		1
P55DDR	0	1	X
Pin function	P55 input pin	P55 output pin	SSI input p
	$\overline{\text{IRQ13}}$ input pin		

[Legend] X: Don't care.

- P54/$\overline{\text{IRQ12}}$/SSO

  The pin function is switched as shown below according to the TE bit in SSER of the S
  the P54DDR bit. When the ISS12 bit in ISSR16 is cleared to 0 and the IRQ12E bit in
  the interrupt controller is set to 1, this pin can be used as the $\overline{\text{IRQ12}}$ input pin. To use
  $\overline{\text{IRQ12}}$ input pin, clear the P54DDR bit to 0.

TE	0		1
P54DDR	0	1	X
Pin function	P54 input pin	P54 output pin	SSO output
	$\overline{\text{IRQ12}}$ input pin		

[Legend] X: Don't care.

	IRQ11 input pin		

[Legend] X: Don't care.

- P52/$\overline{\text{IRQ10}}$/TxD1

  The pin function is switched as shown below according to the combination of the TE
  SCR of SCI_1 and the P52DDR bit.

  When the ISS10 bit in ISSR16 is cleared to 0 and the IRQ10E bit in IER16 of the int
  controller is set to 1, this pin can be used as the $\overline{\text{IRQ10}}$ input pin. To use as the $\overline{\text{IRQ1}}$
  pin, clear the P52DDR bit to 0.

TE	0		1
P52DDR	0	1	X
Pin function	P52 input pin	P52 output pin	TxD1 output
	$\overline{\text{IRQ10}}$ input pin		

[Legend] X: Don't care.

	IRQ9 input pin		

[Legend] X: Don't care.

- P50/$\overline{\text{IRQ8}}$/TxDF

  The pin function is switched as shown below according to the combination of the enable/disable setting of the SCIF and the P50DDR bit.

  When the ISS8 bit in ISSR16 is cleared to 0 and the IRQ8E bit in IER16 of the interru controller is set to 1, this pin can be used as the $\overline{\text{IRQ8}}$ input pin. To use as the $\overline{\text{IRQ8}}$ in clear the P50DDR bit to 0.

SCIF	Disabled		Enabled
P50DDR	0	1	X
Pin function	P50 input pin	P50 output pin	TxDF output p
	$\overline{\text{IRQ8}}$ input pin		

[Legend] X: Don't care.

(1) Port 6 Data Direction Register (P6DDR)

The individual bits of P6DDR specify input or output for the pins of port 6.

Bit	Bit Name	Initial Value	R/W	Description
7	P67DDR	0	W	If port 6 pins are specified for use as the ge port, the corresponding pins function as out when the P6DDR bits are set to 1, and as in when cleared to 0.
6	P66DDR	0	W	
5	P65DDR	0	W	
4	P64DDR	0	W	
3	P63DDR	0	W	• Normal extended mode (16-bit bus) These bits have no effect on operation.
2	P62DDR	0	W	
1	P61DDR	0	W	• Other modes If port 6 pins are specified for use as the I/O port, the corresponding pins function port when the P6DDR bits are set to 1, a input port when cleared to 0.
0	P60DDR	0	W	

3	P63DR	0	R/W
2	P62DR	0	R/W
1	P61DR	0	R/W
0	P60DR	0	R/W

- Normal extended mode (16-bit data bus)

  Since the corresponding pins function as bidirectional data bus pins, the value in th has no effect on operation.
  If this register is read, the P6DR values a for the bits with the corresponding P6DDI to 1. For the bits with the corresponding P bits cleared to 0, 1 is read.

- Other modes

  These bits store output data for the port 6 that are used as the general output port.

  If this register is read, the P6DR values a for the bits with the corresponding P6DDI to 1. For the bits with the corresponding P bits cleared to 0, the pin states are read.

2	P62PCR	0	R/W
1	P61PCR	0	R/W
0	P60PCR	0	R/W

(4) Pin Functions

(a) Normal Extended Mode

- 16-bit bus mode
  Port pins 63 to 60 are automatically set to function as bidirectional data bus pins.
- 8-bit bus mode
  The operation is the same as that in single-chip mode.

(b) Address-Data Multiplex Extended Mode

The operation is the same as that in single-chip mode.

(c) Single-Chip Mode

Port 6 pins can operate as the PWMX output, SCIF and SSU control input/output, interr
or general I/O port pins. The relationship between register setting values and pin functio
follows.

- P66/$\overline{\text{ExIRQ9}}$/$\overline{\text{SCS}}$

  The pin function is switched as shown below according to the CSS1 and CSS0 bits in of the SSU and the P66DDR bit. When the ISS9 bit in ISSR16 is set to 1, this pin can as the $\overline{\text{ExIRQ9}}$ input pin. To use as the $\overline{\text{ExIRQ9}}$ input pin, clear the P66DDR bit to 0.

CSS1, CSS0	00		01 or 1X
P66DDR	0	1	X
Pin function	P66 input pin	P66 output pin	$\overline{\text{SCS}}$ I/O pi
	$\overline{\text{ExIRQ9}}$ input pin		

[Legend] X: Don't care.

- P65/$\overline{\text{ExIRQ10}}$/$\overline{\text{RTS}}$

  The pin function is switched as shown below according to the combination of the enable/disable setting of the SCIF and the P65DDR bit.

  When the ISS10 bit in ISSR16 is set to 1, this pin can be used as the $\overline{\text{ExIRQ10}}$ input p use as the $\overline{\text{ExIRQ10}}$ input pin, clear the P65DDR bit to 0.

SCIF	Disabled		Enabled
P65DDR	0	1	X
Pin function	P65 input pin	P65 output pin	$\overline{\text{RTS}}$ output pi
	$\overline{\text{IRQ10}}$ input pin		

[Legend] X: Don't care.

[Legend] X: Don't care.

- P63/PWX3

  The pin function is switched as shown below according to the combination of the OE DACR and the PWMXS bit in PTCNT0 of PWMX_1 and the P63DDR bit.

P63DDR	0		1		X
PWMXS	0	1	0	1	0
OEB	0	X	0	X	1
Pin function	P63 input pin		P63 output pin		PWX3 outp

[Legend] X: Don't care.

- P62/PWX2

  The pin function is switched as shown below according to the combination of the OE DACR and the PWMXS bit in PTCNT0 of PWMX_1 and the P62DDR bit.

P62DDR	0		1		X
PWMXS	0	1	0	1	0
OEA	0	X	0	X	1
Pin function	P62 input pin		P62 output pin		PWX2 outp

[Legend] X: Don't care.

[Legend] X: Don't care.

- P60/$\overline{\text{IRQ14}}$/PWX1

  The pin function is switched as shown below according to the combination of the OE
  DACR and the PWMXS bit in PTCNT0 of PWMX_0 and the P60DDR bit. To use th
  the $\overline{\text{IRQ14}}$ input pin, clear the P60DDR bit to 0.

P60DDR	0		1		X
PWMXS	0	1	0	1	0
OEA	0	X	0	X	1
Pin function	P60 input pin		P60 output pin		PWX0 outpu
	$\overline{\text{IRQ14}}$ input pin				

[Legend] X: Don't care.

(5) Port 6 Input Pull-Up MOS

Port 6 has built-in input pull-up MOSs that can be controlled by software. Table 8.6 sum
the input pull-up MOS states.

Table 8.6 Port 6 Input Pull-Up MOS States

Reset	Hardware Standby Mode	Software Standby Mode	In Other Opera
Off	Off	On/Off	On/Off

[Legend]
Off: Always off.
On/Off: On when input state, P6DDR = 0, and P6PCR = 1; otherwise off.

7	P77PIN	Undefined*	R	When this register is read, the pin states are
6	P76PIN	Undefined*	R	Since this register is allocated to the same a PBDDR, writing to this register writes data t and the port B setting is changed.
5	P75PIN	Undefined*	R	
4	P74PIN	Undefined*	R	
3	P73PIN	Undefined*	R	
2	P72PIN	Undefined*	R	
1	P71PIN	Undefined*	R	
0	P70PIN	Undefined*	R	

Note: * The initial values are determined in accordance with the pin states of P77 to P

(2) Pin Functions

Each pin of port 7 can also be used as the analog input pins of the A/D converter (AN0 t

- P77/AN7

  The pin function is switched as shown below according to the CH2 to CH0 bits in AD the A/D converter. Do not set these bits to other values than those shown in the follo table.

CH2 to CH0	B'111	Other than B'111
Pin function	AN7 input pin	P77 input pin

The pin function is switched as shown below according to the combination of the SCA in ADCR and the CH2 to CH0 bits in ADCSR of the A/D converter. Do not set these other values than those shown in the following table.

SCANE	0		1	
CH2 to CH0	B'101	Other than B'101	B'101 to B'111	B'000 to B'
Pin function	AN5 input pin	P75 input pin	AN5 input pin	P75 input p

- P74/AN4

  The pin function is switched as shown below according to the combination of the SCA in ADCR and the CH2 to CH0 bits in ADCSR of the A/D converter. Do not set these other values than those shown in the following table.

SCANE	0		1	
CH2 to CH0	B'100	Other than B'100	B'100 to B'111	B'000 to B'
Pin function	AN4 input pin	P74 input pin	AN4 input pin	P74 input p

[Legend] X: Don't care.

- P72/AN2

  The pin function is switched as shown below according to the combination of the SC
  SCANE bits in ADCR and the CH2 to CH0 bits in ADCSR of the A/D converter. Do
  these bits to other values than those shown in the following table.

SCANE	0		1			
SCANS	X		0		1	
CH2 to CH0	B'010	Other than B'010	B'010 to B'011	Other than B'010 to B'011	B'010 to B'111	
Pin function	AN2 input pin	P72 input pin	AN2 input pin	P72 input pin	AN2 input pin	P7

[Legend] X: Don't care.

- P71/AN1

  The pin function is switched as shown below according to the combination of the SC
  SCANS bits in ADCR and the CH2 to CH0 bits in ADCSR of the A/D converter. Do
  these bits to other values than those shown in the following table.

SCANE	0		1			
SCANS	X		0		1	
CH2 to CH0	B'001	Other than B'001	B'001 to B'011	Other than B'001 to B'011	B'001 to B'111	
Pin function	AN1 input pin	P71 input pin	AN1 input pin	P71 input pin	AN1 input pin	P

[Legend] X: Don't care.

[Legend] X: Don't care.

The individual bits of P8DDR specify input or output for the port 8 pins.

Bit	Bit Name	Initial Value	R/W	Description
7	P87DDR	0	W	If port 8 pins are specified for use as the ge port, the corresponding pins function as out when the P8DDR bits are set to 1, and as in when cleared to 0.
6	P86DDR	0	W	
5	P85DDR	0	W	
4	P84DDR	0	W	Since this register is allocated to the same a PBPIN, states of the port 8 pins are when th is read.
3	P83DDR	0	W	
2	P82DDR	0	W	
1	P81DDR	0	W	
0	P80DDR	0	W	

2	P82DR	0	R/W
1	P81DR	0	R/W
0	P80DR	0	R/W

(3) Pin Functions

The relationship between register setting values and pin functions are as follows.

- P87/$\overline{\text{ExIRQ15}}$/TxD3/$\overline{\text{ADTRG}}$

  The pin function is switched as shown below according to the combination of the TE
  SCR of SCI_3, the SMIF bit in SCMR, and the P87DDR bit.

  When the TRGS1 and EXTRGS bits are both set to 1 and the TRGS0 bit is cleared to
  ADCR of the A/D converter, this pin can be used as the $\overline{\text{ADTRG}}$ input pin.

  When the ISS15 bit in ISSR16 is set to 1, this pin can be used as the $\overline{\text{ExIRQ15}}$ input p
  use this pin as the $\overline{\text{ExIRQ15}}$ input pin, clear the P87DDR bit to 0.

P87DDR	0		1		
SMIF	0	1	0	1	0
TE	0	X	0	X	1
Pin function	P87 input pin		P87 output pin		TxD3 outp
	$\overline{\text{ExIRQ15}}$ input pin/ $\overline{\text{ADTRG}}$ input pin				

[Legend] X: Don't care.

	ExIRQ14 input pin			

- P85/ExIRQ13/SCK1

  The pin function is switched as shown below according to the combination of the C/
  SMR of SCI_1, the CKE1 and CKE0 bits in SCR, and the P85DDR bit.

  When the ISS13 bit in ISSR16 is set to 1, this pin can be used as the ExIRQ13 input
  use this pin as the ExIRQ13 input pin, clear the P85DDR bit to 0.

CKE1	0				
C/A	0			1	
CKE0	0		1	X	
P85DDR	0	1	X	X	
Pin function	P85 input pin	P85 output pin	SCK1 output pin	SCK1 output pin	SC
	ExIRQ13 input pin				

[Legend] X: Don't care.

Pin function	P84 input pin	P84 output pin	SCK3 output pin	SCK3 output pin	SCI
	$\overline{\text{ExIRQ12}}$ input pin				

[Legend] X: Don't care.

- P83/SDA1

  The pin function is switched as shown below according to the combination of the ICE ICCR of IIC_1 and the P83DDR bit.

  When this pin is used as the P83 output pin, the output format is NMOS push-pull out output format for SDA1 is NMOS open-drain output, which allows direct bus drive.

ICE	0		1
P83DDR	0	1	X
Pin function	P83 input pin	P83 output pin	SDA1 input/ou

[Legend] X: Don't care.

[Legend] X: Don't care.

- P81/SDA0

  The pin function is switched as shown below according to the combination of the IC
  ICCR of IIC_0 and the P81DDR bit.

  When this pin is used as the P81 output pin, the output format is NMOS push-pull ou
  output format for SDA0 is NMOS open-drain output, which allows direct bus drive.

ICE	0		1
P81DDR	0	1	X
Pin function	P81 input pin	P81 output pin	SDA0 input/o

[Legend] X: Don't care.

- P80/SCL0

  The pin function is switched as shown below according to the combination of the IC
  ICCR of IIC_0 and the P80DDR bit.

  When this pin is used as the P80 output pin, the output format is NMOS push-pull ou
  output format for SCL0 is NMOS open-drain output, which allows direct bus drive.

ICE	0		1
P80DDR	0	1	X
Pin function	P80 input pin	P80 output pin	SCL0 input/o

[Legend] X: Don't care.

Bit	Bit Name	Initial Value	R/W	Description
7	P97DDR	0	W	If port 9 pins are specified for use as the gen
6	P96DDR	0	W	port, the corresponding pins function as outp
5	P95DDR	0	W	when the P9DDR bits are set to 1, and as inp
4	P94DDR	0	W	when cleared to 0.
3	P93DDR	0	W	
2	P92DDR	0	W	
1	P91DDR	0	W	
0	P90DDR	0	W	

2	P92DR	0	R/W
1	P91DR	0	R/W
0	P90DR	0	R/W

(3) Pin Functions

The relationship between register setting values and pin functions are as follows.

- P97/$\overline{\text{WAIT}}$/$\overline{\text{CS256}}$

  The pin function is switched as shown below according to the operating mode and th
  combination of the CS256E bit in SYSCR, the WMS1 bit in WSCR and the P97DDI

Operating mode	Extended mode				Single-chip	
WMS1	0			1	X	
CS256E	0		1	X	X	
P97DDR	0	1	X	X	0	
Pin function	P97 input pin	P97 output pin	$\overline{\text{CS256}}$ output pin	$\overline{\text{WAIT}}$ input pin	P97 input pin	P

[Legend] X: Don't care.

Operating mode	Extended mode		Single-chip mode	
P95DDR	X		0	1
IOSE	0	1	X	X
Pin function	$\overline{AS}$ output pin	$\overline{IOS}$ output pin	P95 input pin	P95 out

[Legend] X: Don't care.

- P94/ExPWX1

  The pin function is switched as shown below according to the combination of the OE
  DACR and the PWMXS bit in PTCNT0 of the PWMX_0 module and the P94DDR bi

P94DDR	0		1		X
PWMXS	0	1	0	1	1
OEB	X	0	X	0	1
Pin function	P94 input pin		P94 output pin		ExPWX1 o

[Legend] X: Don't care.

- P93/ExPWX0

  The pin function is switched as shown below according to the combination of the OE
  DACR and the PWMXS bit in PTCNT0 of the PWMX_0 module and the P93DDR bi

P93DDR	0		1		X
PWMXS	0	1	0	1	1
OEA	X	0	X	0	1
Pin function	P93 input pin		P93 output pin		ExPWX0 o

[Legend] X: Don't care.

- P91/$\overline{\text{AH}}$

  The pin function is switched as shown below according to the operating mode, the A
  bit in SYSCR2, and the P91DDR bit.

Operating mode	Extended mode			Single-chip m	
ADMXE	0		1	X	
P91DDR	0	1	X	0	
Pin function	P91 input pin	P91 output pin	$\overline{\text{AH}}$ output pin	P91 input pin	P91

[Legend] X: Don't care.

- P90/$\overline{\text{LBE}}$

  The pin function is switched as shown below according to the operating mode, the C
  PTCNT0, and the P90DDR bit.

Operating mode	Extended mode			Single-chip m	
OBE	0		1	X	
P90DDR	0	1	X	0	
Pin function	P90 input pin	P90 output pin	$\overline{\text{LBE}}$ output pin	P90 input pin	P90

[Legend] X: Don't care.

(2) Port A Data Direction Register (PADDR)

The individual bits of PADDR specify input or output for the port A pins.

Bit	Bit Name	Initial Value	R/W	Description
7	PA7DDR	0	W	When set to 1, the corresponding pins functio output port pins; when cleared to 0, function port pins.
6	PA6DDR	0	W	
5	PA5DDR	0	W	As the address of this register is the same as PAPIN, reading from this register indicates th of port A.
4	PA4DDR	0	W	
3	PA3DDR	0	W	
2	PA2DDR	0	W	
1	PA1DDR	0	W	
0	PA0DDR	0	W	

2	PA2ODR	0	R/W
1	PA1ODR	0	R/W
0	PA0ODR	0	R/W

(3) Port A Input Data Register (PAPIN)

PAPIN indicates the states of the port A pins.

Bit	Bit Name	Initial Value	R/W	Description
7	PA7PIN	Undefined*	R	Pin states are read from this register.
6	PA6PIN	Undefined*	R	As the address of this register is the same a
5	PA5PIN	Undefined*	R	PADDR, writing to this register changes the
4	PA4PIN	Undefined*	R	of port A, that have been written to PADDR.
3	PA3PIN	Undefined*	R	
2	PA2PIN	Undefined*	R	
1	PA1PIN	Undefined*	R	
0	PA0PIN	Undefined*	R	

Note: The initial values are determined in accordance with the pin states of PA7 to PA0

Address 18 = 1: $\overline{\text{ADFULLE}}$

Address 13 = 1: $\overline{\text{ADFULLE}}$ • $\overline{\text{CS256E}}$ • IOSE

- PA7/$\overline{\text{ExIRQ7}}$/EVENT7/A23/EXOUT

  The pin function is switched as shown below according to the setting of address 18 an
  PA7DDR bit.

  Setting the ISS7 bit in ISSR makes the pin to function as the $\overline{\text{ExIRQ7}}$ input pin.

  When using the pin as the $\overline{\text{ExIRQ7}}$ input or an EVENT input pin, clear the PA7DDR
  Though the settings for the EVENT input pin have been made, set the PA7DDR bit to
  using the pin as the PA7 or A23 output pin.

  When the module stop mode is cleared in both the EtherC and E-DMAC, this pin func
  the EXOUT output pin.

PA7DDR	0	1	1
Address 18	X	1	0
Pin function	PA7 input pin	PA7 output pin	A23 out
	$\overline{\text{ExIRQ7}}$ input pin/EVENT7 input pin		

[Legend] X: Don't care.

PA5DDR	0	1	1
Address 18	1		
Pin function	PA6 input pin	PA6 output pin	A22 ou
	$\overline{\text{ExIRQ6}}$ input pin/EVENT6 input pin		

- PA5/$\overline{\text{ExIRQ5}}$/EVENT5/A21/WOL

  The pin function is switched as shown below according to the setting of the address PA5DDR bit.

  Setting the ISS5 bit in ISSR to 1 makes the pin function as the $\overline{\text{ExIRQ5}}$ input pin.

  When using the pin as the $\overline{\text{ExIRQ5}}$ input, or an EVENT input pin, clear the PA5DDI Though the settings for the EVENT input pin have been made, set the PA5DDR bit t using the pin as the A21 or PA5 output pin.

  When the module stop mode is cleared in both the EtherC, and E-DMAC, this pin fu the WOL output pin.

PA5DDR	0	1	1
Address 18	X	1	0
Pin function	PA5 input pin	PA5 output pin	A21 out
	$\overline{\text{ExIRQ5}}$ input pin/ EVENT5 input pin		

Pin function	PAn input pin	PAn output pin	Am outp
	$\overline{\text{ExIRQn}}$ input pin/EVENTn input pin		

[Legend] n = 4 to 2, m = 20 to 18
X: Don't care

- PA1/$\overline{\text{ExIRQ1}}$/EVENT1/A17, PA0/$\overline{\text{ExIRQ0}}$/EVENT0/A16

  The pin function is switched as shown below according to the setting of address 13 an PAnDDR bit.

  Setting the ISSn bit in ISSR makes the pin to function as the $\overline{\text{ExIRQn}}$ input pin.

  When using the pin as the $\overline{\text{ExIRQn}}$ input or an EVENT input pin, clear the PAnDDR Though the settings for the EVENT input pin have been made, set the PAnDDR bit to using the pin as the PAn or Am output pin.

PAnDDR	0	1	
Address 13	X	1	0
Pin function	PAn input pin	PAn output pin	Am outp
	$\overline{\text{ExIRQn}}$ input pin/EVENTn input pin		

[Legend] n = 1, 0; m = 17, 16
X: Don't care

the EXOUT output pin.

PA7DDR	0	1
Pin function	PA7 input pin	PA7 output pin
	$\overline{\text{ExIRQ7}}$ input pin/EVENT7 input pin	

- PA6/$\overline{\text{ExIRQ6}}$/EVENT6/LNKSTA

  The pin function is switched as shown below according to the PA6DDR bit.

  Setting the ISS6 bit in ISSR makes the pin to function as the $\overline{\text{ExIRQ6}}$ input pin.

  When using this pin as the $\overline{\text{ExIRQ6}}$ input, EVENT6 input, input pin, clear the PA6D 0. Though the settings for the EVENT input pin have been made, set the PA6DDR b use the pin as the PA6 output pin.

  When the module stop mode is cleared in both the EtherC and E-DMAC, this pin fu the LNKSTA input pin.

PA6DDR	0	1
Pin function	PA6 input pin	PA6 output pin
	$\overline{\text{ExIRQ6}}$ input pin/EVENT6 input pin	

Pin function	PA5 input pin	PA5 output pin
	$\overline{\text{ExIRQ5}}$ input pin/EVENT5 input pin	

- PA4/$\overline{\text{ExIRQ4}}$/EVENT4, PA3/$\overline{\text{ExIRQ3}}$/EVENT3, PA2/$\overline{\text{ExIRQ2}}$/EVENT2, PA1/$\overline{\text{ExIRQ1}}$/EVENT1, PA0/$\overline{\text{ExIRQ0}}$/EVENT0

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

  Setting the ISSn bit in ISSR makes the pin to function as the $\overline{\text{ExIRQn}}$ input pin.

  When using this pin as the $\overline{\text{ExIRQn}}$ input or EVENTn input pin, clear the PAnDDR b
  Though the settings for the EVENT input pin have been made, set the PAnDDR bit to
  the pin as the PAn output pin.

PAnDDR	0	1
Pin function	PAn input pin	PAn output pin
	$\overline{\text{ExIRQn}}$ input pin/EVENTn input pin	

[Legend] n = 4 to 0

The input pull-up MOS is in the OFF state after a reset and in hardware standby mode. It state is retained in software standby mode.

Table 8.7 summarizes the input pull-up MOS states.

Table 8.7 Input Pull-Up MOS States

Reset	Hardware Standby Mode	Software Standby Mode	In Other Operat
Off	Off	On/Off	On/Off

[Legend]

Off: Always off.

On/Off: On when PADDR = 0 and PAODR = 1; otherwise off.

- Noise canceler mode control register (P4BNCMC)
- Noise cancel cycle setting register (NCCS)

(1) Port B Data Direction Register (PBDDR)

The individual bits of PBDDR specify input or output for the pins of port B.

Bit	Bit Name	Initial Value	R/W	Description
7	PB7DDR	0	W	When set to 1, the corresponding pins function output port pins; when cleared to 0, function as port pins.
6	PB6DDR	0	W	
5	PB5DDR	0	W	
4	PB4DDR	0	W	
3	PB3DDR	0	W	
2	PB2DDR	0	W	
1	PB1DDR	0	W	
0	PB0DDR	0	W	

2	PB2DR	0	R/W
1	PB1DR	0	R/W
0	PB0DR	0	R/W

(3) Port B Input Data Register (PBPIN)

PBPIN indicates the states of the port B pins.

Bit	Bit Name	Initial Value	R/W	Description
7	PB7PIN	Undefined*	R	When this register is read, the pin states are
6	PB6PIN	Undefined*	R	Since this register is allocated to the same a
5	PB5PIN	Undefined*	R	P8DDR, writing to this register writes data t
4	PB4PIN	Undefined*	R	and the port 8 setting is changed.
3	PB3PIN	Undefined*	R	
2	PB2PIN	Undefined*	R	
1	PB1PIN	Undefined*	R	
0	PB0PIN	Undefined*	R	

Note: * The initial values are determined in accordance with the pin states of PB7 to

0	PB0NCE	0	R/W	bits. See section 8.1.11 (7), Pin Functions, for

(5) Noise Canceler Mode Control Register (P4BNCMC)

P4BNCMC controls whether 1 or 0 is expected for the input signal to port 4 and port B in units.

Bit	Bit Name	Initial Value	R/W	Description
7 to 4	P47NCMC to P44NCMC	All 1	R/W	Bits for port 4 setting
3	PB3NCMC	1	R/W	Expected value setting 1 expected: 1 is stored in the port data register is input stably. 0 expected: 0 is stored in the port data register is input stably.
2	PB2NCMC	1	R/W	
1	PB1NCMC	1	R/W	
0	PB0NCMC	1	R/W	

001: 0.94 µs	ϕ/32	101: 1.9 ms
010: 15.1 µs	ϕ/512	110: 3.9 ms
011: 240.9 µs	ϕ/8192	111: 7.7 ms

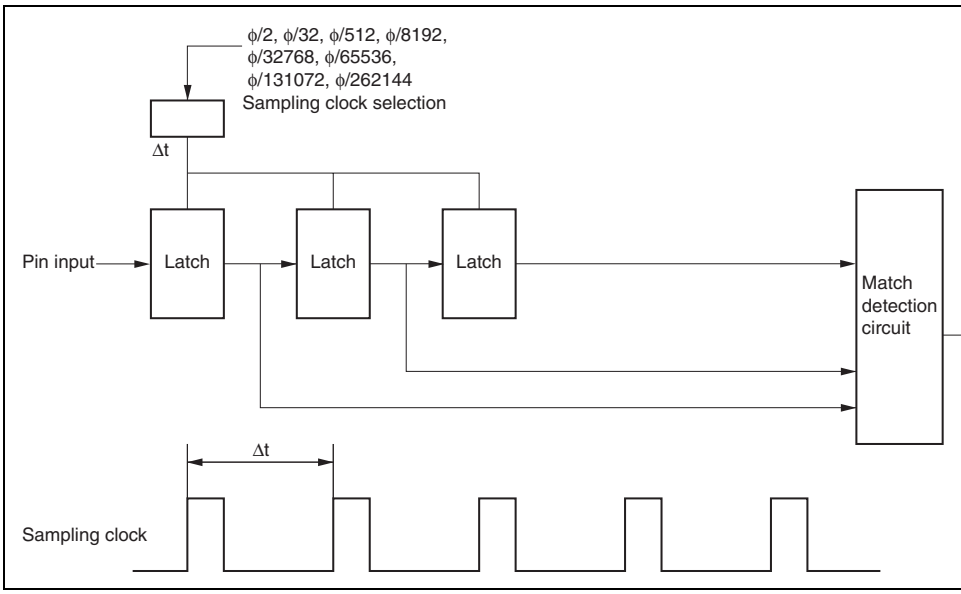


Figure 8.5 Noise Canceler Circuit

(n = 3 to 0)

Figure 8.6 Noise Canceler Operation

Event counter	Disabled	Enabled	X	X
Pin function	PBn input pin	EVENTm input pin	PBn output pin	RM_X EtherC

[Legend] n = 7 to 4, m = 15 to 12, X: Don't care.

Note: * See section 7.3, DTC Event Counter, for the event counter settings.

- PB3/EVENT11/DB3/RM_RXD1, PB2/EVENT10/DB2/RM_RXD0, PB1/EVENT9/DB1/RM_TXD1, PB0/EVENT8/DB0/RM_TXD0

  The pin function is switched as shown below according to the combination of the mo state in the EtherC, E-DMAC, the PBnDDR bit and the PBnNCE bit.

EtherC, E-DMAC	Either of them is stopped				Both of t stop
PBnDDR	0			1	X
Event counter	Disabled		Enabled	X	X
PBnNCE	0	1	X	X	X
Pin function	PBn input pin		EVENTm input pin	PBn output pin	RM_X EtherC

[Legend] n = 3 to 0, m = 11 to 8, X: Don't care.

The individual bits of PCDDR specify input or output for the port C pins.

Bit	Bit Name	Initial Value	R/W	Description
7	PC7DDR	0	W	When set to 1, the corresponding pins functio output port pins; when cleared to 0, function port pins. Since this register is allocated to the same ad PCPIN, states of the port C pins are returned this register is read.
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	PC2ODR	0	R/W	
1	PC1ODR	0	R/W	
0	PC0ODR	0	R/W	

(3) Port C Input Data Register (PCPIN)

PCPIN indicates the pin states of port C.

Bit	Bit Name	Initial Value	R/W	Description
7	PC7PIN	Undefined*	R	When this register is read, the pin states are
6	PC6 PIN	Undefined*	R	Since this register is allocated to the same a
5	PC5PIN	Undefined*	R	PCDDR, writing to this register writes data t
4	PC4 PIN	Undefined*	R	and the port C setting is changed.
3	PC3 PIN	Undefined*	R	
2	PC2 PIN	Undefined*	R	
1	PC1 PIN	Undefined*	R	
0	PC0 PIN	Undefined*	R	

Note: The initial values are determined in accordance with the states of PC7 to PC0 pin

bus width, the pin function is the same as that in single-chip mode.

- PC5 to PC0

  The pin functions are the same as those in single-chip mode.

(b) Single-Chip Mode

- PC7, PC6

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

PCnDDR	0	1
Pin function	PCn input pin	PCn output pin

[Legend] n = 7, 6

- PC5/SDA4

  The pin function is switched as shown below according to the combination of the ICE ICCR of the IIC_4 and the PC5DDR bit.

ICE	0		1
PC5DDR	0	1	X
Pin function	PC5 input pin	PC5 output pin	SDA4 input/out

[Legend] X: Don't care.

The pin function is switched as shown below according to the combination of the IC
ICCR of the IIC_3 and the PC3DDR bit.

ICE	0		1
PC3DDR	0	1	X
Pin function	PC3 input pin	PC3 output pin	SDA3 input/ou

[Legend] X: Don't care.

- PC2/SCL3

  The pin function is switched as shown below according to the combination of the IC
  ICCR of the IIC_3 and the PC2DDR bit.

ICE	0		1
PC2DDR	0	1	X
Pin function	PC2 input pin	PC2 output pin	SCL3 input/o

[Legend] X: Don't care.

- PC1/SDA2

  The pin function is switched as shown below according to the combination of the IC
  ICCR of the IIC_2 and the PC1DDR bit.

ICE	0		1
PC1DDR	0	1	X
Pin function	PC1 input pin	PC1 output pin	SDA2 input/o

[Legend] X: Don't care.

(2) Port D Data Direction Register (PDDDR)

The individual bits of PDDDR specify input or output for the port D pins.

Bit	Bit Name	Initial Value	R/W	Description
7	PD7DDR	0	W	If port D pins are specified for use as the ge
6	PD6DDR	0	W	port, the corresponding pins function as out
5	PD5DDR	0	W	when the PDDDR bits are set to 1, and as i
4	PD4DDR	0	W	when cleared to 0.
3	PD3DDR	0	W	Since this register is allocated to the same a
2	PD2DDR	0	W	PDPIN, the states of the port D pins are retu
1	PD1DDR	0	W	when this register is read.
0	PD0DDR	0	W	

2	PD2ODR	0	R/W
1	PD1ODR	0	R/W
0	PD0ODR	0	R/W

(3) Port D Input Data Register (PDPIN)

PDPIN indicates the pin states of port D.

Bit	Bit Name	Initial Value	R/W	Description
7	PD7PIN	Undefined*	R	When this register is read, the pin states are
6	PD6 PIN	Undefined*	R	Since this register is allocated to the same ad
5	PD5PIN	Undefined*	R	PDDDR, writing to this register writes data to
4	PD4 PIN	Undefined*	R	and the port D setting is changed.
3	PD3 PIN	Undefined*	R	
2	PD2 PIN	Undefined*	R	
1	PD1 PIN	Undefined*	R	
0	PD0 PIN	Undefined*	R	

Note: The initial values are determined in accordance with the states of PD7 to PD0 pins

ICE	0		1
PD7DDR	0	1	X
Pin function	PD7 input pin	PD7 output pin	SDA5 input/ou

[Legend] X: Don't care.

- PD6/SCL5

  The pin function is switched as shown below according to the combination of the IC ICCR of the IIC_5 and the PD6DDR bit.

ICE	0		1
PD6DDR	0	1	X
Pin function	PD6 input pin	PD6 output pin	SCL5 input/ou

[Legend] X: Don't care.

- PD5/$\overline{\text{LPCPD}}$

  The pin function is switched as shown below according to the PD5DDR bit. This pin used as the $\overline{\text{LPCPD}}$ input pin when the LPC is enabled.

LPC	Disabled		Enable
PD5DDR	0	1	0
Pin function	PD5 input pin	PD5 output pin	$\overline{\text{LPCPD}}$ inp

in HICR0 of the LPC and the PD3DDR bit.

FGA20E	0		1
PD3DDR	0	1	0
Pin function	PD3 input pin	PD3 output pin	GA20 output

- PD2/$\overline{\text{PME}}$

  The pin function is switched as shown below according to the combination of the PM
  HICR0 of the LPC and the PD2DDR bit.

PMEE	0		1
PD2DDR	0	1	0
Pin function	PD2 input pin	PD2 output pin	$\overline{\text{PME}}$ output

- PD1/$\overline{\text{LSMI}}$

  The pin function is switched as shown below according to the combination of the LSM
  HICR0 of the LPC and the PD1DDR bit.

LSMIE	0		1
PD1DDR	0	1	0
Pin function	PD1 input pin	PD1 output pin	$\overline{\text{LSMI}}$ output

Port pins D5 to D0 have built-in input pull-up MOSs that can be controlled by software. pull-up MOS can be used in any operating mode, and can be specified as on or off on a b basis.

PDnDDR	0		1
PDnODR	1	0	X
PDn pull-up MOS	ON	OFF	OFF

[Legend] n = 5 to 0, X: Don't care.

The input pull-up MOS is in the off state after a reset and in hardware standby mode. Th state is retained in software standby mode.

Table 8.8 summarizes the input pull-up MOS states.

Table 8.8 Port D Input Pull-Up MOS States

Reset	Hardware Standby Mode	Software Standby Mode	In Other Operat
Off	Off	On/Off	On/Off

[Legend]

Off: Always off.

On/Off: On when PDDDR = 0 and PDODR = 1; otherwise off.

The individual bits of PEDDR specify input or output for the port E pins.

Bit	Bit Name	Initial Value	R/W	Description
7	PE7DDR	0	W	When set to 1, the corresponding pins functio output port pins; when cleared to 0, function port pins. Since this register is allocated to the same ad PEPIN, states of the port E pins are returned this register is read.
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) Port E Output Data Register (PEODR)

PEODR stores output data for the port E pins.

Bit	Bit Name	Initial Value	R/W	Description
7	PE7ODR	0	R/W	The PEODR register stores the output data f pins that are used as the general output port.
6	PE6ODR	0	R/W	
5	PE5ODR	0	R/W	
4	PE4ODR	0	R/W	
3	PE3ODR	0	R/W	
2	PE2ODR	0	R/W	
1	PE1ODR	0	R/W	
0	PE0ODR	0	R/W	

2	PE2PIN	Undefined*	R
1	PE1PIN	Undefined*	R
0	PE0PIN	Undefined*	R

Note: The initial value of these pins is determined in accordance with the state of pins P PE0.

(4) Pin Functions

Port E pins can also function as LPC input/output pins. The pin function is switched acc whether the LPC module is enabled or disabled. The LPC is disabled when all of the bit LPC2E, and LPC3E in HICR0 and SCIFE in HICR5 are cleared to 0.

- PE7/SERIRQ

  The pin function is switched as shown below according to whether the LPC is enable disabled and the PE7DDR bit.

LPC	Disabled		Enable
PE7DDR	0	1	X
Pin function	PE7 input pin	PE7 output pin	SERIRQ input/

[Legend] X: Don't care.

The pin function is switched as shown below according to whether the LPC is enabled disabled and the PE5DDR bit.

LPC	Disabled		Enabled
PE5DDR	0	1	X
Pin function	PE5 input pin	PE5 output pin	$\overline{\text{LRESET}}$ inp

[Legend] X: Don't care.

- PE4/$\overline{\text{LFRAME}}$

  The pin function is switched as shown below according to whether the LPC is enabled disabled and the PE4DDR bit.

LPC	Disabled		Enabled
PE4DDR	0	1	X
Pin function	PE4 input pin	PE4 output pin	$\overline{\text{LFRAME}}$ inp

[Legend] X: Don't care.

- PE3/LAD3

  The pin function is switched as shown below according to whether the LPC is enabled disabled and the PE3DDR bit.

LPC	Disabled		Enabled
PE3DDR	0	1	X
Pin function	PE3 input pin	PE3 output pin	LAD3 input/out

[Legend] X: Don't care.

The pin function is switched as shown below according to whether the LPC is enable disabled and the PE1DDR bit.

LPC	Disabled		Enable
PE1DDR	0	1	X
Pin function	PE1 input pin	PE1 output pin	LAD1 input/ou

[Legend] X: Don't care.

- PE0/LAD0

  The pin function is switched as shown below according to whether the LPC is enable disabled and the PE0DDR bit.

LPC	Disabled		Enable
PE0DDR	0	1	X
Pin function	PE0 input pin	PE0 output pin	LAD0 input/ou

[Legend] X: Don't care.

The individual bits of PFDDR specify input or output for the port F pins. PFDDR is initia
only by a system reset, and retains the value even if an internal reset signal of the WDT is
generated.

Bit	Bit Name	Initial Value	R/W	Description
7	—	—	—	Reserved
6	PF6DDR	0	W	When set to 1, the corresponding pin functio
5	PF5DDR	0	W	output port pin; when cleared to 0, functions
4	PF4DDR	0	W	input port pin.
3	PF3DDR	0	W	Since this register is allocated to the same ad
2	PF2DDR	0	W	PFPIN, states of the port F pins are returned
1	PF1DDR	0	W	this register is read.
0	PF0DDR	0	W	

4	PF4ODR	0	R/W	
3	PF3ODR	0	R/W	
2	PF2ODR	0	R/W	
1	PF1ODR	0	R/W	
0	PF0ODR	0	R/W	

(3) Port F Input Data Register (PFPIN)

PFPIN indicates the pin states of port F.

Bit	Bit Name	Initial Value	R/W	Description
7	—	—	—	Reserved Undefined value is read from this bit.
6	PF6PIN	Undefined*	R	When this register is read, the pin states are
5	PF5PIN	Undefined*	R	Since this register is allocated to the same a
4	PF4PIN	Undefined*	R	PFDDR, writing to this register writes data t
3	PF3PIN	Undefined*	R	and the port F setting is changed.
2	PF2PIN	Undefined*	R	
1	PF1PIN	Undefined*	R	
0	PF0PIN	Undefined*	R	

Note: The initial value of these pins is determined in accordance with the state of pins P PF0.

PWMXS	0	1	0	1	1
OEA	X	0	X	0	1
Pin function	PF6 input pin		PF6 output pin		ExPWX2 outp

[Legend] X: Don't care.

- PF5/RS13

  The pin function is switched as shown below according to the PF5DDR bit.

PF5DDR	0	1
Pin function	PF5 input pin	PF5 output pin

- PF4/RS12

  The pin function is switched as shown below according to the PF4DDR bit.

PF4DDR	0	1
Pin function	PF4 input pin	PF4 output pin

- PF3/ExPWX3/RS11

  The pin function is switched as shown below according to the combination of the OE
  DACR and the PWMXS bit in PTCNT0 of PWMX_1 and the PF3DDR bit.

PF3DDR	0		1		X
PWMXS	0	1	0	1	1
OEA	X	0	X	0	1
Pin function	PF3 input pin		PF3 output pin		ExPWX3 outp

[Legend] X: Don't care.

EtherC, E-DMAC	Ether of them is stopped		Both of them a
PFnDDR	0	1	X
Pin function	PFn input pin	PFn output pin	MDC outp MDIO input/o

[Legend] X: Don't care. n = 1, 0

MOSs.

Port 3 pins and pins 47 to 44 and B3 to B0 have built-in de-bouncers (DBn) that eliminat
in the input signals.

Ports 4 and F are designed for retain state outputs (RSn), which retain the output values o
pins even if a reset is generated when the watchdog timer has overflowed.

Ports 1 to 6, and 8 to E can drive a single TTL load and 30 pF capacitive load. All the I/C
can drive a Darlington transistor in output mode. Port pins 80 to 83, C0 to C5, D6, and D
NMOS push-pull output.

Port 2	General I/O port multiplexed with SCIF control I/O	P27/$\overline{\text{DTR}}$ P26/$\overline{\text{DSR}}$ P25/$\overline{\text{RI}}$ P24/$\overline{\text{DCD}}$	Same as left	Bu pu
	General I/O port multiplexed with address output and address-data multiplex I/O	P23 P22 P21 P20	P23/A11/AD11 P22/A10/AD10 P21/A9/AD9 P20/A8/AD8	
Port 3	General I/O port multiplexed with de-bounced input and bidirectional data bus I/O	P37/ExDB7 P36/ExDB6 P35/ExDB5 P34/ExDB4 P33/ExDB3 P32/ExDB2 P31/ExDB1 P30/ExDB0	P37/ExDB7/D15 P36/ExDB6/D14 P35/ExDB5/D13 P34/ExDB4/D12 P33/ExDB3/D11 P32/ExDB2/D10 P31/ExDB1/D9 P30/ExDB0/D8	Bu pu
Port 4	General I/O port multiplexed with interrupt input, de-bounced input, address output, and address-data multiplex I/O	P47/$\overline{\text{IRQ7}}$/RS7/DB7/HC7 P46/$\overline{\text{IRQ6}}$/RS6/DB6/HC6 P45/$\overline{\text{IRQ5}}$/RS5/DB5/HC5 P44/$\overline{\text{IRQ4}}$/RS4/DB4/HC4	P47/A15/AD15 P46/A14/AD14 P45/A13/AD13 P44/A12/AD12	Bu pu LE ca (si 12
	General I/O port multiplexed with interrupt input and bidirectional data bus* I/O	P43/$\overline{\text{IRQ3}}$/RS3/HC3 P42/$\overline{\text{IRQ2}}$/RS2/HC2 P41/$\overline{\text{IRQ1}}$/RS1/HC1 P40/$\overline{\text{IRQ0}}$/RS0/HC0	P43/$\overline{\text{IRQ3}}$/RS3/HC3/D7* P42/$\overline{\text{IRQ2}}$/RS2/HC2/D6* P41/$\overline{\text{IRQ1}}$/RS1/HC1/D5* P40/$\overline{\text{IRQ0}}$/RS0/HC0/D4*	

	and SCI_1 I/O	P50/IRQ8/TxDF		
Port 6	General I/O port multiplexed with interrupt input, SCIF control I/O, and SSU control I/O	P67/ExIRQ8/SSCK P66/ExIRQ9/SCS P65/ExIRQ10/RTS P64/ExIRQ11/CTS	Same as left	Buil pull
	General I/O port multiplexed with interrupt input, PWMX output, and bidirectional data bus* I/O	P63/PWX3 P62/PWX2 P61/IRQ15/PWX1 P60/IRQ14/PWX0	P63/PWX3/D3* P62/PWX2/D2* P61/IRQ15/PWX1/D1* P60/IRQ14/PWX0/D0*	
Port 7	General input port multiplexed with A/D converter analog input	P77/AN7 P76/AN6 P75/AN5 P74/AN4 P73/AN3 P72/AN2 P71/AN1 P70/AN0	Same as left	
Port 8	General I/O port multiplexed with interrupt input, A/D converter external trigger input, and SCI_1 and SCI_3 I/O	P87/ExIRQ15/TxD3/ADTRG P86/ExIRQ14/RxD3 P85/ExIRQ13/SCK1 P84/ExIRQ12/SCK3	Same as left	
	General I/O port multiplexed with IIC_0 and IIC_1 I/O	P83/SDA1 P82/SCL1 P81/SDA0 P80/SCL0	Same as left	NM pus outp

Port A	General I/O port multiplexed with interrupt input, DTC event counter input, EtherC control I/O, and address output	PA7/ExIRQ7/EVENT7 PA6/ExIRQ6/EVENT6/LNKSTA PA5/ExIRQ5/EVENT5/WOL PA4/ExIRQ4/EVENT4 PA3/ExIRQ3/EVENT3 PA2/ExIRQ2/EVENT2 PA1/ExIRQ1/EVENT1 PA0/ExIRQ0/EVENT0	PA7/ExIRQ7/EVENT7/A23 PA6/ExIRQ6/EVENT6/LNKSTA/A22 PA5/ExIRQ5/EVENT5/WOL/A21 PA4/ExIRQ4/EVENT4/A20 PA3/ExIRQ3/EVENT3/A19 PA2/ExIRQ2/EVENT2/A18 PA1/ExIRQ1/EVENT1/A17 PA0/ExIRQ0/EVENT0/A16	Bu pu
Port B	General I/O port multiplexed with DTC event counter input and EtherC control I/O	PB7/EVENT15/RM_RX-ER PB6/EVENT14/RM_CRS-DV PB5/EVENT13/RM_REF-CLK PB4/EVENT12/RM_TX-EN	Same as left	
	General I/O port multiplexed with de-bounced input, DTC event counter input, and EtherC control I/O	PB3/DB3/EVENT11/RM_RXD1 PB2/DB2/EVENT10/RM_RXD0 PB1/DB1/EVENT9/RM_TXD1 PB0/DB0/EVENT8/RM_TXD0	Same as left	
Port C	General I/O port multiplexed with bus control output	PC7 PC6	RD PC6/LWR	
	General I/O port multiplexed with IIC_2 to IIC_4 I/O	PC5/SDA4 PC4/SCL4 PC3/SDA3 PC2/SCL3 PC1/SDA2 PC0/SCL2	Same as left	NM pu ou

	multiplexed with LPC I/O	PE6/LCLK PE5/$\overline{\text{LRESET}}$ PE4/$\overline{\text{LFRAME}}$ PE3/LAD3 PE2/LAD2 PE1/LAD1 PE0/LAD0	
Port F	General I/O port multiplexed with PWMX output and EtherC control I/O	PF6/ExPWX2/RS14 PF1/RS9/MDC PF0/RS8/MDIO	Same as left

Note: * Available when configured for 16-bit data bus.

(2) Port 1 Data Direction Register (P1DDR)

The individual bits of P1DDR specify input or output for the pins of port 1.

Bit	Bit Name	Initial Value	R/W	Description
7	P17DDR	0	W	• Normal extended mode (ADMXE = 0)
6	P16DDR	0	W	When set to 1, the corresponding pins fu
5	P15DDR	0	W	address output pins; when cleared to 0,
4	P14DDR	0	W	as input port pins.
3	P13DDR	0	W	• Address-data multiplex extended mode
2	P12DDR	0	W	1)
1	P11DDR	0	W	These bits correspond to the AD7 to AD
0	P10DDR	0	W	the address-data multiplex bus.
				• Single-chip mode
				When set to 1, the corresponding pins fu
				output port pins; when cleared to 0, func
				input port pins.

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

(3) Port 1 Pull-Up MOS Control Register (P1PCR)

P1PCR controls the port 1 built-in input pull-up MOSs.

Bit	Bit Name	Initial Value	R/W	Description
7	P17PCR	0	R/W	When the pins are in the input state, the corresponding input pull-up MOS is turned on if a P1PCR bit is set to 1. Do not change the initial value when using the address-data multiplex extended bus mode.
6	P16PCR	0	R/W	
5	P15PCR	0	R/W	
4	P14PCR	0	R/W	
3	P13PCR	0	R/W	
2	P12PCR	0	R/W	
1	P11PCR	0	R/W	
0	P10PCR	0	R/W	

Pin function	P1n input pin	ADn input/output pin	P1n input pin	An output pin	Setting prohibited	P

[Legend] n = 7 to 0, X: Don't care.

(b) Single-Chip Mode (EXPE = 0)

The pin function is switched as shown below according to the P1nDDR bit.

P1nDDR	0	1
Pin function	P1n input pin	P1n output pin

[Legend] n = 7 to 0

(5) Port 1 Input Pull-Up MOS

Port 1 has built-in input pull-up MOSs that can be controlled by software. The input pul
can be used regardless of the operating mode. Table 8.10 summarizes the input pull-up M
states.

Table 8.10 Port 1 Input Pull-Up MOS States

Reset	Hardware Standby Mode	Software Standby Mode	In Other Operations
Off	Off	On/Off	On/Off

[Legend]

Off: Always off.

On/Off: On when P1DDR = 0 and P1PCR = 1; otherwise off.

The individual bits of P2DDR specify input or output for the pins of port 2.

Bit	Bit Name	Initial Value	R/W	Description
7	P27DDR	0	W	When set to 1, the corresponding pins fun output port pins; when cleared to 0, functi input port pins.
6	P26DDR	0	W	
5	P25DDR	0	W	
4	P24DDR	0	W	
3	P23DDR	0	W	• Normal extended mode (ADMXE = 0) When set to 1, the corresponding pins as address output pins; when cleared function as input port pins. The address output pins used are in a with the settings of the IOSE and CS2 of SYSCR. • Address-data multiplex extended mod (ADMXE = 1) These bits correspond to the AD11 to of the address-data multiplex bus. • Single-chip mode When set to 1, the corresponding pins as output port pins; when cleared to 0, as input port pins.
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	

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

P2PCR controls the port 2 built-in input pull-up MOSs.

Bit	Bit Name	Initial Value	R/W	Description
7	P27PCR	0	R/W	When the pins are in the input state, the corresponding input pull-up MOS is turne when a P2PCR bit is set to 1.
6	P26PCR	0	R/W	
5	P25PCR	0	R/W	
4	P24PCR	0	R/W	
3	P23PCR	0	R/W	When the pins are in the input state, the corresponding input pull-up MOS is turne when a P2PCR bit is set to 1.
2	P22PCR	0	R/W	
1	P21PCR	0	R/W	Do not change the initial value when usin
0	P20PCR	0	R/W	address-data multiplex extended bus mo

11 in the table below is expressed by the following logical expression.

Address 11 = 1: $\overline{\text{ADFULLE}} \bullet \overline{\text{CS256E}} \bullet \text{IOSE}$

P23DDR	0		1		
ADMXE	0	1	0		
Address 11	X	X	0	1	X
Pin function	P23 input pin	AD11 input/output pin	A11 output pin	P23 output pin	AD11 inp p

- P22 to P20

P2nDDR	0		1	
ADMXE	0	1	0	1
Pin function	P2n input pin	ADm input/output pin	Am output pin	ADm inpu pir

[Legend] m = 10 to 8, n = 2 to 0, X: Don't care.

Pin function	P27 input pin	P27 output pin	$\overline{\text{DTR}}$ output pin	P27 input pin	P27 output pin	$\overline{\text{D}}$

[Legend] X: Don't care.

- P26/$\overline{\text{DSR}}$, P25/$\overline{\text{RI}}$, P24/$\overline{\text{DCD}}$

  The pin function is switched as shown below according to the P2nDDR bit.

P2nDDR	0	1
Pin function	P2n input pin $\overline{\text{DSR}}$/$\overline{\text{RI}}$/$\overline{\text{DCD}}$ input pin	P2n output pin

[Legend] n = 6 to 4

- P23 to P20

  The pin function is switched as shown below according to the P2nDDR bit.

P2nDDR	0	1
Pin function	P2n input pin	P2n output pin

[Legend] n = 3 to 0

[Legend]

Off: Always off.

On/Off: On when P2DDR = 0 and P2PCR = 1; otherwise off.

- Noise canceler mode control register (P3NCMC)
- Noise cancel cycle setting register (NCCS)

(1) Port 3 Data Direction Register (P3DDR)

The individual bits of P3DDR specify input or output for the port 3 pins.

Bit	Bit Name	Initial Value	R/W	Description
7	P37DDR	0	W	• Normal extended mode (ADMXE = 0) The pins function as bidirectional data bus • Other modes When set to 1, the corresponding pins fun output port pins; when cleared to 0, functi port pins.
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	cleared to 0, 1 is read.
1	P31DR	0	R/W	• Other modes
0	P30DR	0	R/W	P3DR stores output data for the port 3 pins used as the general output port. If this register is read, the P3DR values are the bits with the corresponding P3DDR bits For the bits with the corresponding P3DDR cleared to 0, the pin states are read.

(3) Port 3 Pull-Up MOS Control Register (P3PCR)

P3PCR controls the port 3 built-in input pull-up MOSs.

Bit	Bit Name	Initial Value	R/W	Description
7	P37PCR	0	R/W	• Normal extended mode (ADMXE = 0)
6	P36PCR	0	R/W	This register has no effect on operation.
5	P35PCR	0	R/W	• Other modes
4	P34PCR	0	R/W	When the pins are in the input state, the corresponding input pull-up MOS is turned a P3PCR bit is set to 1.
3	P33PCR	0	R/W	
2	P32PCR	0	R/W	
1	P31PCR	0	R/W	
0	P30PCR	0	R/W	

2	P32NCE	0	R/W	P3DR at the sampling cycle set by NCCS
1	P31NCE	0	R/W	The operation changes according to the o control bits. See section 8.2.3 (7), Pin Fur details.
0	P30NCE	0	R/W	

(5) Noise Canceler Mode Control Register (P3NCMC)

When the noise canceler is enabled, P3NCMC controls whether 1 or 0 is expected for th signal to port 3 in bit units.

Bit	Bit Name	Initial Value	R/W	Description
7	P37NCMC	1	R/W	• Normal extended mode (ADMXE = 0)
6	P36NCMC	1	R/W	This register has no effect on operation.
5	P35NCMC	1	R/W	• Other modes
4	P34NCMC	1	R/W	1 expected: 1 is stored in the port data reg while 1 is input stably.
3	P33NCMC	1	R/W	
2	P32NCMC	1	R/W	0 expected: 0 is stored in the port data reg while 0 is input stably.
1	P31NCMC	1	R/W	
0	P30NCMC	1	R/W	

001: 0.94 µs	φ/32	101: 1.9 ms	φ/65
010: 15.1 µs	φ/512	110: 3.9 ms	φ/13
011: 240.9 µs	φ/8192	111: 7.7 ms	φ/26

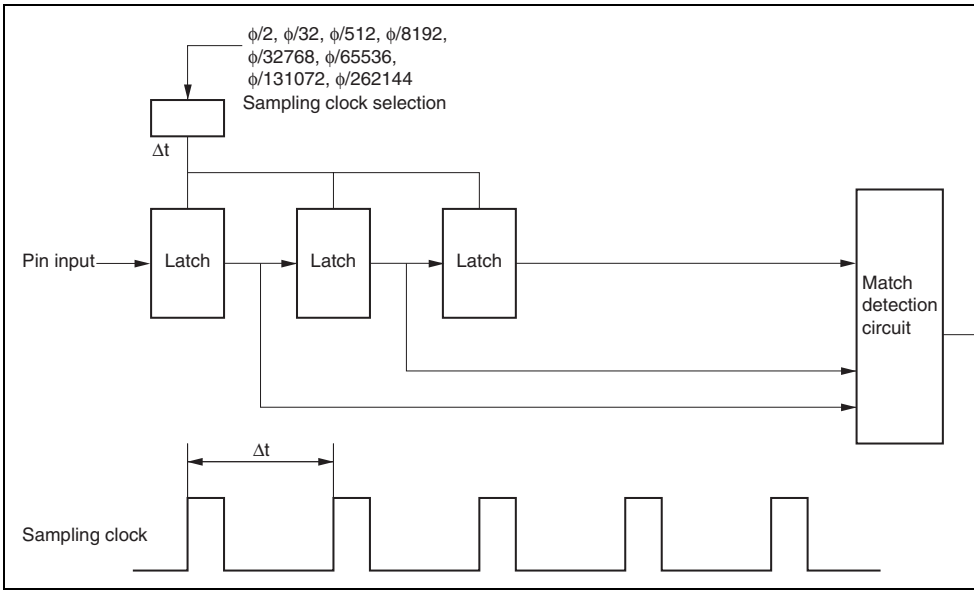


Figure 8.7 Noise Canceler Circuit

(n = 7 to 0)

Figure 8.8 Noise Canceler Operation

(7) Pin Functions

(a) Normal Extended Mode

Port 3 pins are automatically set to function as bidirectional data bus pins.

(b) Address-Data Multiplex Extended Mode

The operation is the same as that in single-chip mode.

(c) Single-Chip Mode

The pin function is switched as shown below according to the P3nDDR bit and the P3nN

P3nDDR	0		1
P3nNCE	0	1	X
Pin function	P3n input pin	DBn input	P3n output pin

[Legend] n = 7 to 0, X: Don't care

Single-chip mode (EXPE = 0)	Off	Off	On/Off	On/O
Address-data multiplex extended mode (EXPE = 1, ADMXE = 1)				

[Legend]

Off: Always off.

On/Off: On when input state and P3PCR = 1; otherwise off.

- Noise canceler mode control register (P4BNCMC)
- Noise cancel cycle setting register (NCCS)

(1) Port 4 Data Direction Register (P4DDR)

The individual bits of P4DDR specify input or output for the port 4 pins. P4DDR is initi
only by a system reset, and retains the value even if an internal reset signal of the WDT
generated.

Bit	Bit Name	Initial Value	R/W	Description
7	P47DDR	0	W	• Normal extended mode (ADMXE = 0) When set to 1, the corresponding pins fun address output pins; when cleared to 0, fu input port pins. The address output pins used are in acco settings of the IOSE and CS256E bits of S • Address-data multiplex extended mode (A 1) These bits correspond to the AD15 to AD the address-data multiplex bus. • Single-chip mode When set to 1, the corresponding pins fun output port pins; when cleared to 0, functi port pins.
6	P46DDR	0	W	
5	P45DDR	0	W	
4	P44DDR	0	W	

P4DR stores output data for the port 4 pins. P4DR is initialized only by a system reset, an the value even if an internal reset signal of the WDT is generated.

Bit	Bit Name	Initial Value	R/W	Description
7	P47DR	0	R/W	These bits store output data for the port 4 pins used as the general output port. If this register is read, the P4DR values are rea bits with the corresponding P4DDR bits set to bits with the corresponding P4DDR bits cleare the pin states are read.
6	P46DR	0	R/W	
5	P45DR	0	R/W	
4	P44DR	0	R/W	
3	P43DR	0	R/W	• Normal extended mode (16-bit data bus) Since the corresponding pins function as bidirectional data bus pins, the value in the has no effect on operation. If this register is read, the P4DR values are the bits with the corresponding P4DDR bits For the bits with the corresponding P4DDR cleared to 0, 1 is read. • Other modes These bits store output data for the port 4 p are used as the general output port. If this register is read, the P4DR values are the bits with the corresponding P4DDR bits For the bits with the corresponding P4DDR cleared to 0, the pin states are read.
2	P42DR	0	R/W	
1	P41DR	0	R/W	
0	P40DR	0	R/W	

2	P42PCR	0	R/W	
1	P41PCR	0	R/W	
0	P40PCR	0	R/W	

(4) Noise Canceler Enable Register (P4BNCE)

The individual bits of P4BNCE enable or disable the noise canceler circuits for ports 4 a

Bit	Bit Name	Initial Value	R/W	Description
7	P47NCE	0	R/W	Enables the noise canceler circuit for the corr
6	P46NCE	0	R/W	pin and the pin state is fetched into P4DR at t
5	P45NCE	0	R/W	sampling cycle set by NCCS.
4	P44NCE	0	R/W	The operation changes according to the other bits. See section 8.2.4 (7), Pin Functions, for
3 to 0	PB3NCE to PB0NCE	All 0	R/W	Bits for port B setting

	to PB0NCMC			

(6) Noise Canceler Cycle Setting Register (NCCS)

NCCS controls the sampling cycle of the noise cancelers.

Bit	Bit Name	Initial Value	R/W	Description
7 to 3	—	Undefined	R/W	Reserved Undefined value is read from these bits.
2	NCCK2	0	R/W	These bits set the sampling cycle of the noise cancelers.
1	NCCK1	0	R/W	
0	NCCK0	0	R/W	• When ϕ = 34 MHz 000: 0.06 µs ϕ/2 100: 963.8 µs ϕ 001: 0.94 µs ϕ/32 101: 1.9 ms ϕ 010: 15.1 µs ϕ/512 110: 3.9 ms ϕ 011: 240.9 µs ϕ/8192 111: 7.7 ms ϕ

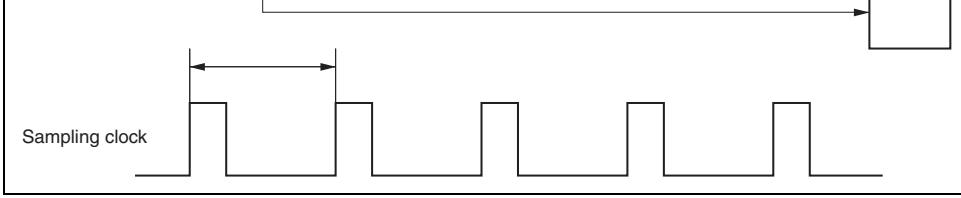


Figure 8.9 Noise Canceler Circuit

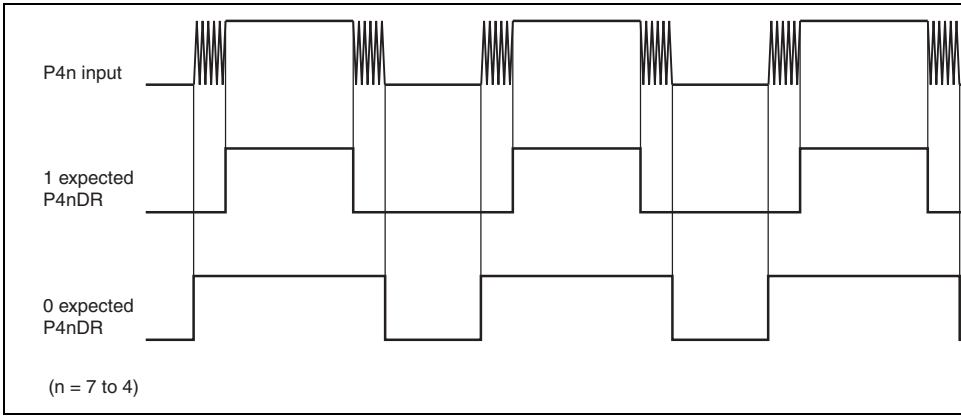


Figure 8.10 Noise Canceler Operation

P4nDDR	0	1	
Address 13	X	0	1
Pin function	P4n input pin	Am output pin	P4n outp

[Legend] n = 15 to 12, n = 7 to 4, X: Don't care

- P43 to P40

Port pins 43 to 40 function as bidirectional data bus pins in 16-bit bus extension, and can as general I/O port pins in 8-bit bus extension.

(b) Address-Data Multiplex Extended Mode

Port pins 47 to 44 are automatically set to function as address bus pins. Port pins 43 to 40 used as general I/O port pins.

P4nNCE	0	1	X
Pin function	P4n input	DBn input	P4n output p
	$\overline{\text{IRQn}}$ input	$\overline{\text{IRQn}}$ input (with the noise canceler)	

[Legend] n = 7 to 4, X: Don't care

- P44 to P40

The pin function is switched as shown below according to the P4nDDR bit. When the IS ISSR is cleared to 0 and the IRQnE bit in IER of the interrupt controller is set to 1, the p used as the $\overline{\text{IRQn}}$ input pin. To use as the $\overline{\text{IRQn}}$ input pin, clear the P4nDDR bit to 0.

P4nDDR	0	1
Pin function	P4n input pin	P4n output pin
	$\overline{\text{IRQn}}$ input pin	

[Legend] n = 3 to 0, X: Don't care

Single-chip mode (EXPE = 0)	Off	Off	On/Off	On/O
Address-data multiplex extended mode (EXPE = 1, ADMXE = 1)				

[Legend]

Off: Always off.

On/Off: On when input state and P4PCR = 1; otherwise off.

The individual bits of P5DDR specify input or output for the port 5 pins.

Bit	Bit Name	Initial Value	R/W	Description
7	P57DDR	0	W	If port 5 pins are specified for use as the ge port, the corresponding pins function as out when the P5DDR bits are set to 1, and as ir when cleared to 0.
6	P56DDR	0	W	The corresponding port 5 pin functions as th clock output pin (φ) when this bit is set to 1, the general I/O port when cleared to 0.
5	P55DDR	0	W	If port 5 pins are specified for use as the ge port, the corresponding pins function as out when the P5DDR bits are set to 1, and as ir when cleared to 0.
4	P54DDR	0	W	
3	P53DDR	0	W	
2	P52DDR	0	W	
1	P51DDR	0	W	
0	P50DDR	0	W	

2	P52DR	0	R/W
1	P51DR	0	R/W
0	P50DR	0	R/W

Note: * The initial value is determined in accordance with the pin state of P56.

(3) Pin Functions

(a) Normal Extended Mode and Address-Data Multiplex Extended Mode

Port pin 57 is automatically set to function as a bus control output pin. The functions of p
56 to 50 are the same as those in single-chip mode.

(b) Single-Chip Mode

Port 5 pins can operate as the SCIF, SCI_1, and SSU input/output, noise canceler input, o
I/O port pins. The relationship between register setting values and pin functions are as fol

- P57

  The pin function is switched as shown below according to the P57DDR bit.

P57DDR	0	1
Pin function	P57 input pin	P57 output pin

The pin function is switched as shown below according to the RE bit in SSER of the
the P55DDR bit. When the ISS13 bit in ISSR16 is cleared to 0 and the IRQ13E bit i
the interrupt controller is set to 1, this pin can be used as the $\overline{\text{IRQ13}}$ input pin. To use
$\overline{\text{IRQ13}}$ input pin, clear the P55DDR bit to 0.

RE	0		1
P55DDR	0	1	X
Pin function	P55 input pin	P55 output pin	SSI input
	$\overline{\text{IRQ13}}$ input pin		

[Legend] X: Don't care.

- P54/$\overline{\text{IRQ12}}$/SSO

  The pin function is switched as shown below according to the TE bit in SSER of the
  the P54DDR bit. When the ISS12 bit in ISSR16 is cleared to 0 and the IRQ12E bit i
  the interrupt controller is set to 1, this pin can be used as the $\overline{\text{IRQ12}}$ input pin. To use
  $\overline{\text{IRQ12}}$ input pin, clear the P54DDR bit to 0.

TE	0		1
P54DDR	0	1	X
Pin function	P54 input pin	P54 output pin	SSO output
	$\overline{\text{IRQ12}}$ input pin		

[Legend] X: Don't care.

	IRQ11 input pin		

[Legend] X: Don't care.

- P52/$\overline{\text{IRQ10}}$/TxD1

  The pin function is switched as shown below according to the combination of the TE
  SCR of SCI_1 and the P52DDR bit.

  When the ISS10 bit in ISSR16 is cleared to 0 and the IRQ10E bit in IER16 of the inte
  controller is set to 1, this pin can be used as the $\overline{\text{IRQ10}}$ input pin. To use as the $\overline{\text{IRQ10}}$
  pin, clear the P52DDR bit to 0.

TE	0		1
P52DDR	0	1	X
Pin function	P52 input pin	P52 output pin	TxD1 output p
	$\overline{\text{IRQ10}}$ input pin		

[Legend] X: Don't care.

	IRQ9 input pin		

[Legend] X: Don't care.

- P50/$\overline{\text{IRQ8}}$/TxDF

  The pin function is switched as shown below according to the combination of the enable/disable setting of the SCIF and the P50DDR bit.

  When the ISS8 bit in ISSR16 is cleared to 0 and the IRQ8E bit in IER16 of the inter controller is set to 1, this pin can be used as the $\overline{\text{IRQ8}}$ input pin. To use as the $\overline{\text{IRQ8}}$ i clear the P50DDR bit to 0.

SCIF	Disabled		Enabled
P50DDR	0	1	X
Pin function	P50 input pin	P50 output pin	TxDF output
	$\overline{\text{IRQ8}}$ input pin		

[Legend] X: Don't care.

(1) Port 6 Data Direction Register (P6DDR)

The individual bits of P6DDR specify input or output for the pins of port 6.

Bit	Bit Name	Initial Value	R/W	Description
7	P67DDR	0	W	If port 6 pins are specified for use as the gen port, the corresponding pins function as outp when the P6DDR bits are set to 1, and as inp when cleared to 0.
6	P66DDR	0	W	
5	P65DDR	0	W	
4	P64DDR	0	W	
3	P63DDR	0	W	• Normal extended mode (16-bit bus) These bits have no effect on operation.
2	P62DDR	0	W	
1	P61DDR	0	W	• Other modes If port 6 pins are specified for use as the I/O port, the corresponding pins function port when the P6DDR bits are set to 1, ar input port when cleared to 0.
0	P60DDR	0	W	

3	P63DR	0	R/W	• Normal extended mode (16-bit data bus
2	P62DR	0	R/W	Since the corresponding pins function as bidirectional data bus pins, the value in t has no effect on operation. If this register is read, the P6DR values for the bits with the corresponding P6DD to 1. For the bits with the corresponding bits cleared to 0, 1 is read.
1	P61DR	0	R/W	
0	P60DR	0	R/W	• Other modes These bits store output data for the port that are used as the general output port If this register is read, the P6DR values for the bits with the corresponding P6DD to 1. For the bits with the corresponding bits cleared to 0, the pin states are read

2	P62PCR	0	R/W
1	P61PCR	0	R/W
0	P60PCR	0	R/W

(4) Pin Functions

(a) Normal Extended Mode

- 16-bit bus mode
  Port pins 63 to 60 are automatically set to function as bidirectional data bus pins.
- 8-bit bus mode
  The operation is the same as that in single-chip mode.

(b) Address-Data Multiplex Extended Mode

The operation is the same as that in single-chip mode.

P67DDR	0	1	X
Pin function	P67 input pin	P67 output pin	SSCK I/O
	$\overline{\text{ExIRQ8}}$ input pin		

[Legend] X: Don't care.

- P66/$\overline{\text{ExIRQ9}}$/$\overline{\text{SCS}}$

  The pin function is switched as shown below according to the CSS1 and CSS0 bits i of the SSU and the P66DDR bit. When the ISS9 bit in ISSR16 is set to 1, this pin ca as the $\overline{\text{ExIRQ9}}$ input pin. To use as the $\overline{\text{ExIRQ9}}$ input pin, clear the P66DDR bit to 0

CSS1, CSS0	00		01 or 1X
P66DDR	0	1	X
Pin function	P66 input pin	P66 output pin	$\overline{\text{SCS}}$ I/O p
	$\overline{\text{ExIRQ9}}$ input pin		

[Legend] X: Don't care.

[Legend] X: Don't care.

- P64/$\overline{\text{ExIRQ11}}$/$\overline{\text{CTS}}$

  The pin function is switched as shown below according to the combination of the enable/disable setting of the SCIF and the P64DDR bit.

  When the ISS10 bit in ISSR16 is set to 1, this pin can be used as the $\overline{\text{ExIRQ11}}$ input p use as the $\overline{\text{ExIRQ11}}$ input pin, clear the P64DDR bit to 0.

SCIF	Disabled		Enabled
P64DDR	0	1	X
Pin function	P64 input pin	P64 output pin	$\overline{\text{CTS}}$ input pi
	$\overline{\text{IRQ11}}$ input pin		

[Legend] X: Don't care.

- P63/PWX3

  The pin function is switched as shown below according to the combination of the OE DACR and the PWMXS bit in PTCNT0 of PWMX_1 and the P63DDR bit.

P63DDR	0		1		X
PWMXS	0	1	0	1	0
OEB	0	X	0	X	1
Pin function	P63 input pin		P63 output pin		PWX3 outpu

[Legend] X: Don't care.

- P61/$\overline{\text{IRQ15}}$/PWX1

  The pin function is switched as shown below according to the combination of the OE DACR and the PWMXS bit in PTCNT0 of PWMX_0 and the P61DDR bit. To use t the $\overline{\text{IRQ15}}$ input pin, clear the P61DDR bit to 0.

P61DDR	0		1		X
PWMXS	0	1	0	1	0
OEB	0	X	0	X	1
Pin function	P61 input pin		P61 output pin		PWX1 outp
	$\overline{\text{IRQ15}}$ input pin				

[Legend] X: Don't care.

- P60/$\overline{\text{IRQ14}}$/PWX1

  The pin function is switched as shown below according to the combination of the OE DACR and the PWMXS bit in PTCNT0 of PWMX_0 and the P60DDR bit. To use t the $\overline{\text{IRQ14}}$ input pin, clear the P60DDR bit to 0.

P60DDR	0		1		X
PWMXS	0	1	0	1	0
OEA	0	X	0	X	1
Pin function	P60 input pin		P60 output pin		PWX0 outp
	$\overline{\text{IRQ14}}$ input pin				

[Legend] X: Don't care.

On/Off: On when input state, P6DDR = 0, and P6PCR = 1, otherwise off.

Bit	Bit Name	Initial Value	R/W	Description
7	P77PIN	Undefined*	R	When this register is read, the pin states are
6	P76PIN	Undefined*	R	Since this register is allocated to the same a PBDDR, writing to this register writes data t and the port B setting is changed.
5	P75PIN	Undefined*	R	
4	P74PIN	Undefined*	R	
3	P73PIN	Undefined*	R	
2	P72PIN	Undefined*	R	
1	P71PIN	Undefined*	R	
0	P70PIN	Undefined*	R	

Note: * The initial values are determined in accordance with the pin states of P77 to P

- P76/AN6

  The pin function is switched as shown below according to the combination of the SCA
  in ADCR and the CH2 to CH0 bits in ADCSR of the A/D converter. Do not set these
  other values than those shown in the following table.

SCANE	0		1	
CH2 to CH0	B'110	Other than B'110	B'110 to B'111	B'000 to B'
Pin function	AN6 input pin	P76 input pin	AN6 input pin	P76 input

- P75/AN5

  The pin function is switched as shown below according to the combination of the SCA
  in ADCR and the CH2 to CH0 bits in ADCSR of the A/D converter. Do not set these
  other values than those shown in the following table.

SCANE	0		1	
CH2 to CH0	B'101	Other than B'101	B'101 to B'111	B'000 to B'
Pin function	AN5 input pin	P75 input pin	AN5 input pin	P75 input

The pin function is switched as shown below according to the combination of the SC
SCANE bits in ADCR and the CH2 to CH0 bits in ADCSR of the A/D converter. Do
these bits to other values than those shown in the following table.

SCANE	0		1			
SCANS	X		0		1	
CH2 to CH0	B'011	Other than B'011	B'011	Other than B'011	B'011 to B'111	B'0
Pin function	AN3 input pin	P73 input pin	AN3 input pin	P73 input pin	AN3 input pin	P7

[Legend] X: Don't care.

- P72/AN2

  The pin function is switched as shown below according to the combination of the SC
  SCANE bits in ADCR and the CH2 to CH0 bits in ADCSR of the A/D converter. Do
  these bits to other values than those shown in the following table.

SCANE	0		1			
SCANS	X		0		1	
CH2 to CH0	B'010	Other than B'010	B'010 to B'011	Other than B'010 to B'011	B'010 to B'111	
Pin function	AN2 input pin	P72 input pin	AN2 input pin	P72 input pin	AN2 input pin	P7

[Legend] X: Don't care.

[Legend] X: Don't care.

- P70/AN0

  The pin function is switched as shown below according to the combination of the SCA
  SCANS bits in ADCR and the CH2 to CH0 bits in ADCSR of the A/D converter. Do
  these bits to other values than those shown in the following table.

SCANE	0		1		
SCANS	X		0		1
CH2 to CH0	B'000	Other than B'000	B'000 to B'011	Other than B'000 to B'011	B'000 to
Pin function	AN0 input pin	P70 input pin	AN0 input pin	P70 input pin	AN0 inp

[Legend] X: Don't care.

The individual bits of P8DDR specify input or output for the port 8 pins.

Bit	Bit Name	Initial Value	R/W	Description
7	P87DDR	0	W	If port 8 pins are specified for use as the ge port, the corresponding pins function as out when the P8DDR bits are set to 1, and as in when cleared to 0.
6	P86DDR	0	W	
5	P85DDR	0	W	
4	P84DDR	0	W	Since this register is allocated to the same a PBPIN, states of the port 8 pins are when th is read.
3	P83DDR	0	W	
2	P82DDR	0	W	
1	P81DDR	0	W	
0	P80DDR	0	W	

2	P82DR	0	R/W
1	P81DR	0	R/W
0	P80DR	0	R/W

(3) Pin Functions

The relationship between register setting values and pin functions are as follows.

- P87/$\overline{\text{ExIRQ15}}$/TxD3/$\overline{\text{ADTRG}}$

  The pin function is switched as shown below according to the combination of the TE
  SCR of SCI_3, the SMIF bit in SCMR, and the P87DDR bit.

  When the TRGS1 and EXTRGS bits are both set to 1 and the TRGS0 bit is cleared to
  ADCR of the A/D converter, this pin can be used as the $\overline{\text{ADTRG}}$ input pin.

  When the ISS15 bit in ISSR16 is set to 1, this pin can be used as the $\overline{\text{ExIRQ15}}$ input p
  use this pin as the $\overline{\text{ExIRQ15}}$ input pin, clear the P87DDR bit to 0.

P87DDR	0		1		
SMIF	0	1	0	1	0
TE	0	X	0	X	1
Pin function	P87 input pin		P87 output pin		TxD3 outp
	$\overline{\text{ExIRQ15}}$ input pin/ $\overline{\text{ADTRG}}$ input pin				

[Legend] X: Don't care.

	ExIRQ14 input pin			

- P85/ExIRQ13/SCK1

  The pin function is switched as shown below according to the combination of the C/
  SMR of SCI_1, the CKE1 and CKE0 bits in SCR, and the P85DDR bit.

  When the ISS13 bit in ISSR16 is set to 1, this pin can be used as the ExIRQ13 input
  use this pin as the ExIRQ13 input pin, clear the P85DDR bit to 0.

CKE1	0				
C/A	0			1	
CKE0	0		1	X	
P85DDR	0	1	X	X	
Pin function	P85 input pin	P85 output pin	SCK1 output pin	SCK1 output pin	SC
	ExIRQ13 input pin				

[Legend] X: Don't care.

Pin function	P84 input pin	P84 output pin	SCK3 output pin	SCK3 output pin	SCI
	$\overline{\text{ExIRQ12}}$ input pin				

[Legend] X: Don't care.

- P83/SDA1

  The pin function is switched as shown below according to the combination of the ICE ICCR of IIC_1 and the P83DDR bit.

  When this pin is used as the P83 output pin, the output format is NMOS push-pull out output format for SDA1 is NMOS open-drain output, which allows direct bus drive.

ICE	0		1
P83DDR	0	1	X
Pin function	P83 input pin	P83 output pin	SDA1 input/ou

[Legend] X: Don't care.

[Legend] X: Don't care.

- P81/SDA0

  The pin function is switched as shown below according to the combination of the IC
  ICCR of IIC_0 and the P81DDR bit.

  When this pin is used as the P81 output pin, the output format is NMOS push-pull ou
  output format for SDA0 is NMOS open-drain output, which allows direct bus drive.

ICE	0		1
P81DDR	0	1	X
Pin function	P81 input pin	P81 output pin	SDA0 input/o

[Legend] X: Don't care.

- P80/SCL0

  The pin function is switched as shown below according to the combination of the IC
  ICCR of IIC_0 and the P80DDR bit.

  When this pin is used as the P80 output pin, the output format is NMOS push-pull ou
  output format for SCL0 is NMOS open-drain output, which allows direct bus drive.

ICE	0		1
P80DDR	0	1	X
Pin function	P80 input pin	P80 output pin	SCL0 input/o

[Legend] X: Don't care.

Bit	Bit Name	Initial Value	R/W	Description
7	P97DDR	0	W	If port 9 pins are specified for use as the gen
6	P96DDR	0	W	port, the corresponding pins function as outp
5	P95DDR	0	W	when the P9DDR bits are set to 1, and as inp
4	P94DDR	0	W	when cleared to 0.
3	P93DDR	0	W	
2	P92DDR	0	W	
1	P91DDR	0	W	
0	P90DDR	0	W	

2	P92DR	0	R/W
1	P91DR	0	R/W
0	P90DR	0	R/W

(3) Pin Functions

The relationship between register setting values and pin functions are as follows.

- P97/$\overline{\text{WAIT}}$/$\overline{\text{CS256}}$

  The pin function is switched as shown below according to the operating mode and th
  combination of the CS256E bit in SYSCR, the WMS1 bit in WSCR and the P97DDI

Operating mode	Extended mode				Single-chip	
WMS1	0			1	X	
CS256E	0		1	X	X	
P97DDR	0	1	X	X	0	
Pin function	P97 input pin	P97 output pin	$\overline{\text{CS256}}$ output pin	$\overline{\text{WAIT}}$ input pin	P97 input pin	P

[Legend] X: Don't care.

Operating mode	Extended mode		Single-chip mode	
P95DDR	X		0	1
IOSE	0	1	X	X
Pin function	$\overline{AS}$ output pin	$\overline{IOS}$ output pin	P95 input pin	P95 ou

[Legend] X: Don't care.

- P94/ExPWX1

  The pin function is switched as shown below according to the combination of the OE
  DACR and the PWMXS bit in PTCNT0 of the PWMX_0 module and the P94DDR bi

P94DDR	0		1		X
PWMXS	0	1	0	1	1
OEB	X	0	X	0	1
Pin function	P94 input pin		P94 output pin		ExPWX1 o

[Legend] X: Don't care.

- P93/ExPWX0

  The pin function is switched as shown below according to the combination of the OE
  DACR and the PWMXS bit in PTCNT0 of the PWMX_0 module and the P93DDR bi

P93DDR	0		1		X
PWMXS	0	1	0	1	1
OEA	X	0	X	0	1
Pin function	P93 input pin		P93 output pin		ExPWX0 o

[Legend] X: Don't care.

- P91/$\overline{AH}$

  The pin function is switched as shown below according to the operating mode, the A
  bit in SYSCR2, and the P91DDR bit.

Operating mode	Extended mode			Single-chip m	
ADMXE	0		1	X	
P91DDR	0	1	X	0	
Pin function	P91 input pin	P91 output pin	$\overline{AH}$ output pin	P91 input pin	P91

[Legend] X: Don't care.

- P90/$\overline{LBE}$

  The pin function is switched as shown below according to the operating mode, the C
  PTCNT0, and the P90DDR bit.

Operating mode	Extended mode			Single-chip m	
OBE	0		1	X	
P90DDR	0	1	X	0	
Pin function	P90 input pin	P90 output pin	$\overline{LBE}$ output pin	P90 input pin	P90

[Legend] X: Don't care.

(2) Port A Data Direction Register (PADDR)

The individual bits of PADDR specify input or output for the port A pins.

Bit	Bit Name	Initial Value	R/W	Description
7	PA7DDR	0	W	When set to 1, the corresponding pins functio output port pins; when cleared to 0, function port pins.
6	PA6DDR	0	W	
5	PA5DDR	0	W	As the address of this register is the same as PAPIN, reading from this register indicates th of port A.
4	PA4DDR	0	W	
3	PA3DDR	0	W	
2	PA2DDR	0	W	
1	PA1DDR	0	W	
0	PA0DDR	0	W	

2	PA2ODR	0	R/W	
1	PA1ODR	0	R/W	
0	PA0ODR	0	R/W	

(3) Port A Input Data Register (PAPIN)

PAPIN indicates the states of the port A pins.

Bit	Bit Name	Initial Value	R/W	Description
7	PA7PIN	Undefined*	R	Pin states are read from this register.
6	PA6PIN	Undefined*	R	As the address of this register is the same a
5	PA5PIN	Undefined*	R	PADDR, writing to this register changes the
4	PA4PIN	Undefined*	R	of port A, that have been written to PADDR.
3	PA3PIN	Undefined*	R	
2	PA2PIN	Undefined*	R	
1	PA1PIN	Undefined*	R	
0	PA0PIN	Undefined*	R	

Note: * The initial values are determined in accordance with the pin states of PA7 to P

Address 18 = 1: $\overline{\text{ADFULLE}}$

Address 13 = 1: $\overline{\text{ADFULLE}}$ • $\overline{\text{CS256E}}$ • IOSE

- PA7/$\overline{\text{ExIRQ7}}$/EVENT7/A23/EXOUT

  The pin function is switched as shown below according to the setting of address 18 an
  PA7DDR bit.

  Setting the ISS7 bit in ISSR makes the pin to function as the $\overline{\text{ExIRQ7}}$ input pin.

  When using the pin as the $\overline{\text{ExIRQ7}}$ input or an EVENT input pin, clear the PA7DDR
  Though the settings for the EVENT input pin have been made, set the PA7DDR bit to
  using the pin as the PA7 or A23 output pin.

  When the module stop mode is cleared in both the EtherC and E-DMAC, this pin func
  the EXOUT output pin.

PA7DDR	0	1	1
Address 18	X	1	0
Pin function	PA7 input pin	PA7 output pin	A23 out
	$\overline{\text{ExIRQ7}}$ input pin/EVENT7 input pin		

[Legend] X: Don't care.

PA6DDR	0	1	
Address 18	1		
Pin function	PA6 input pin	PA6 output pin	A22 ou
	$\overline{\text{ExIRQ6}}$ input pin/EVENT6 input pin		

- PA5/$\overline{\text{ExIRQ5}}$/EVENT5/A21/WOL

  The pin function is switched as shown below according to the setting of the MPDE b ECMR in EtherC, the address 18, and the PA5DDR bit.

  Setting the ISS5 bit in ISSR to 1 makes the pin function as the ExIRQ5 input pin.

  When using the pin as the ExIRQ5 input, or an EVENT input pin, clear the PA5DDI Though the settings for the EVENT input pin have been made, set the PA5DDR bit t using the pin as the A21 or PA5 output pin.

  When the module stop mode is cleared in both the EtherC and E-DMAC, this pin fu the WOL output pin.

MPDE	0		
PA5DDR	0	1	1
Address 18	X	1	0
Pin function	PA5 input pin	PA5 output pin	A21 out
	$\overline{\text{ExIRQ5}}$ input pin/ EVENT5 input pin		

Pin function	PAn input pin	PAn output pin	Am outp
	$\overline{\text{ExIRQn}}$ input pin/EVENTn input pin		

[Legend]n = 4 to 2, m = 20 to 18, X: Don't care.

- PA1/$\overline{\text{ExIRQ1}}$/EVENT1/A17, PA0/$\overline{\text{ExIRQ0}}$/EVENT0/A16

  The pin function is switched as shown below according to the setting of address 13 an PAnDDR bit.

  Setting the ISSn bit in ISSR makes the pin to function as the $\overline{\text{ExIRQn}}$ input pin.

  When using the pin as the $\overline{\text{ExIRQn}}$ input, clear the PAnDDR bit to 0. When using the EVENT input pin, clear the PAnDDR bit to 0. Though the settings for the EVENT inp have been made, set the PAnDDR bit to 1 when using the pin as the PAn or Am outpu

PAnDDR	0	1	
Address 13	X	1	0
Pin function	PAn input pin	PAn output pin	Am outp
	$\overline{\text{ExIRQn}}$ input pin/EVENTn input pin		

[Legend] n = 1, 0; m = 17, 16, X: Don't care.

When the module stop mode is cleared in both the EtherC and E-DMAC, this pin fu
the EXOUT output pin.

PA7DDR	0	1
Pin function	PA7 input pin	PA7 output pin
	$\overline{\text{ExIRQ7}}$ input pin/EVENT7 input pin	

- PA6/$\overline{\text{ExIRQ6}}$/EVENT6/LNKSTA

  The pin function is switched as shown below according to the PA6DDR bit.

  Setting the ISS6 bit in ISSR makes the pin to function as the $\overline{\text{ExIRQ6}}$ input pin.

  When using this pin as the $\overline{\text{ExIRQ6}}$ input, or EVENT6 input pin, clear the PA6DDR
  Though the settings for the EVENT input pin have been made, set the PA6DDR bit t
  the pin as the PA6 output pin.

  When the module stop mode is cleared in both the EtherC and E-DMAC, this pin fu
  the LNKSTA input pin.

PA6DDR	0	1
Pin function	PA6 input pin	PA6 output pin
	$\overline{\text{ExIRQ6}}$ input pin/EVENT6 input pin	

PA5DDR	0	1
Pin function	PA5 input pin	PA5 output pin
	$\overline{\text{ExIRQ5}}$ input pin/EVENT5 input pin	

- PA4/$\overline{\text{ExIRQ4}}$/EVENT4, PA3/$\overline{\text{ExIRQ3}}$/EVENT3, PA2/$\overline{\text{ExIRQ2}}$/EVENT2, PA1/$\overline{\text{ExIRQ1}}$/EVENT1, PA0/$\overline{\text{ExIRQ0}}$/EVENT0

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

  Setting the ISSn bit in ISSR makes the pin to function as the $\overline{\text{ExIRQn}}$ input pin.

  When using this pin as the $\overline{\text{ExIRQn}}$ input or EVENTn input pin, clear the PAnDDR b
  Though the settings for the EVENT input pin have been made, set the PAnDDR bit to
  the pin as the PAn output pin.

PAnDDR	0	1
Pin function	PAn input pin	PAn output pin
	$\overline{\text{ExIRQn}}$ input pin/EVENTn input pin	

[Legend] n = 4 to 0

(5) Input Pull-Up MOS

Port A has built-in input pull-up MOSs that can be controlled by software. This input pul
MOS can be used in any operating mode, and can be specified as on or off on a bit-by-bit

PAnDDR	0		1
PAnODR	1	0	X
PAn pull-up MOS	ON	OFF	OFF

[Legend] n = 7 to 0, X: Don't care.

Off: Always off

On/Off: On when PADDR = 0 and PAODR = 1; otherwise off.

- Noise canceler mode control register (P4BNCMC)
- Noise cancel cycle setting register (NCCS)

(1) Port B Data Direction Register (PBDDR)

The individual bits of PBDDR specify input or output for the pins of port B.

Bit	Bit Name	Initial Value	R/W	Description
7	PB7DDR	0	W	When set to 1, the corresponding pins function output port pins; when cleared to 0, function as port pins.
6	PB6DDR	0	W	
5	PB5DDR	0	W	
4	PB4DDR	0	W	
3	PB3DDR	0	W	
2	PB2DDR	0	W	
1	PB1DDR	0	W	
0	PB0DDR	0	W	

2	PB2DR	0	R/W
1	PB1DR	0	R/W
0	PB0DR	0	R/W

(3) Port B Input Data Register (PBPIN)

PBPIN indicates the states of the port B pins.

Bit	Bit Name	Initial Value	R/W	Description
7	PB7PIN	Undefined*	R	When this register is read, the pin states are
6	PB6PIN	Undefined*	R	Since this register is allocated to the same a
5	PB5PIN	Undefined*	R	P8DDR, writing to this register writes data t
4	PB4PIN	Undefined*	R	and the port 8 setting is changed.
3	PB3PIN	Undefined*	R	
2	PB2PIN	Undefined*	R	
1	PB1PIN	Undefined*	R	
0	PB0PIN	Undefined*	R	

Note: * The initial values are determined in accordance with the pin states of PB7 to

0	PB0NCE	0	R/W	bits. See section 8.2.11 (7), Pin Functions, for

(5) Noise Canceler Mode Control Register (P4BNCMC)

P4BNCMC controls whether 1 or 0 is expected for the input signal to port 4 and port B in units.

Bit	Bit Name	Initial Value	R/W	Description
7 to 4	P47NCMC to P44NCMC	All 1	R/W	Bits for port 4 setting
3	PB3NCMC	1	R/W	Expected value setting
2	PB2NCMC	1	R/W	1 expected: 1 is stored in the port data register is input stably.
1	PB1NCMC	1	R/W	
0	PB0NCMC	1	R/W	0 expected: 0 is stored in the port data register is input stably.

001: 0.94 µs	ϕ/32	101: 1.9 ms
010: 15.1 µs	ϕ/512	110: 3.9 ms
011: 240.9 µs	ϕ/8192	111: 7.7 ms

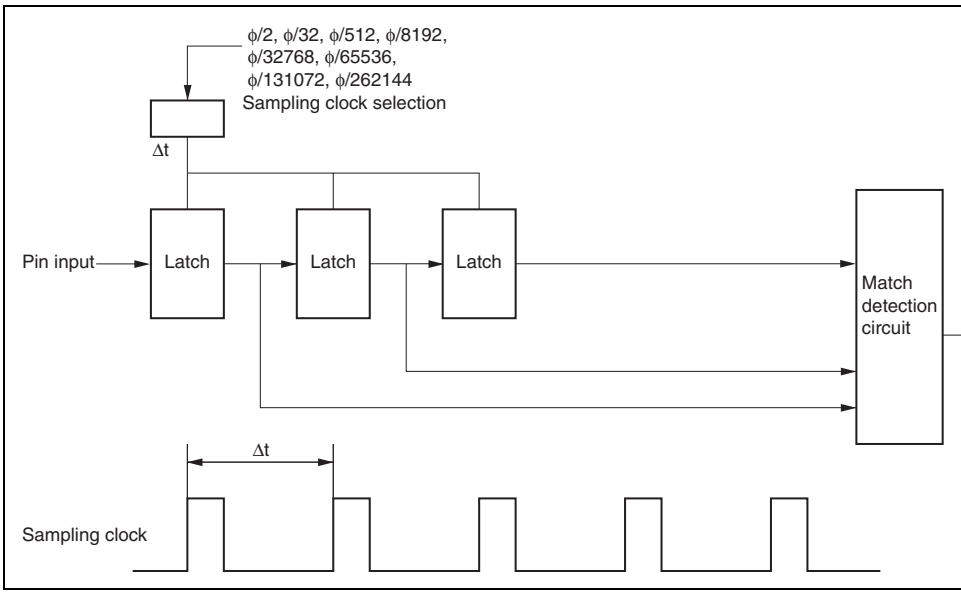


Figure 8.11 Noise Canceler Circuit

(n = 3 to 0)

Figure 8.12 Noise Canceler Operation

(7) Pin Functions

- PB7/EVENT15/RM_RX-ER, PB6/EVENT14/RM_CRS-DV, PB5/EVENT13/RM_R
  PB4/EVENT12/RM_TX-EN

  The pin function is switched as shown below according to the PBnDDR bit. When usi
  pin as the EVENT input pin, clear the PBnDDR bit to 0. These pins can be used as Et
  pins when the EtherC is enabled.

EtherC, E-DMAC	Either of them is stopped			Both of the stopp
PBnDDR	0		1	X
Event counter	Disabled	Enabled	X	X
Pin function	PBn input pin	EVENTm input pin	PBn output pin	RM_xx EtherC I/

[Legend] n = 7 to 4, m = 15 to 8, X: Don't care.

Note: * See section 7.3, DTC Event Counter, for the event counter settings.

Pin function	PBn input	DBn input	EVENTm input	PBn output pin	RM_X EtherC

[Legend] n = 3 to 0, m = 11 to 8, X: Don't care.

The individual bits of PCDDR specify input or output for the port C pins.

Bit	Bit Name	Initial Value	R/W	Description
7	PC7DDR	0	W	When set to 1, the corresponding pins functio output port pins; when cleared to 0, function port pins. Since this register is allocated to the same ad PCPIN, states of the port C pins are returned this register is read.
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	PC2ODR	0	R/W	
1	PC1ODR	0	R/W	
0	PC0ODR	0	R/W	

(3) Port C Input Data Register (PCPIN)

PCPIN indicates the pin states of port C.

Bit	Bit Name	Initial Value	R/W	Description
7	PC7PIN	Undefined*	R	When this register is read, the pin states are
6	PC6 PIN	Undefined*	R	Since this register is allocated to the same a
5	PC5PIN	Undefined*	R	PCDDR, writing to this register writes data t
4	PC4 PIN	Undefined*	R	and the port C setting is changed.
3	PC3 PIN	Undefined*	R	
2	PC2 PIN	Undefined*	R	
1	PC1 PIN	Undefined*	R	
0	PC0 PIN	Undefined*	R	

Note: The initial values are determined in accordance with the states of PC7 to PC0 pin

bus width, the pin function is the same as that in single-chip mode.

- PC5 to PC0

  The pin functions are the same as those in single-chip mode.

The pin function is switched as shown below according to the combination of the IC
ICCR of the IIC_4 and the PC5DDR bit.

ICE	0		1
PC5DDR	0	1	X
Pin function	PC5 input pin	PC5 output pin	SDA4 input/ou

[Legend] X: Don't care.

- PC4/SCL4

  The pin function is switched as shown below according to the combination of the IC
  ICCR of the IIC_4 and the PC4DDR bit.

ICE	0		1
PC4DDR	0	1	X
Pin function	PC4 input pin	PC4 output pin	SCL4 input/ou

[Legend] X: Don't care.

- PC3/SDA3

  The pin function is switched as shown below according to the combination of the IC
  ICCR of the IIC_3 and the PC3DDR bit.

ICE	0		1
PC3DDR	0	1	X
Pin function	PC3 input pin	PC3 output pin	SDA3 input/ou

[Legend] X: Don't care.

The pin function is switched as shown below according to the combination of the ICE
ICCR of the IIC_2 and the PC1DDR bit.

ICE	0		1
PC1DDR	0	1	X
Pin function	PC1 input pin	PC1 output pin	SDA2 input/ou

[Legend] X: Don't care.

- PC0/SCL2

  The pin function is switched as shown below according to the combination of the ICE
  ICCR of the IIC_2 and the PC0DDR bit.

ICE	0		1
PC0DDR	0	1	X
Pin function	PC0 input pin	PC0 output pin	SCL2 input/ou

[Legend] X: Don't care.

(2) Port D Data Direction Register (PDDDR)

The individual bits of PDDDR specify input or output for the port D pins.

Bit	Bit Name	Initial Value	R/W	Description
7	PD7DDR	0	W	If port D pins are specified for use as the ge
6	PD6DDR	0	W	port, the corresponding pins function as out
5	PD5DDR	0	W	when the PDDDR bits are set to 1, and as i
4	PD4DDR	0	W	when cleared to 0.
3	PD3DDR	0	W	Since this register is allocated to the same a
2	PD2DDR	0	W	PDPIN, the states of the port D pins are retu
1	PD1DDR	0	W	when this register is read.
0	PD0DDR	0	W	

2	PD2ODR	0	R/W
1	PD1ODR	0	R/W
0	PD0ODR	0	R/W

(3) Port D Input Data Register (PDPIN)

PDPIN indicates the pin states of port D.

Bit	Bit Name	Initial Value	R/W	Description
7	PD7PIN	Undefined*	R	When this register is read, the pin states are
6	PD6PIN	Undefined*	R	Since this register is allocated to the same ad
5	PD5PIN	Undefined*	R	PDDDR, writing to this register writes data to
4	PD4PIN	Undefined*	R	and the port D setting is changed.
3	PD3PIN	Undefined*	R	
2	PD2PIN	Undefined*	R	
1	PD1PIN	Undefined*	R	
0	PD0PIN	Undefined*	R	

Note: The initial values are determined in accordance with the states of PD7 to PD0 pins

ICE	0		1
PD7DDR	0	1	X
Pin function	PD7 input pin	PD7 output pin	SDA5 input/ou

[Legend] X: Don't care.

- PD6/SCL5

  The pin function is switched as shown below according to the combination of the IC ICCR of the IIC_5 and the PD6DDR bit.

ICE	0		1
PD6DDR	0	1	X
Pin function	PD6 input pin	PD6 output pin	SCL5 input/ou

[Legend] X: Don't care.

- PD5/$\overline{\text{LPCPD}}$

  The pin function is switched as shown below according to the PD5DDR bit. This pin used as the $\overline{\text{LPCPD}}$ input pin when the LPC is enabled.

LPC	Disabled		Enable
PD5DDR	0	1	0
Pin function	PD5 input pin	PD5 output pin	$\overline{\text{LPCPD}}$ inp

in HICR0 of the LPC and the PD3DDR bit.

FGA20E	0		1
PD3DDR	0	1	0
Pin function	PD3 input pin	PD3 output pin	GA20 output

- PD2/$\overline{\text{PME}}$

  The pin function is switched as shown below according to the combination of the PM
  HICR0 of the LPC and the PD2DDR bit.

PMEE	0		1
PD2DDR	0	1	0
Pin function	PD2 input pin	PD2 output pin	$\overline{\text{PME}}$ output

- PD1/$\overline{\text{LSMI}}$

  The pin function is switched as shown below according to the combination of the LSM
  HICR0 of the LPC and the PD1DDR bit.

LSMIE	0		1
PD1DDR	0	1	0
Pin function	PD1 input pin	PD1 output pin	$\overline{\text{LSMI}}$ output

Port pins D5 to D0 have built-in input pull-up MOSs that can be controlled by software. pull-up MOS can be used in any operating mode, and can be specified as on or off on a b basis.

PDnDDR	0		1
PDnODR	1	0	X
PDn pull-up MOS	ON	OFF	OFF

[Legend] n = 5 to 0, X: Don't care.

The input pull-up MOS is in the off state after a reset and in hardware standby mode. Th state is retained in software standby mode.

Table 8.16 summarizes the input pull-up MOS states.

Table 8.16 Port D Input Pull-Up MOS States

Reset	Hardware Standby Mode	Software Standby Mode	In Other Operat
Off	Off	On/Off	On/Off

[Legend]

Off: Always off.

On/Off: On when PDDDR = 0 and PDODR = 1; otherwise off.

The individual bits of PEDDR specify input or output for the port E pins.

Bit	Bit Name	Initial Value	R/W	Description
7	PE7DDR	0	W	When set to 1, the corresponding pins functio output port pins; when cleared to 0, function port pins. Since this register is allocated to the same ad PEPIN, states of the port E pins are returned this register is read.
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	PE2ODR	0	R/W	
1	PE1ODR	0	R/W	
0	PE0ODR	0	R/W	

(3) Port E Input Data Register (PEPIN)

PEPIN indicates the pin states of port E.

Bit	Bit Name	Initial Value	R/W	Description
7	PE7PIN	Undefined*	R	When this register is read, the pin states are
6	PE6PIN	Undefined*	R	Since this register is allocated to the same a
5	PE5PIN	Undefined*	R	PEDDR, writing to this register writes data t
4	PE4PIN	Undefined*	R	and the port E setting is changed.
3	PE3PIN	Undefined*	R	
2	PE2PIN	Undefined*	R	
1	PE1PIN	Undefined*	R	
0	PE0PIN	Undefined*	R	

Note: The initial value of these pins is determined in accordance with the state of pins P
PE0.

PE7DDR	0	1	X
Pin function	PE7 input pin	PE7 output pin	SERIRQ input/o

[Legend] X: Don't care.

- PE6/LCLK

  The pin function is switched as shown below according to whether the LPC is enabled disabled and the PE6DDR bit.

LPC	Disabled		Enabled
PE6DDR	0	1	X
Pin function	PE6 input pin	PE6 output pin	LCLK input

[Legend] X: Don't care.

- PE5/$\overline{\text{LRESET}}$

  The pin function is switched as shown below according to whether the LPC is enabled disabled and the PE5DDR bit.

LPC	Disabled		Enabled
PE5DDR	0	1	X
Pin function	PE5 input pin	PE5 output pin	$\overline{\text{LRESET}}$ inpu

[Legend] X: Don't care.

The pin function is switched as shown below according to whether the LPC is enable disabled and the PE3DDR bit.

LPC	Disabled		Enable
PE3DDR	0	1	X
Pin function	PE3 input pin	PE3 output pin	LAD3 input/ou

[Legend] X: Don't care.

- PE2/LAD2

  The pin function is switched as shown below according to whether the LPC is enable disabled and the PE2DDR bit.

LPC	Disabled		Enable
PE2DDR	0	1	X
Pin function	PE2 input pin	PE2 output pin	LAD2 input/ou

[Legend] X: Don't care.

- PE1/LAD1

  The pin function is switched as shown below according to whether the LPC is enable disabled and the PE1DDR bit.

LPC	Disabled		Enable
PE1DDR	0	1	X
Pin function	PE1 input pin	PE1 output pin	LAD1 input/ou

[Legend] X: Don't care.

The individual bits of PFDDR specify input or output for the port F pins. PFDDR is init
only by a system reset, and retains the value even if an internal reset signal of the WDT
generated.

Bit	Bit Name	Initial Value	R/W	Description
7	—	—	—	Reserved
6	PF6DDR	0	W	When set to 1, the corresponding pin functi output port pin; when cleared to 0, functions input port pin. Since this register is allocated to the same a PFPIN, states of the port F pins are returne this register is read.
5 to 2	—	—	—	Reserved
1	PF1DDR	0	W	When set to 1, the corresponding pin functi output port pin; when cleared to 0, functions input port pin. Since this register is allocated to the same a PFPIN, states of the port F pins are returne this register is read.
0	PF0DDR	0	W	

				Undefined values are read from these bits.
1	PF1ODR	0	R/W	Store the output data for the pins that are use
0	PF0ODR	0	R/W	general output port.

(3) Port F Input Data Register (PFPIN)

PFPIN indicates the pin states of port F.

Bit	Bit Name	Initial Value	R/W	Description
7	—	—	—	Reserved Undefined value is read from this bit.
6	PF6PIN	Undefined*	R	When this register is read, the pin states are Since this register is allocated to the same ad PFDDR, writing to this register writes data to and the port F setting is changed.
5 to 2	—	—	—	Reserved Undefined values are read from these bits.
1	PF1PIN	Undefined*	R	When this register is read, the pin states are
0	PF0PIN	Undefined*	R	Since this register is allocated to the same ad PFDDR, writing to this register writes data to and the port F setting is changed.

Note: * The initial value of these pins is determined in accordance with the state of pins PF1, and PF0.

OEA	X	0	X	0	1
Pin function	PF6 input pin		PF6 output pin		ExPWX2 ou

[Legend] X: Don't care.

- PF1/RS9/MDC

  The pin function is switched as shown below according to the combination of the mo
  state in the EtherC and e-DMAC and the PF1DDR bit.

EtherC, E-DMAC	Either of them is stopped		Both of them a
PF1DDR	0	1	X
Pin function	PF1 input pin	PF1 output pin	MDC outp

[Legend] X: Don't care.

- PF0/RS0/MDIO

  The pin function is switched as shown below according to the combination of the mo
  state in the EtherC and e-DMAC and the PF0DDR bit.

EtherC, E-DMAC	Either of them is stopped		Both of them a
PF0DDR	0	1	X
Pin function	PF0 input pin	PF0 output pin	MDIO input/o

[Legend] X: Don't care.

ISSR16 and ISSR select pins for the $\overline{\text{IRQ15}}$ to $\overline{\text{IRQ0}}$ inputs.

- ISSR16

Bit	Bit Name	Initial Value	R/W	Description
15	ISS15	0	R/W	0: P61/$\overline{\text{IRQ15}}$ is selected 1: P87/$\overline{\text{ExIRQ15}}$ is selected
14	ISS14	0	R/W	0: P60/$\overline{\text{IRQ14}}$ is selected 1: P86/$\overline{\text{ExIRQ14}}$ is selected
13	ISS13	0	R/W	0: P55/$\overline{\text{IRQ13}}$ is selected 1: P85/$\overline{\text{ExIRQ13}}$ is selected
12	ISS12	0	R/W	0: P54/$\overline{\text{IRQ12}}$ is selected 1: P84/$\overline{\text{ExIRQ12}}$ is selected
11	ISS11	0	R/W	0: P53/$\overline{\text{IRQ11}}$ is selected 1: P64/$\overline{\text{ExIRQ11}}$ is selected
10	ISS10	0	R/W	0: P52/$\overline{\text{IRQ10}}$ is selected 1: P65/$\overline{\text{ExIRQ10}}$ is selected
9	ISS9	0	R/W	0: P51/$\overline{\text{IRQ9}}$ is selected 1: P66/$\overline{\text{ExIRQ9}}$ is selected
8	ISS8	0	R/W	0: P50/$\overline{\text{IRQ8}}$ is selected 1: P67/$\overline{\text{ExIRQ8}}$ is selected

				1: PA4/$\overline{\text{ExIRQ4}}$ is selected
3	ISS3	0	R/W	0: P43/$\overline{\text{IRQ3}}$ is selected
				1: PA3/$\overline{\text{ExIRQ3}}$ is selected
2	ISS2	0	R/W	0: P42/$\overline{\text{IRQ2}}$ is selected
				1: PA2/$\overline{\text{ExIRQ2}}$ is selected
1	ISS1	0	R/W	0: P41/$\overline{\text{IRQ1}}$ is selected
				1: PA1/$\overline{\text{ExIRQ1}}$ is selected
0	ISS0	0	R/W	0: P40/$\overline{\text{IRQ0}}$ is selected
				1: PA0/$\overline{\text{ExIRQ0}}$ is selected

				with the SCI_3 as the smart card interface 0: TxD3 and RxD3 are not internally conn 1: TxD3 and RxD3 are internally connecte
5, 4	—	All 0	R/W	Reserved The initial value should not be changed.
3	PWMXS	0	R/W	Selects pins for 14-bit PWM timer outputs 0: P60/PWX0, P61/PWX1, P62/PWX2, P6 are selected 1: P93/ExPWX0, P94/ExPWX1, PF6/ExP PF3/ExPWX3* are selected
2	—	0	R/W	Reserved The initial value should not be changed.
1	OBE	0	R/W	Selects glueless extension. 0: Control by $\overline{RD}$, $\overline{HWR}$, $\overline{LWR}$ 1: Control by $\overline{RD}$, $\overline{WR}$, $\overline{HBE}$, $\overline{LBE}$ (glueless extension)
0	—	0	R/W	Reserved The initial value should not be changed.

Note: * The PF3/EXPWX3 pin is available only in the H8S/2472 Group.

- Two base cycle settings
  The base cycle can be set equal to T × 64 or T × 256, where T is the resolution.
- Sixteen operation clocks (by combination of eight resolution settings and two base c
  settings)

Figure 9.1 shows a block diagram of the PWM (D/A) module.

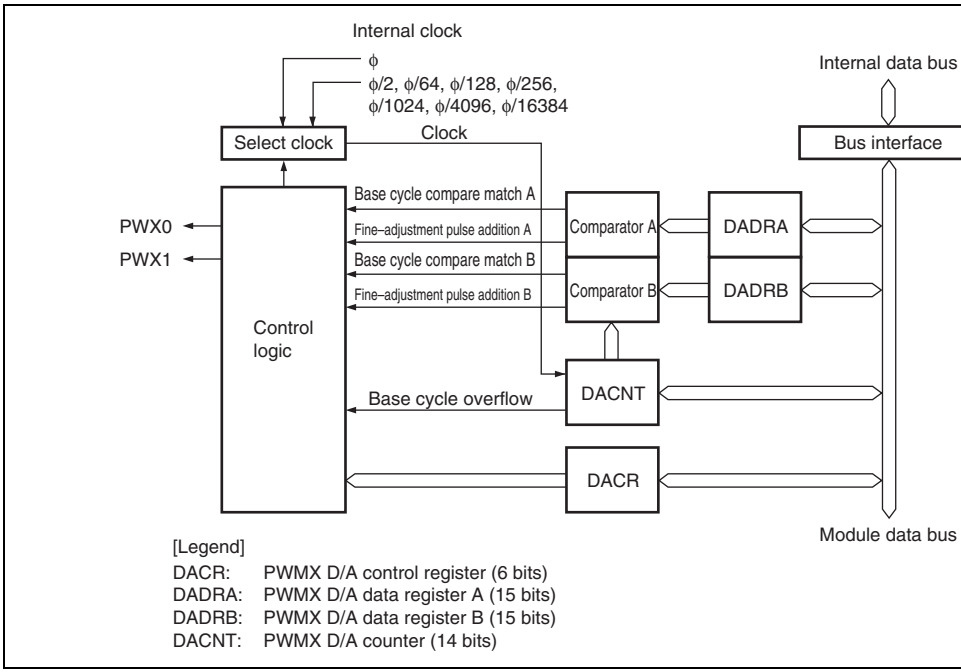


Figure 9.1 PWMX (D/A) Block Diagram

PWMX output pin 3	PWX3	Output	PWM timer pulse output of PWMX_1 ch

9.3 Register Descriptions

The PWMX (D/A) module has the following registers. For details on the module stop co register, see section 28.1.3, Module Stop Control Registers H, L, and A (MSTPCRH, MS MSTPCRA).

- PWMX (D/A) counter (DACNT)
- PWMX (D/A) data register A (DADRA)
- PWMX (D/A) data register B (DADRB)
- PWMX (D/A) control register (DACR)
- Peripheral clock select register (PCSR)

Note: The same addresses are shared by DADRA and DACR, and by DADRB and DA Switching is performed by the REGS bit in DACNT or DADRB.

Bit	Bit Name	Initial Value	R/W	Description
15 to 8	UC7 to UC0	All 0	R/W	Lower Up-Counter
7 to 2	UC8 to UC13	All 0	R/W	Upper Up-Counter
1	—	1	R	Reserved This bit is always read as 1 and cannot be mo
0	REGS	1	R/W	Register Select DADRA and DACR, and DADRB and DACNT located at the same addresses. The REGS bit which registers can be accessed. When chang register to be accessed, set this bit in advance 0: DADRA and DADRB can be accessed 1: DACR and DACNT can be accessed

				These bits set a digital value to be converted to analog value. In each base cycle, the DACNT value is continu compared with the DADR value to determine th cycle of the output waveform, and to decide wh output a fine-adjustment pulse equal in width to resolution. To enable this operation, this registe set within a range that depends on the CFS bit. DADR value is outside this range, the PWM out held constant. A channel can be operated with 12-bit precision DA0 and DA1 to 0. The two data bits are not co with UC12 and UC13 of DACNT.
1	CFS	1	R/W	Carrier Frequency Select 0: Base cycle = resolution (T) × 64 The range of DA13 to DA0: H'0100 to H'3FFF 1: Base cycle = resolution (T) × 256 The range of DA13 to DA0: H'0040 to H'3FFF
0	—	1	R	Reserved This bit is always read as 1 and cannot be modi

				be set within a range that depends on the CF DADR value is outside this range, the PWM c held constant. A channel can be operated with 12-bit precisi fixing DA0 and DA1 to 0. The two data bits ar compared with UC12 and UC13 of DACNT.
1	CFS	1	R/W	Carrier Frequency Select 0: Base cycle = resolution (T) × 64 DA13 to DA0 range = H'0100 to H'3FFF 1: Base cycle = resolution (T) × 256 DA13 to DA0 range = H'0040 to H'3FFF
0	REGS	1	R/W	Register Select DADRA and DACR, and DADRB and DACNT located at the same addresses. The REGS bi which registers can be accessed. When char register to be accessed, set this bit in advanc 0: DADRA and DADRB can be accessed 1: DACR and DACNT can be accessed

				0: DACNT operates as a 14-bit up-counter
				1: DACNT halts at H'0003
5, 4	—	All 1	R	Reserved
				These bits are always read as 1 and cannot be m
3	OEB	0	R/W	Output Enable B
				Enables or disables output on PWMX (D/A) chan
				0: PWMX (D/A) channel B output (at the PWX1, P pins) is disabled
				1: PWMX (D/A) channel B output (at the PWX1, P pins) is enabled
2	OEA	0	R/W	Output Enable A
				Enables or disables output on PWMX (D/A) chan
				0: PWMX (D/A) channel A output (at the PWX0, P pin) is disabled
				1: PWMX (D/A) channel A output (at the PWX0, P pins) is enabled
1	OS	0	R/W	Output Select
				Selects the phase of the PWMX (D/A) output.
				0: Direct PWMX (D/A) output
				1: Inverted PWMX (D/A) output
0	CKS	0	R/W	Clock Select
				Selects the PWMX (D/A) resolution. Eight kinds o resolution can be selected.
				0: Operates at resolution (T) = system clock cycle (t_{cyc})
				1: Operates at resolution (T) = system clock cycle (t_{cyc}) × 2, × 64, × 128, × 256, × 1024, × 4096, an 16384.

4	PWCKX0A	0	R/W	These bits select a clock cycle with the CKS bit of PWMX_0 being 1. See table 9.2.
3	PWCKX1C	0	R/W	PWMX_1 Clock Select This bit selects a clock cycle with the CKS bit of PWMX_1 being 1. See table 9.2.
2	—	0	R/W	Reserved
1	—	0	R/W	The initial value should not be changed.
0	PWCKX0C	0	R/W	PWMX_0 Clock Select This bit selects a clock cycle with the CKS bit of PWMX_0 being 1. See table 9.2.

Table 9.2 Clock Select of PWMX_1 and PWMX_0

PWCKX0C PWCKX1C	PWCKX0B PWCKX1B	PWCKX0A PWCKX1A	Resolution (T)
0	0	0	Operates on the system clock cycle (t_{cyc})
0	0	1	Operates on the system clock cycle (t_{cyc})
0	1	0	Operates on the system clock cycle (t_{cyc})
0	1	1	Operates on the system clock cycle (t_{cyc})
1	0	0	Operates on the system clock cycle (t_{cyc})
1	0	1	Operates on the system clock cycle (t_{cyc})
1	1	0	Operates on the system clock cycle (t_{cyc})
1	1	1	Setting prohibited

combined 16-bit value is written in the register.

- Read

  When the upper byte is read from, the upper-byte value is transferred to the CPU and lower-byte value is transferred to TEMP. Next, when the lower byte is read from, the byte value in TEMP is transferred to the CPU.

These registers should always be accessed 16 bits at a time with a MOV instruction, and t byte should always be accessed before the lower byte. Correct data will not be transferred the upper byte or only the lower byte is accessed. Also note that a bit manipulation instru cannot be used to access these registers.

Example 1: Write to DACNT

```
MOV.W R0, @DACNT      ; Write R0 contents to DACNT
```

Example 2: Read DADRA

```
MOV.W @DADRA, R0      ; Copy contents of DADRA to R0
```

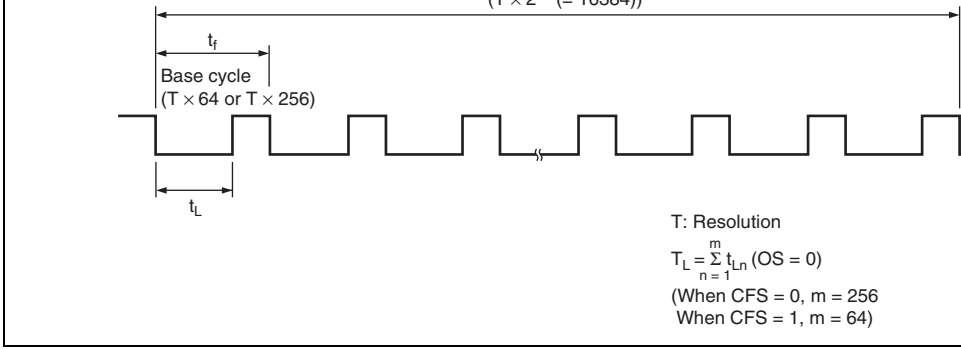


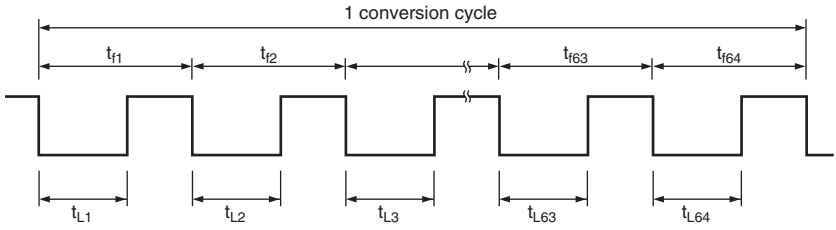
Figure 9.2 PWMX (D/A) Operation

Table 9.3 summarizes the relationships between the CKS and CFS bit settings and the re
base cycle, and conversion cycle. The PWM output remains fixed unless DA13 to DA0
contain at least a certain minimum value. The relationship between the OS bit and the ou
waveform is shown in figures 9.3 and 9.4.

						132.8 kHz		DA13 to 0 = H'0000 to H'003F	12			0	0	
								(Data value) × T						
								DA13 to 0 = H'0040 to H'3FFF	10	0	0	0	0	
0	0	0	1	0.06	0	3.76 µs	0.964 ms	Always low/high output	14					
				(ϕ/2)		265.6 kHz		DA13 to 0 = H'0000 to H'00FF	12			0	0	
								(Data value) × T						
								DA13 to 0 = H'0100 to H'3FFF	10	0	0	0	0	
					1	15.06 µs	0.964 ms	Always low/high output	14					
						66.4 kHz		DA13 to 0 = H'0000 to H'003F	12			0	0	
								(Data value) × T						
								DA13 to 0 = H'0040 to H'3FFF	10	0	0	0	0	
0	0	1	1	1.88	0	120.5 µs	30.840 ms	Always low/high output	14					
				(ϕ/64)		8.3 kHz		DA13 to 0 = H'0000 to H'00FF	12			0	0	
								(Data value) × T						
								DA13 to 0 = H'0100 to H'3FFF	10	0	0	0	0	
					1	481.9 µs	30.840 ms	Always low/high output	14					
						2.1 kHz		DA13 to 0 = H'0000 to H'003F	12			0	0	
								(Data value) × T						
								DA13 to 0 = H'0040 to H'3FFF	10	0	0	0	0	
0	1	0	1	3.76	0	240.9 µs	61.681 ms	Always low/high output	14					
				(ϕ/128)		4.2 kHz		DA13 to 0 = H'0000 to H'00FF	12			0	0	
								(Data value) × T						
								DA13 to 0 = H'0100 to H'3FFF	10	0	0	0	0	
					1	963.8 µs	61.681 ms	Always low/high output	14					
						1.0 kHz		DA13 to 0 = H'0000 to H'003F	12			0	0	
								(Data value) × T						
								DA13 to 0 = H'0040 to H'3FFF	10	0	0	0	0	

								DA13 to 0 = H'0040 to H'3FFF	10	0	0	0	0
1	0	0	1	30.12 (ϕ/1024)	0	1.93 ms 518.8 Hz	493.45 ms	Always low/high output	14				
								DA13 to 0 = H'0000 to H'00FF	12			0	0
								(Data value) × T					
								DA13 to 0 = H'0100 to H'3FFF	10	0	0	0	0
					1	7.71 ms 129.7 Hz	493.45 ms	Always low/high output	14				
								DA13 to 0 = H'0000 to H'003F	12			0	0
								(Data value) × T					
								DA13 to 0 = H'0040 to H'3FFF	10	0	0	0	0
1	0	1	1	120.47 (ϕ/4096)	0	7.71 ms 129.7 Hz	1.974 s	Always low/high output	14				
								DA13 to 0 = H'0000 to H'00FF	12			0	0
								(Data value) × T					
								DA13 to 0 = H'0100 to H'3FFF	10	0	0	0	0
					1	30.84 ms 32.4 Hz	1.974 s	Always low/high output	14				
								DA13 to 0 = H'0000 to H'003F	12			0	0
								(Data value) × T					
								DA13 to 0 = H'0040 to H'3FFF	10	0	0	0	0
1	1	0	1	481.88 (ϕ/16384)	0	30.84 ms 32.4 Hz	7.895 s	Always low/high output	14				
								DA13 to 0 = H'0000 to H'00FF	12			0	0
								(Data value) × T					
								DA13 to 0 = H'0100 to H'3FFF	10	0	0	0	0
					1	123.36 ms 8.1 Hz	7.895 s	Always low/high output	14				
								DA13 to 0 = H'0000 to H'003F					
								(Data value) × T	12			0	0
								DA13 to 0 = H'0040 to H'3FFF					
									10	0	0	0	0
1	1	1	1	Setting prohibited	—	—	—	—	—	—	—	—	—

Note: * Indicates the conversion cycle when specific DA3 to DA0 bits are fixed.



$t_{f1} = t_{f2} = t_{f3} = \cdots = t_{f63} = t_{f64} = T \times 256$
$t_{L1} + t_{L2} + t_{L3} + \cdots + t_{L63} + t_{L64} = T_L$

b. CFS = 1 [base cycle = resolution (T) × 256]

Figure 9.3 Output Waveform (OS = 0, DADR corresponds to T_L)

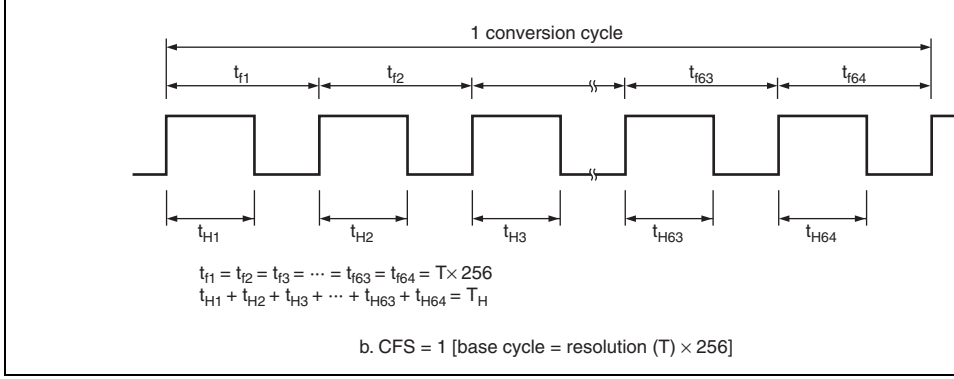


Figure 9.4 Output Waveform (OS = 1, DADR corresponds to T_H)

An example of the additional pulses when CFS = 1 (base cycle = resolution (T) × 256) a
(inverted PWM output) is described below. When CFS = 1, the upper eight bits (DA13 t
DADR determine the duty cycle of the base pulse while the subsequent six bits (DA5 to
determine the locations of the additional pulses as shown in figure 9.5.

Table 9.4 lists the locations of the additional pulses.

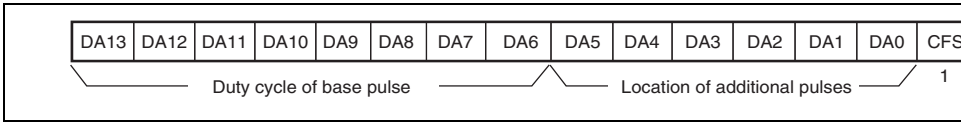


Figure 9.5 D/A Data Register Configuration when CFS = 1

In this example, DADR = H'0207 (B'0000 0010 0000 0111). The output waveform is sh
figure 9.6. Since CFS = 1 and the value of the upper eight bits is B'0000 0010, the high
the base pulse duty cycle is 2/256 × (T).

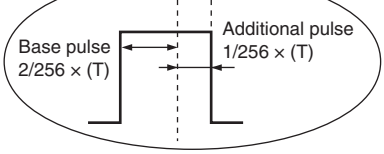


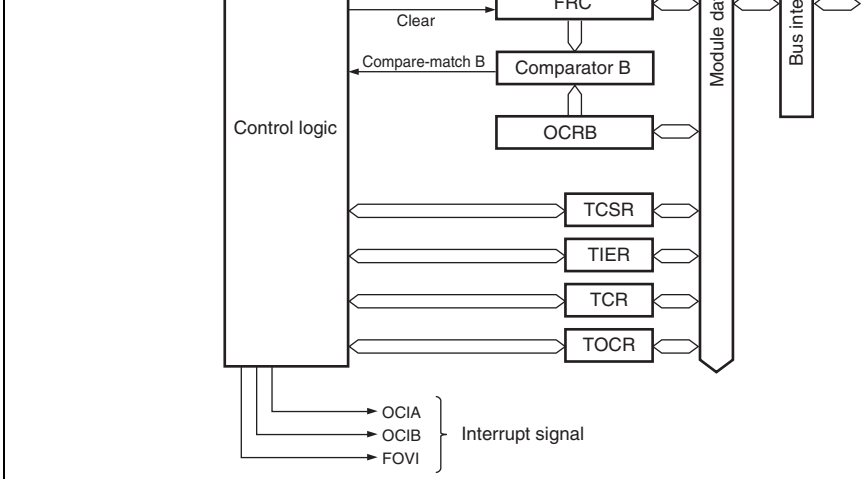
Figure 9.6 Output Waveform when DADR = H'0207 (OS = 1)

However, when CFS = 0 (base cycle = resolution (T) × 64), the duty cycle of the base pul
determined by the upper six bits and the locations of the additional pulses by the subsequ
bits with a method similar to as above.

Base pulse No.
Lower 6 bits

— The free-running counters can be cleared on compare-match A.

- Three independent interrupts
  — Two compare-match interrupts and one overflow interrupt can be requested inde
- Special functions provided by automatic addition function
  — The contents of OCRAR and OCRAF can be added to the contents of OCRA automatically, enabling a periodic waveform to be generated without software in



[Legend]

OCRA, OCRB:	Output compare registers A and B (16 bits)
OCRAR,OCRAF:	Output compare registers AR and AF (16 bits)
FRC:	Free-running counter (16 bits)
TCSR:	Timer control/status register (8 bits)
TIER:	Timer interrupt enable register (8 bits)
TCR:	Timer control register (8 bits)
TOCR:	Timer output compare control register (8 bits)

Figure 10.1 Block Diagram of 16-Bit Free-Running Timer

- Timer control/status register (TCSR)
- Timer control register (TCR)
- Timer output compare control register (TOCR)

Note: OCRA and OCRB share the same address. Register selection is controlled by th bit in TOCR.

10.2.1 Free-Running Counter (FRC)

FRC is a 16-bit readable/writable up-counter. The clock source is selected by bits CKS1 CKS0 in TCR. FRC can be cleared by compare-match A. When FRC overflows from H H'0000, the overflow flag bit (OVF) in TCSR is set to 1. FRC should always be accesse units; cannot be accessed in 8-bit units. FRC is initialized to H'0000.

10.2.2 Output Compare Registers A and B (OCRA and OCRB)

The FRT has two output compare registers, OCRA and OCRB, each of which is a 16-bi readable/writable register whose contents are continually compared with the value in FR a match is detected (compare-match), the corresponding output compare flag (OCFA or set to 1 in TCSR. OCR should always be accessed in 16-bit units; cannot be accessed in units. OCR is initialized to H'FFFF.

input clock together with a set value of H'0001 or less for OCRAR (or OCRAF).

OCRAR and OCRAF should always be accessed in 16-bit units; cannot be accessed in 8-
OCRAR and OCRAF are initialized to H'FFFF.

				request (OCIA) when output compare flag A (OC TCSR is set to 1. 0: OCIA requested by OCFA is disabled 1: OCIA requested by OCFA is enabled
2	OCIBE	0	R/W	Output Compare Interrupt B Enable Selects whether to enable output compare interr request (OCIB) when output compare flag B (OC TCSR is set to 1. 0: OCIB requested by OCFB is disabled 1: OCIB requested by OCFB is enabled
1	OVIE	0	R/W	Timer Overflow Interrupt Enable Selects whether to enable a free-running timer o request interrupt (FOVI) when the timer overflow (OVF) in TCSR is set to 1. 0: FOVI requested by OVF is disabled 1: FOVI requested by OVF is enabled
0	—	0	R	Reserved This bit is always read as 0 and cannot be modi

				[Setting condition] When FRC = OCRA [Clearing condition] Read OCFA when OCFA = 1, then write 0 to OC
2	OCFB	0	R/(W)*	Output Compare Flag B Indicates that the FRC value matches the OCRB [Setting condition] When FRC = OCRB [Clearing condition] Read OCFB when OCFB = 1, then write 0 to OC
1	OVF	0	R/(W)*	Overflow Flag Indicates that the FRC has overflowed. [Setting condition] When FRC overflows (changes from H'FFFF to H [Clearing condition] Read OVF when OVF = 1, then write 0 to OVF
0	CCLRA	0	R/W	Counter Clear A Selects whether the FRC is to be cleared on com match A (when the FRC and OCRA values match 0: FRC clearing is disabled 1: FRC is cleared on compare-match A

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

0	CKS0	0	R/W	Select clock source for FRC. 00: ϕ/2 internal clock source 01: ϕ/8 internal clock source 10: ϕ/32 internal clock source 11: Reserved

				Specifies whether OCRA is used in the normal op mode or in the operating mode using OCRAR and OCRAF. 0: The normal operating mode is specified for OC 1: The operating mode using OCRAR and OCRA specified for OCRA
5	ICRS	0	R/W	Input Capture Register Select Controls the access to OCRAR and OCRAF. 0: Access is disabled 1: Access is enabled
4	OCRS	0	R/W	Output Compare Register Select OCRA and OCRB share the same address. When address is accessed, the OCRS bit selects which is accessed. The operation of OCRA or OCRB is affected. 0: OCRA is selected 1: OCRB is selected
3 to 0	—	All 0	R	Reserved These bits are always read as 0 and cannot be m

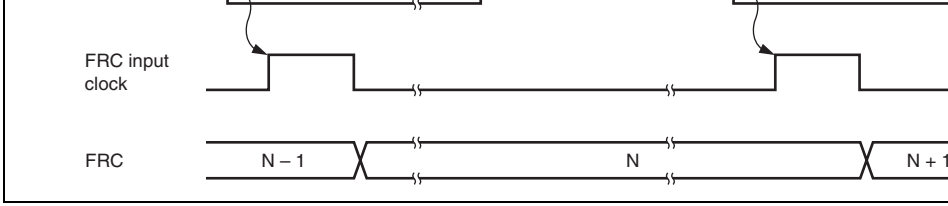


Figure 10.2 Increment Timing with Internal Clock Source

10.3.2 Output Compare Output Timing

A compare-match signal occurs at the last state when the FRC and OCR values match (a
timing when the FRC updates the counter value). Figure 10.3 shows the timing of this o
for compare-match A.

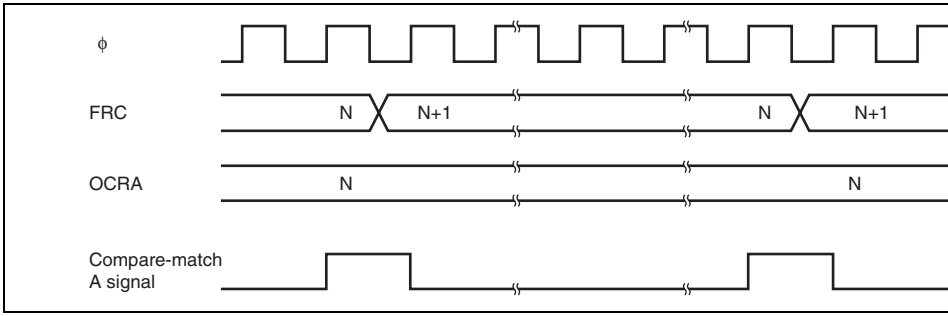


Figure 10.3 Timing of Output Compare A Output

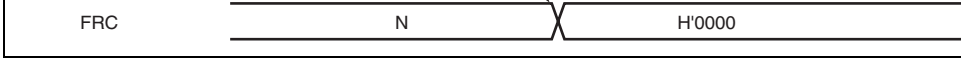


Figure 10.4 Clearing of FRC by Compare-Match A Signal

10.3.4 Timing of Output Compare Flag (OCF) Setting

The output compare flag, OCFA or OCFB, is set to 1 by a compare-match signal generate
the FRC value matches the OCRA or OCRB value. This compare-match signal is generat
last state in which the two values match, just before FRC increments to a new value. Whe
FRC and OCRA or OCRB value match, the compare-match signal is not generated until t
cycle of the clock source. Figure 10.5 shows the timing of setting the OCFA or OCFB fla

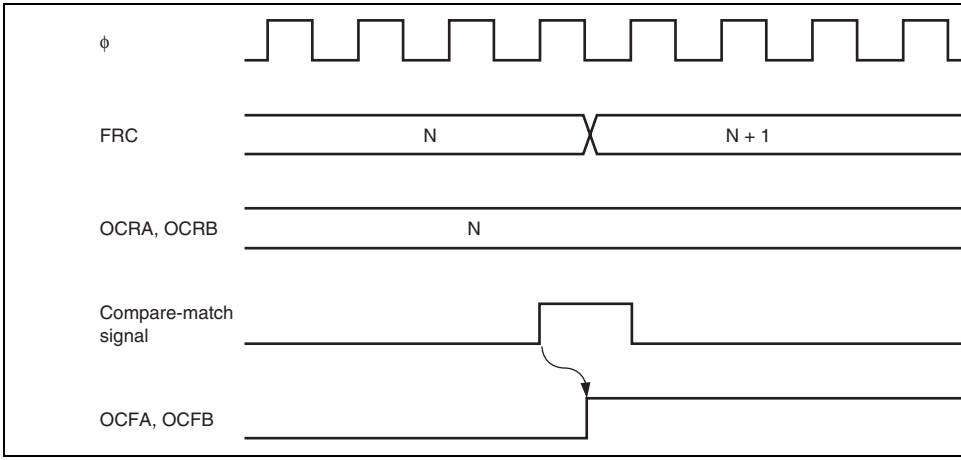


Figure 10.5 Timing of Output Compare Flag (OCFA or OCFB) Setting

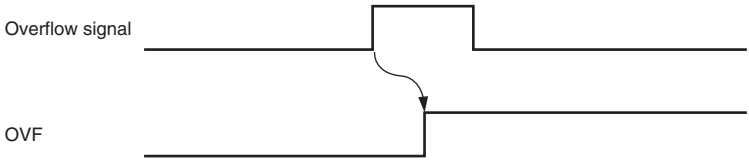


Figure 10.6 Timing of Overflow Flag (OVF) Setting

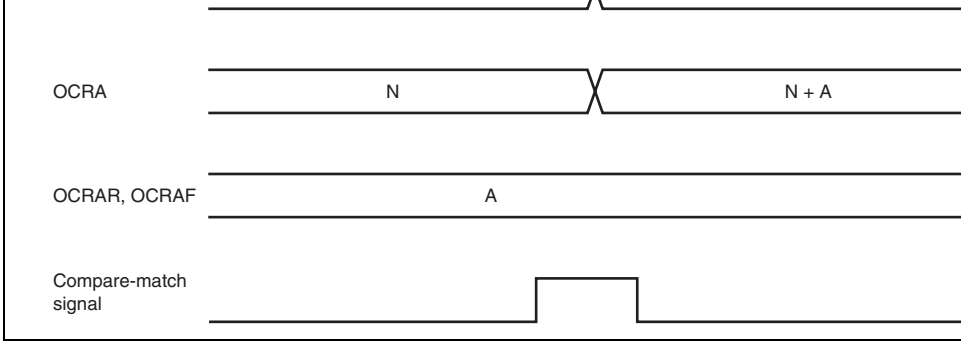


Figure 10.7 OCRA Automatic Addition Timing

10.4 Interrupt Sources

The free-running timer can request three interrupts: OCIA, OCIB, and FOVI. Each interr
be enabled or disabled by an enable bit in TIER. Independent signals are sent to the interr
controller for each interrupt. Table 10.1 lists the sources and priorities of these interrupts.

The OCIA and OCIB interrupts can be used as the on-chip DTC activation sources.

Table 10.1 FRT Interrupt Sources

Interrupt	Interrupt Source	Interrupt Flag	DTC Activation	Prio
OCIA	Compare match of OCRA	OCFA	Possible	High
OCIB	Compare match of OCRB	OCFB	Possible	↑
FOVI	Overflow of FRC	OVF	Not possible	Low

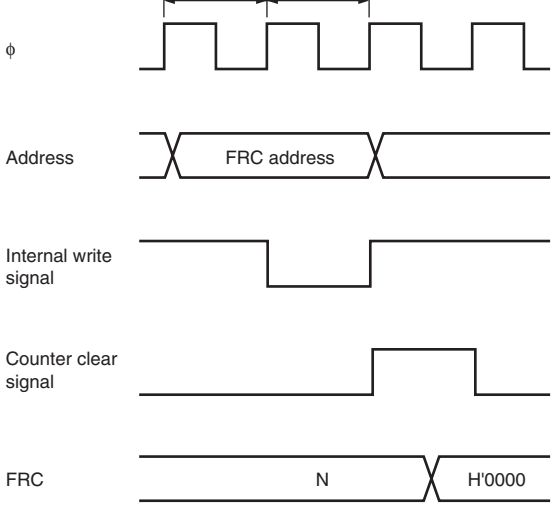


Figure 10.8 Conflict between FRC Write and Clear

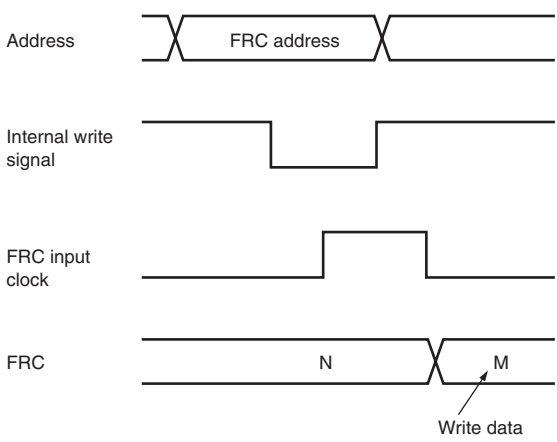


Figure 10.9 Conflict between FRC Write and Increment

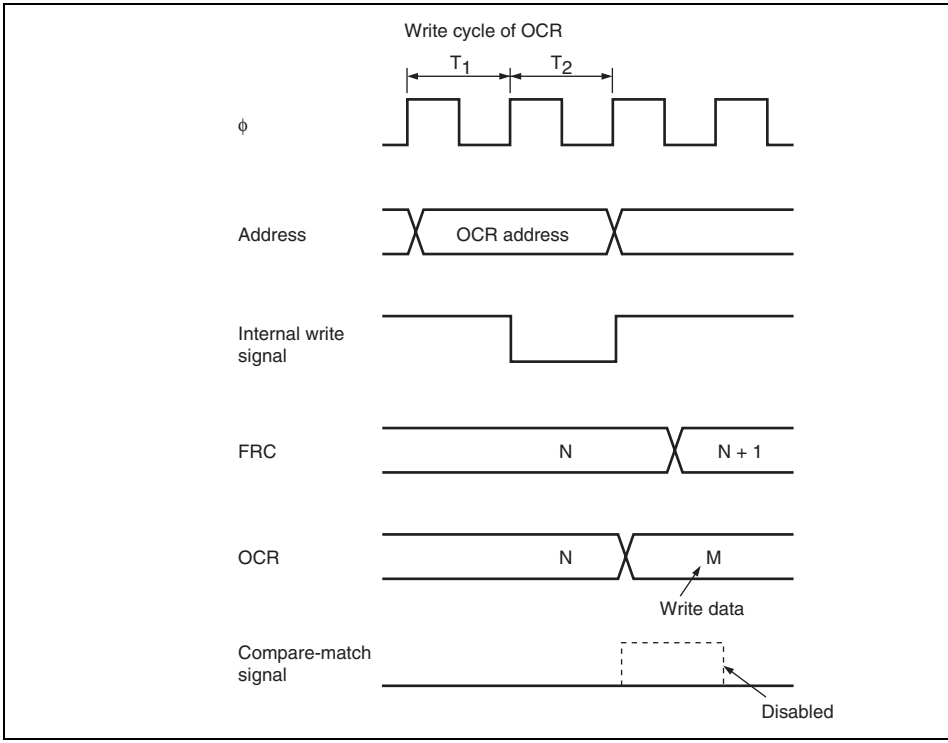


Figure 10.10 Conflict between OCR Write and Compare-Match (When Automatic Addition Function is Not Used)

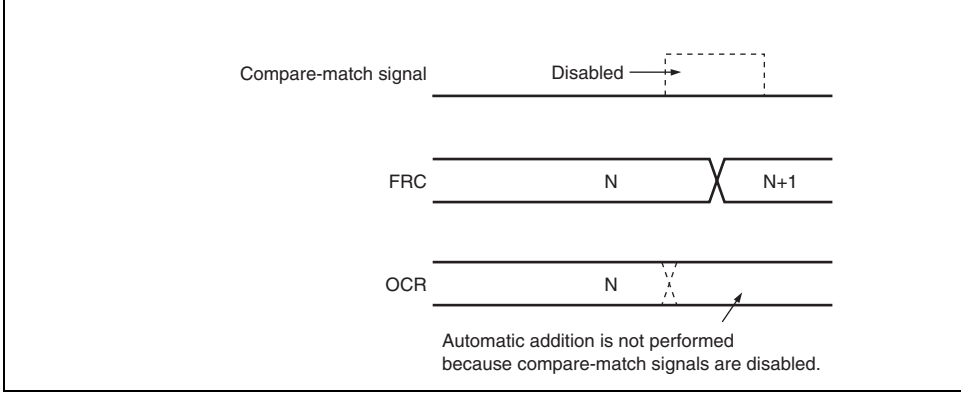
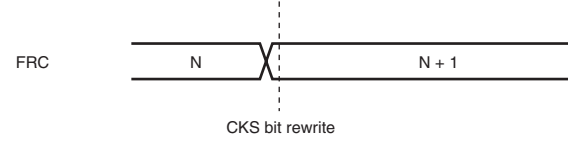


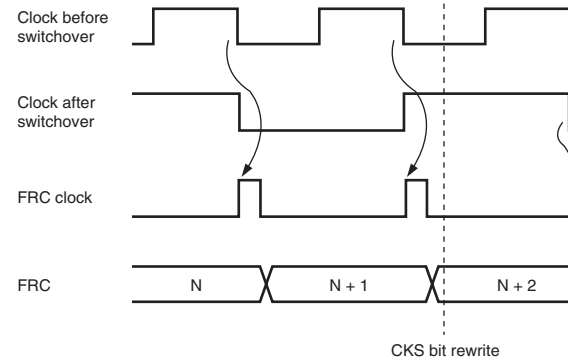
Figure 10.11 Conflict between OCR Write and Compare-Match (When Automatic Addition Function is Used)

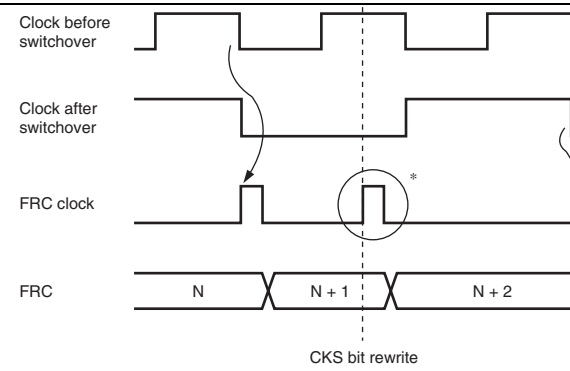
10.5.4 Switching of Internal Clock and FRC Operation

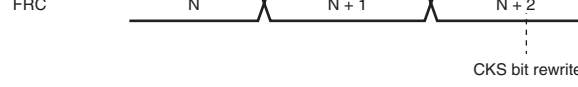
When the internal clock is changed, the changeover may source FRC to increment. This c
on the time at which the clock is switched (bits CKS1 and CKS0 are rewritten), as shown
10.2.

When an internal clock is used, the FRC clock is generated on detection of the falling edg
internal clock scaled from the system clock (ϕ). If the clock is changed when the old sour
and the new source is low, as in case no. 3 in table 10.2, the changeover is regarded as a f
edge that triggers the FRC clock, and FRC is incremented. Switching between an internal
and external clock can also source FRC to increment.



2	Switching from low to high	

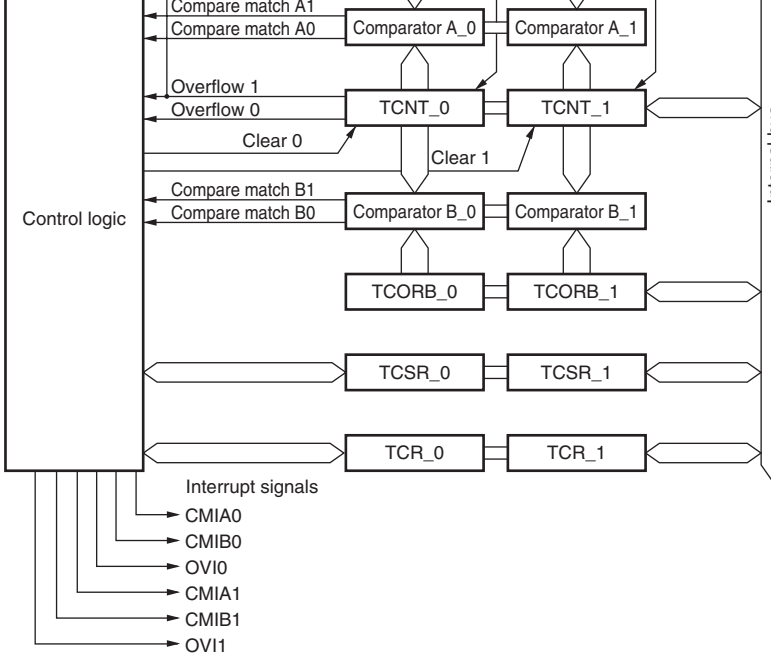
3	Switching from high to low	



Note: * Generated because the switchover is assumed to take place on a falling edge, FRC is incremented.

— TMR_0, TMR_1: The counter input clock can be selected from six internal cloc
— TMR_Y, TMR_X: The counter input clock can be selected from three internal cl

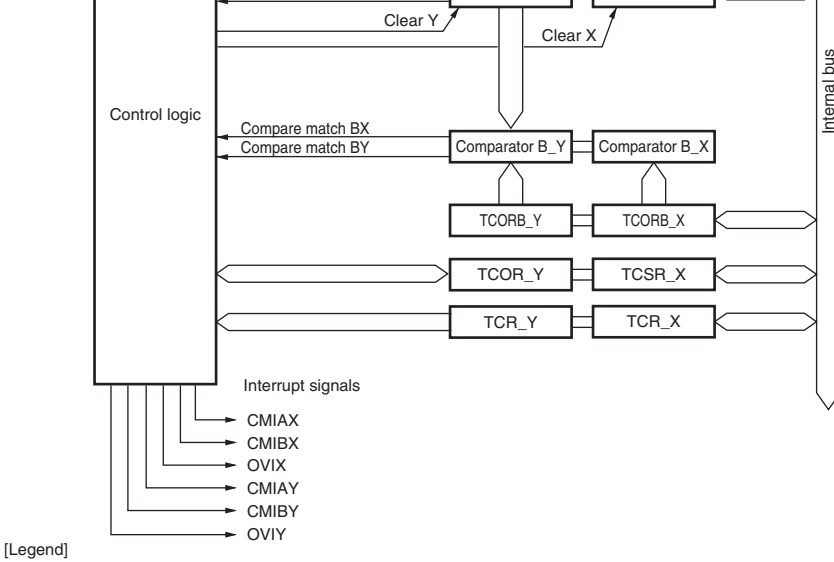
- Selection of two ways to clear the counters
  — The counters can be cleared on compare-match A and compare-match B.
- Cascading of TMR_0 and TMR_1
  (Cascading of TMR_Y and TMR_X is not allowed)
  — Operation as a 16-bit timer can be performed using TMR_0 as the upper half and as the lower half (16-bit count mode). TMR_1 can be used to count TMR_0 com match occurrences (compare-match count mode).
- Multiple interrupt sources for each channel
  — TMR_0, TMR_1, TMR_Y and TMR_X: Three interrupts: Compare-match compare-match B, and overflow



[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 (TMR_0 and TMR_1)



[Legend]

TCORA_Y:	Time constant register A_Y	TCORA_X:	Time constant register A_X
TCORB_Y:	Time constant register B_Y	TCORB_X:	Time constant register B_X
TCNT_Y:	Timer counter_Y	TCNT_X:	Timer counter_X
TCSR_Y:	Timer control / status register_Y	TCSR_X:	Timer control / status register_X
TCR_Y:	Timer control register_Y	TCR_X:	Timer control register_X
		TCORC:	Tme constant registerC

Figure 11.2 Block Diagram of 8-Bit Timer (TMR_Y and TMR_X)

- Timer connection register S (TCONRS)*

Notes: Some of the registers of TMR_X and TMR_Y use the same address. The regis be switched by the TMRX/Y bit in TCONRS.

* TCONRS is only provided for TMR_X

11.2.1 Timer Counter (TCNT)

Each TCNT is an 8-bit readable/writable up-counter. TCNT_0 and TCNT_1 comprise a s bit register, so they can be accessed together by word access. The clock source is selected CKS2 to CKS0 bits in TCR. TCNT can be cleared by a compare-match A signal or comp match B signal. The method of clearing can be selected by the CCLR1 and CCLR0 bits i When TCNT overflows (changes from H'FF to H'00), the OVF bit in TCSR is set to 1. T initialized to H'00.

TCNT_Y can be accessed when the TMRX/Y bit in TCONRS is 1. TCNT_X can be acce when the TMRX/Y bit in TCONRS is 0. See section 11.2.6, Timer Connection Register S (TCONRS).

11.2.3 Time Constant Register B (TCORB)

TCORB is an 8-bit readable/writable register. TCORB_0 and TCORB_ 1 comprise a sir
register, so they can be accessed together by word access. TCORB is continually compa
the value in TCNT. When a match is detected, the corresponding compare-match flag B
in TCSR is set to 1. However, comparison is disabled during the T2 state of a TCORB w
TCORB is initialized to H'FF.

TCORB_Y can be accessed when the TMRX/Y bit in TCONRS is 1. TCORB_X can be
when the TMRX/Y bit in TCONRS is 0. See section 11.2.6, Timer Connection Register
(TCONRS).

				Selects whether the CMFB interrupt request (CM enabled or disabled when the CMFB flag in TCSR 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 TCSR 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) enabled or disabled when the OVF flag in TCSR 1. 0: OVF interrupt request (OVI) is disabled 1: OVF interrupt request (OVI) is enabled
4	CCLR1	0	R/W	Counter Clear 1 and 0
3	CCLR0	0	R/W	Specify the clearing conditions of TCNT. 00: Counter clear is disabled 01: Counter clear is enabled on compare-match A 10: Counter clear is enabled on compare-match B 11: Setting prohibited
2 to 0	CKS2 to CKS0	All 0	R/W	Clock Select 2 to 0 Select the clock input to TCNT and count conditio together with the ICKS1 and ICKS0 bits in STCR. details, see table 11.1.

0	1	1	1	Increments at falling edge of internal clock
1	0	0	X	Increments at overflow signal from TCNT_
1	0	1	X	Setting prohibited
1	1	X	X	Setting prohibited

Note: * If the TMR_0 clock input is set as the TCNT_1 overflow signal and the TMR_ input is set as the TCNT_0 compare-match signal simultaneously, a count-up cannot be generated. Simultaneous setting of these conditions should be avo

[Legend] X: Don't care

0	1	1	1	Increments at falling edge of internal clock
1	0	0	X	Increments at compare-match A from TCN
1	0	1	X	Setting prohibited
1	1	X	X	Setting prohibited

Note: * If the TMR_0 clock input is set as the TCNT_1 overflow signal and the TMR_1 input is set as the TCNT_0 compare-match signal simultaneously, a count-up cannot be generated. Simultaneous setting of these conditions should be avoid

[Legend] X: Don't care

Table 11.1 (3) Clock Input to TCNT and Count Condition (TMR_X, TMR_Y)

	TCR			
Channel	**CKS2**	**CKS1**	**CKS0**	**Description**
TMR_Y	0	0	0	Disables clock input
	0	0	1	Increments at falling edge of internal clock
	0	1	0	Increments at falling edge of internal clock
	0	1	1	Increments at falling edge of internal clock
	1	X	X	Setting prohibited
TMR_Y	0	0	0	Disables clock input
	0	0	1	Increments at falling edge of internal clock
	0	1	0	Increments at falling edge of internal clock
	0	1	1	Increments at falling edge of internal clock
	1	X	X	Setting prohibited

				When the values of TCNT_0 and TCORB_0 ma
				[Clearing condition]
				Read CMFB when CMFB = 1, then write 0 in CM
6	CMFA	0	R/(W)*	Compare-Match Flag A
				[Setting condition]
				When the values of TCNT_0 and TCORA_0 ma
				[Clearing condition]
				Read CMFA when CMFA = 1, then write 0 in CM
5	OVF	0	R/(W)*	Timer Overflow Flag
				[Setting condition]
				When TCNT_0 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
				Selects whether the A/D conversion start reques compare match A is enabled or disabled.
				0: A/D conversion start request is disabled
				1: A/D conversion start request is enabled
3 to 0	—	All 1	R	Reserved
				These bits are always read as 1 and cannot be

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

				When the values of TCNT_1 and TCORA_1 mat
				[Clearing condition]
				Read CMFA when CMFA = 1, then write 0 in CM
5	OVF	0	R/(W)*	Timer Overflow Flag
				[Setting condition]
				When TCNT_1 overflows from H'FF to H'00
				[Clearing condition]
				Read OVF when OVF = 1, then write 0 in OVF
4 to 0	—	All 1	R	Reserved
				These bits are always read as 1 and cannot be m

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

				[Setting condition] When the values of TCNT_Y and TCORA_Y ma [Clearing condition] Read CMFA when CMFA = 1, then write 0 in CM
5	OVF	0	R/(W)*	Timer Overflow Flag [Setting condition] When TCNT_Y overflows from H'FF to H'00 [Clearing condition] Read OVF when OVF = 1, then write 0 in OVF
4 to 0	—	All 1	R	Reserved These bits are always read as 1 and cannot be

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

				[Setting condition] When the values of TCNT_X and TCORA_X mat [Clearing condition] Read CMFA when CMFA = 1, then write 0 in CM
5	OVF	0	R/(W)*	Timer Overflow Flag [Setting condition] When TCNT_X overflows from H'FF to H'00 [Clearing condition] Read OVF when OVF = 1, then write 0 in OVF
4 to 0	—	All 1	R	Reserved These bits are always read as 1 and cannot be m

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

				1. The TMR_Y registers are accessed at addres H'FFFFF0 to H'FFFFF5
6 to 0	—	All 0	R/W	Reserved The initial values should not be changed.

Table 11.2 Registers Accessible by TMR_X/TMR_Y

TMRX/Y	H'FFFFF0	H'FFFFF1	H'FFFFF2	H'FFFFF3	H'FFFFF4	H'FFFFF5	H'FFFFF6
0	TMR_X	TMR_X	TMR_X	TMR_X	TMR_X	TMR_X	TMR_X
	TCR_X	TCSR_X			TCNT_X		TCORA_X
1	TMR_Y	TMR_Y	TMR_Y	TMR_Y	TMR_Y	TMR_Y	
	TCR_Y	TCSR_Y	TCORA_Y	TCORB_Y	TCNT_Y		

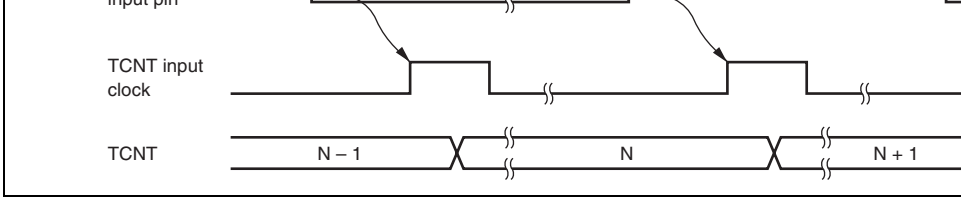


Figure 11.3 Count Timing for Internal Clock Input

11.3.2 Timing of CMFA and CMFB Setting at Compare-Match

The CMFA and CMFB flags in TCSR are set to 1 by a compare-match signal generated v
TCNT and TCOR values match. The compare-match signal is generated at the last state i
the match is true, just when the timer counter is updated. Therefore, when TCNT and TC
match, the compare-match signal is not generated until the next TCNT input clock. Figur
shows the timing of CMF flag setting.

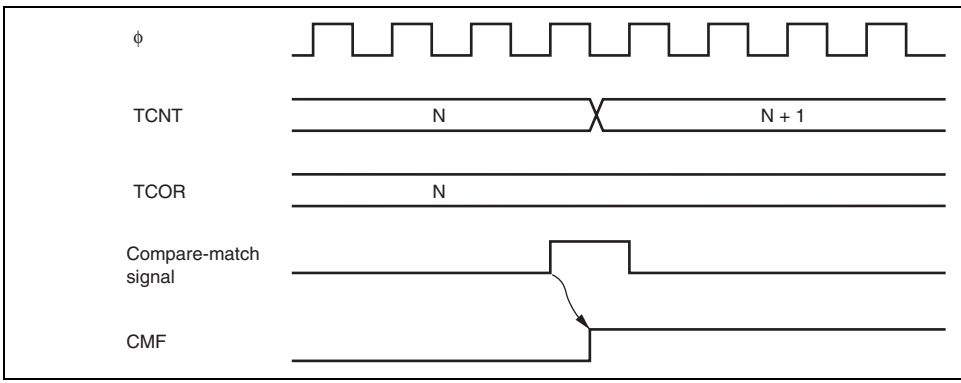


Figure 11.4 Timing of CMF Setting at Compare-Match

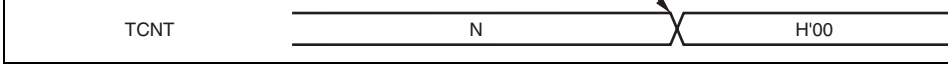


Figure 11.5 Timing of Counter Clear by Compare-Match

11.3.4 Timing of Overflow Flag (OVF) Setting

The OVF bit in TCSR is set to 1 when the TCNT overflows (changes from H'FF to H'00
11.6 shows the timing of OVF flag setting.

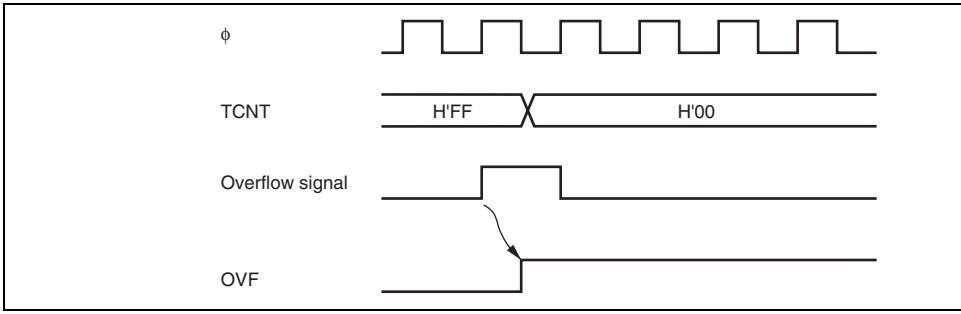


Figure 11.6 Timing of OVF Flag Setting

- 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 compa the 16-bit counter (TCNT_0 and TCNT_1 together) is cleared when a 16-bit comp match occurs. The 16-bit counter (TCNT_0 and TCNT_1 together) is also cleared counter clear by the TMI0 pin has been set.
  - The settings of the CCLR1 and CCLR0 bits in TCR_1 are ignored. The lower 8 bi be cleared independently.

11.4.2 Compare-Match Count Mode

When bits CKS2 to CKS0 in TCR_1 are B′100, TCNT_1 counts the occurrence of compa A for TMR_0. TMR_0 and TMR_1 are controlled independently. Conditions such as sett CMF flag, generation of interrupts, and counter clearing are in accordance with the settin each channel.

Table 11.3 Interrupt Sources of 8-Bit Timers TMR_0, TMR_1, TMR_Y, and TM

Channel	Name	Interrupt Source	Interrupt Flag	DTC Activation	Int Pri
TMR_X	CMIAX	TCORA_X compare-match	CMFA	Possible	Hig
	CMIBX	TCORB_X compare-match	CMFB	Possible	
	OVIX	TCNT_X overflow	OVF	Not possible	
TMR_0	CMIA0	TCORA_0 compare-match	CMFA	Possible	
	CMIB0	TCORB_0 compare-match	CMFB	Possible	
	OVI0	TCNT_0 overflow	OVF	Not possible	
TMR_1	CMIA1	TCORA_1 compare-match	CMFA	Possible	
	CMIB1	TCORB_1 compare-match	CMFB	Possible	
	OVI1	TCNT_1 overflow	OVF	Not possible	
TMR_Y	CMIAY	TCORA_Y compare-match	CMFA	Possible	
	CMIBY	TCORB_Y compare-match	CMFB	Possible	
	OVIY	TCNT_Y overflow	OVF	Not possible	Lov

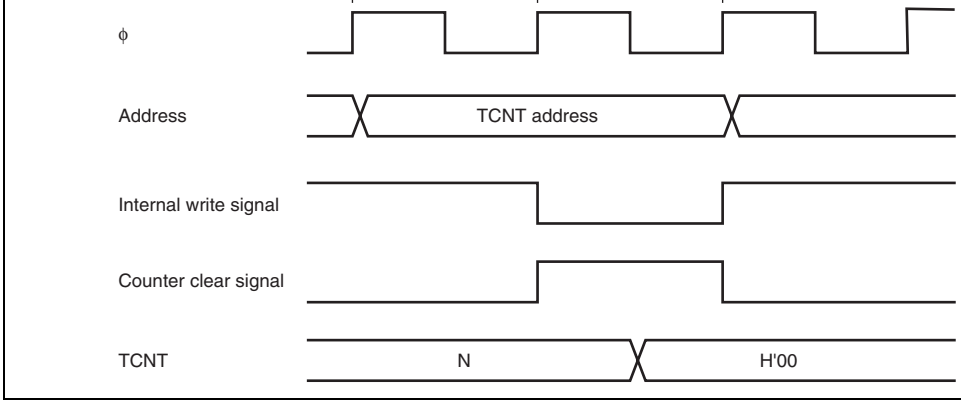


Figure 11.7 Conflict between TCNT Write and Counter Clear

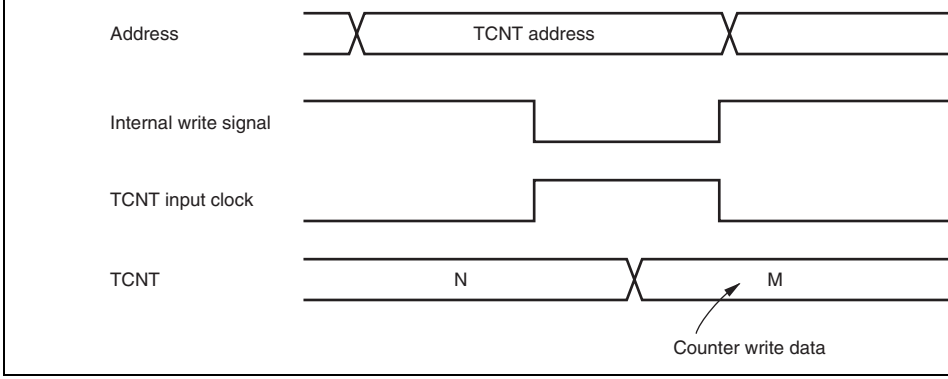


Figure 11.8 Conflict between TCNT Write and Increment

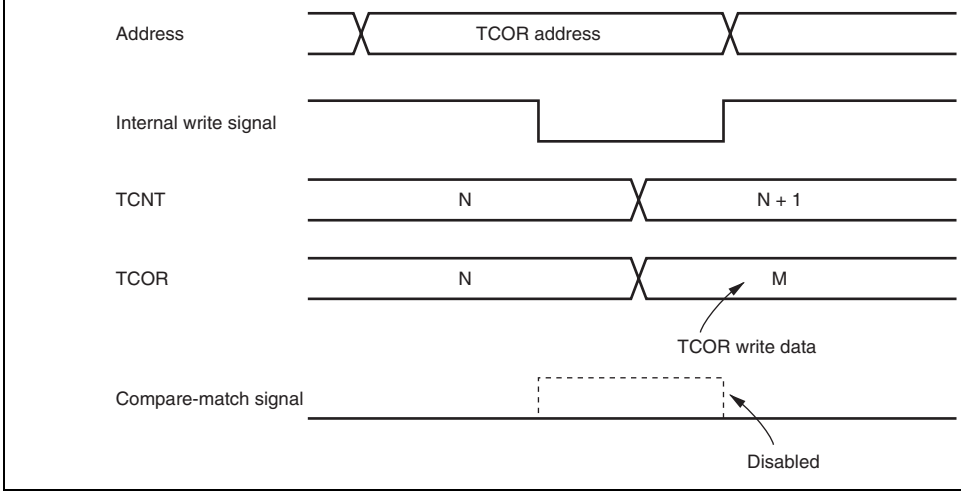
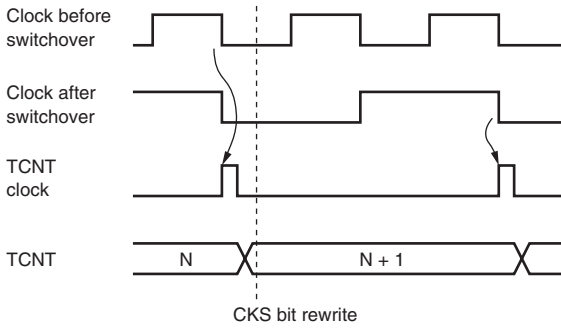
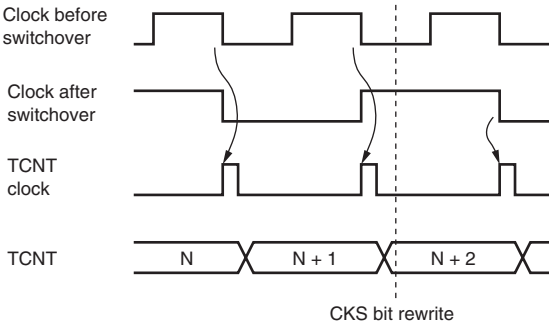
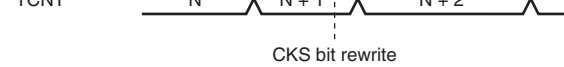


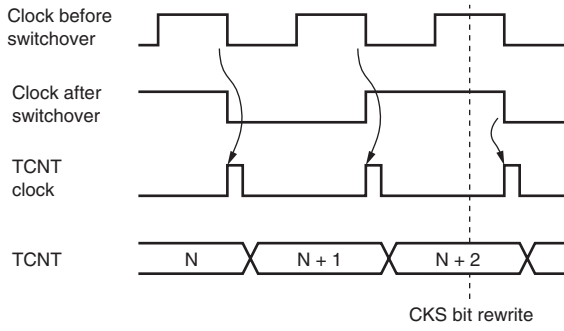
Figure 11.9 Conflict between TCOR Write and Compare-Match

Erroneous incrementation can also happen when switching between internal and externa

Table 11.4 Switching of Internal Clocks and TCNT Operation

No.	Timing of Switchover by Means of CKS1 and CKS0 Bits	TCNT Clock Operation
1	Clock switching from low to low level*[1]	
2	Clock switching from low to high level*[2]	



4	Clock switching from high to high level	

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.

11.6.5 Mode Setting with Cascaded Connection

If the 16-bit count mode and compare-match count mode are set simultaneously, the inpu pulses for TCNT_0 and TCNT_1 are not generated, and thus the counters will stop opera Simultaneous setting of these two modes should be avoided.

12.1 Features

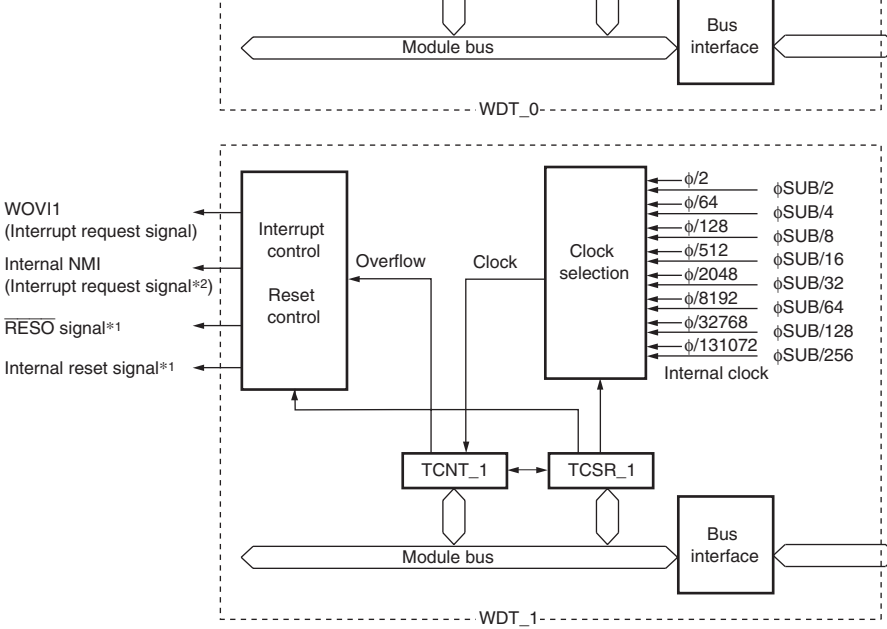
- Selectable from eight (WDT_0) or 16 (WDT_1) counter input clocks.
- Switchable between watchdog timer mode and interval timer mode

Watchdog Timer Mode:

- If the counter overflows, an internal reset or an internal NMI interrupt is generated.
- When the LSI is selected to be internally reset at counter overflow, a low level signa from the $\overline{\text{RESO}}$ pin if the counter overflows.

Internal Timer Mode:

- If the counter overflows, an internal timer interrupt (WOVI) is generated.



[Legend]

TCSR_0: Timer control/status register_0
TCNT_0: Timer counter_0
TCSR_1: Timer control/status register_1
TCNT_1: Timer counter_1

Notes: 1. The RESO signal outputs the low level signal when the internal reset signal is generated due to a TCNT overflow of either WDT_0 or WDT_1. The internal reset signal first resets the WDT in which the overflow has occurred first.

2. The internal NMI interrupt signal can be independently output from either WDT_0 or WDT The interrupt controller does not distinguish the NMI interrupt request from WDT_0 from that from WDT_1.

Figure 12.1 Block Diagram of WDT

12.3 Register Descriptions

The WDT has the following registers. To prevent accidental overwriting, TCSR and TC
to be written to in a method different from normal registers. For details, see section 12.6
on Register Access. For details on the system control register, see section 3.2.2, System
Register (SYSCR).

- Timer counter (TCNT)
- Timer control/status register (TCSR)

12.3.1 Timer Counter (TCNT)

TCNT is an 8-bit readable/writable up-counter. TCNT is initialized to H'00 when the TM
timer control/status register (TCSR) is cleared to 0.

				[Setting conditions] • When TCNT overflows (changes from H'FF to • When internal reset request generation is sele watchdog timer mode, OVF is cleared automa the internal reset. [Clearing conditions] • When TCSR is read when OVF = 1, then 0 is OVF • When 0 is written to TME
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 1: Watchdog timer mode
5	TME	0	R/W	Timer Enable When this bit is set to 1, TCNT starts counting. When this bit is cleared, TCNT stops counting an initialized to H'00.
4	—	0	R/W	Reserved The initial value should not be changed.
3	RST/$\overline{NMI}$	0	R/W	Reset or NMI Selects to request an internal reset or an NMI inte when TCNT has overflowed. 0: An NMI interrupt is requested 1: An internal reset is requested

		101: ϕ/8192 (period: 61.68 ms) 110: ϕ/32768 (period: 246.7 ms) 111: ϕ/131072 (period: 986.9 ms)

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

				the internal reset. [Clearing conditions] • When TCSR is read when OVF = 1*2, then 0 i to OVF • When 0 is written to TME
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 1: Watchdog timer mode
5	TME	0	R/W	Timer Enable When this bit is set to 1, TCNT starts counting. When this bit is cleared, TCNT stops counting an initialized to H'00. When PSS = 1, TCNT is not ini Write H'00 to TCNT to initialize TCNT.
4	PSS	0	R/W	Prescaler Select Selects the clock source to be input to TCNT. 0: Counts the divided cycle of ϕ–based prescaler 1: Counts the divided cycle of ϕSUB–based presc (PSS)
3	RST/$\overline{NMI}$	0	R/W	Reset or NMI Selects to request an internal reset or an NMI inte when TCNT has overflowed. 0: An NMI interrupt is requested 1: An internal reset is requested

011: ϕ/512 (cycle: 3.856 ms)

100: ϕ/2048 (cycle: 15.42 ms)

101: ϕ/8192 (cycle: 61.68 ms)

110: ϕ/32768 (cycle: 246.7 ms)

111: ϕ/131072 (cycle: 986.9 ms)

When PSS = 1:

000: ϕSUB/2 (cycle: 15.6 ms)

001: ϕSUB/4 (cycle: 31.3 ms)

010: ϕSUB/8 (cycle: 62.5 ms)

011: ϕSUB/16 (cycle: 125 ms)

100: ϕSUB/32 (cycle: 250 ms)

101: ϕSUB/64 (cycle: 500 ms)

110: ϕSUB/128 (cycle: 1 s)

111: ϕSUB/256 (cycle: 2 s)

Notes: 1. Only 0 can be written to clear the flag.

2. When OVF is polled with the interval timer interrupt disabled, OVF = 1 must b least twice.

If the RST/NMI bit of TCSR is set to 1, when the TCNT overflows, an internal reset sign
LSI is issued for 518 system clocks, and the low level signal is simultaneously output fro
$\overline{\text{RESO}}$ pin for 132 states, as shown in figure 12.2. If the RST/$\overline{\text{NMI}}$ bit is cleared to 0, whe
TCNT overflows, an NMI interrupt request is generated. Here, the output from the $\overline{\text{RESO}}$
remains high.

An internal reset request from the watchdog timer and a reset input from the $\overline{\text{RES}}$ pin are
processed in the same vector. Reset source can be identified by the XRST bit status in SY
If a reset caused by a signal input to the $\overline{\text{RES}}$ pin occurs at the same time as a reset caused
WDT overflow, the $\overline{\text{RES}}$ pin reset has priority and the XRST bit in SYSCR is set to 1.

An NMI interrupt request from the watchdog timer and an interrupt request from the NM
processed in the same vector. Do not handle an NMI interrupt request from the watchdog
and an interrupt request from the NMI pin at the same time.

WT/$\overline{IT}$: Timer mode select bit
TME: Timer enable bit
OVF: Overflow flag

Note: * After the OVF bit becomes 1, it is cleared to 0 by an internal reset. The XRST bit is also cleared to 0.

Figure 12.2 Watchdog Timer Mode (RST/$\overline{NMI}$ = 1) Operation

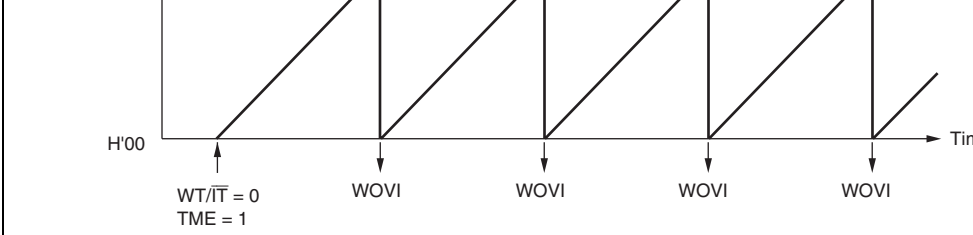


Figure 12.3 Interval Timer Mode Operation

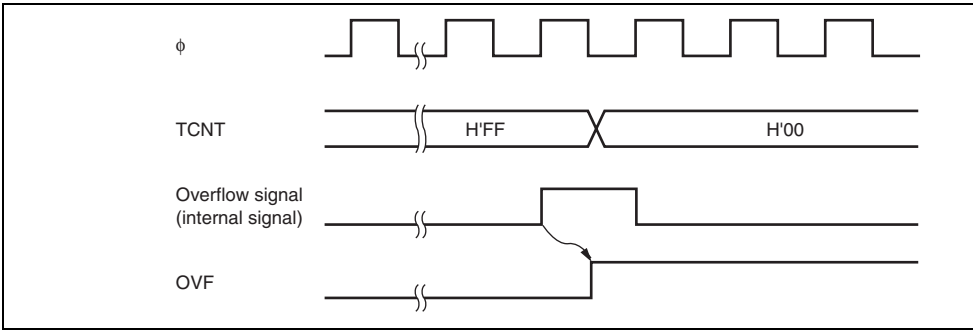


Figure 12.4 OVF Flag Set Timing

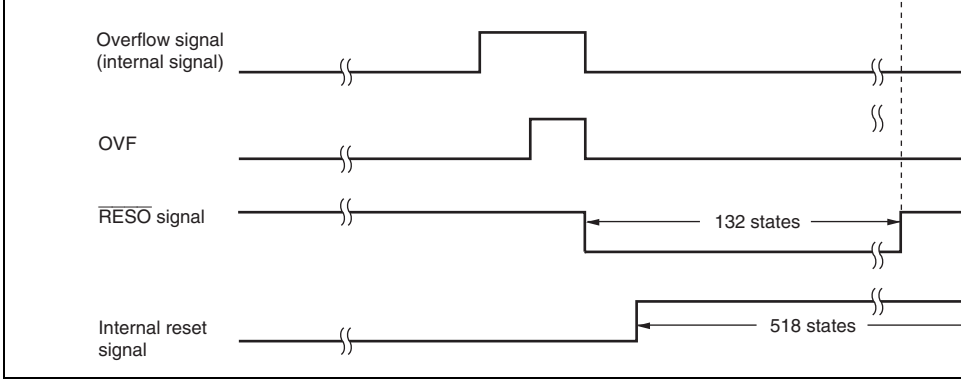


Figure 12.5 Output Timing of $\overline{\text{RESO}}$ signal

This LSI has retain state pins, which are only initialized by a system reset. The outputs o
pins are retained even when an internal reset is generated by the overflow signal of the W
more information, see section 8, I/O Ports.

Name	Interrupt Source	Interrupt Flag	DTC Activa
WOVI	TCNT overflow	OVF	Not possible

These registers must be written to by a word transfer instruction. They cannot be written byte transfer instruction.

TCNT and TCSR both have the same write address. Therefore, satisfy the relative condi shown in figure 12.6 to write to TCNT or TCSR. To write to TCNT, the higher bytes mu the value H'5A and the lower bytes must contain the write data. To write to TCSR, the h bytes must contain the value H'A5 and the lower bytes must contain the write data.

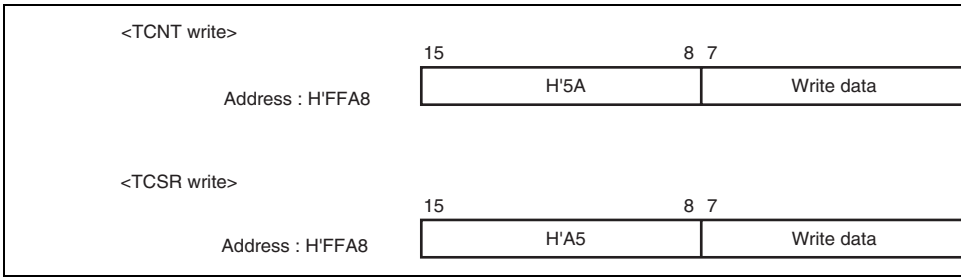


Figure 12.6 Writing to TCNT and TCSR (WDT_0)

Reading from TCNT and TCSR (Example of WDT_0):

These registers are read in the same way as other registers. The read address is H'FFA8 and H'FFA9 for TCNT.

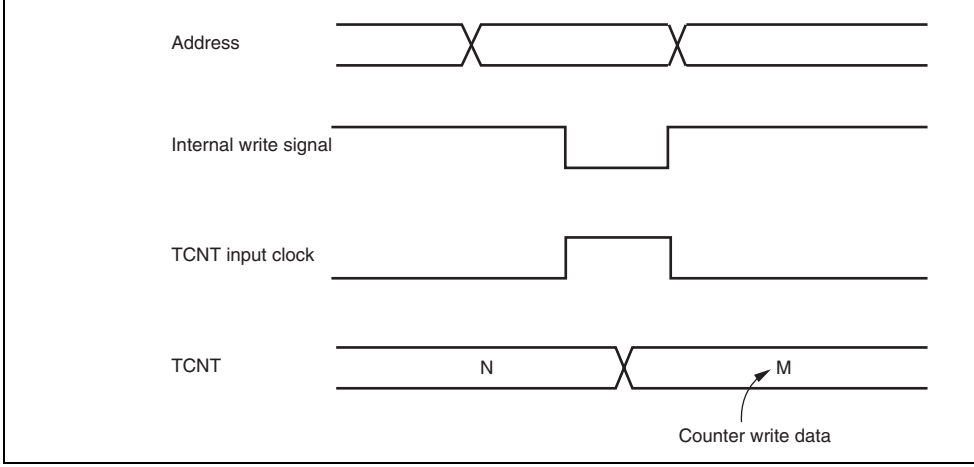


Figure 12.7 Conflict between TCNT Write and Increment

12.6.3 Changing Values of CKS2 to CKS0 Bits

If CKS2 to CKS0 bits in TCSR are written to while the WDT is operating, errors could o
the incrementation. Software must stop the watchdog timer (by clearing the TME bit to 0
changing the values of CKS2 to CKS0 bits.

12.6.4 Changing Value of PSS Bit

If the PSS bit in TCSR_1 is written to while the WDT is operating, errors could occur in
operation. Stop the watchdog timer (by clearing the TME bit to 0) before changing the va
PSS bit.

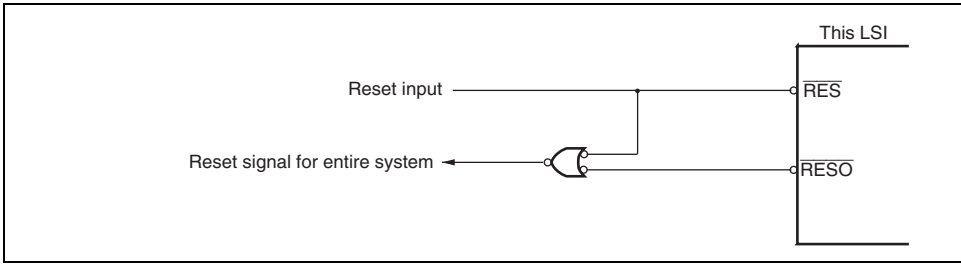


Figure 12.8 Sample Circuit for Resetting the System by the $\overline{\text{RESO}}$ Signa

13.1 Features

- Choice of asynchronous or clock synchronous serial communication mode
- Full-duplex communication capability

  The transmitter and receiver are mutually independent, enabling transmission and re
  be executed simultaneously. Double-buffering is used in both the transmitter and the
  enabling continuous transmission and continuous reception of serial data.
- On-chip baud rate generator allows any bit rate to be selected

  The external clock can be selected as a transfer clock source (except for the smart ca
  interface).
- Choice of LSB-first or MSB-first transfer (except in the case of asynchronous mode
- Four interrupt sources

  Four interrupt sources — transmit-end, transmit-data-empty, receive-data-full, and r
  error — that can issue requests.

  The transmit-data-empty and receive-data-full interrupt sources can activate DTC.
- Module stop mode availability

- Data length: 8 bits
- Receive error detection: Overrun errors

Smart Card Interface:

- An error signal can be automatically transmitted on detection of a parity error during
- Data can be automatically re-transmitted on detection of a error signal during transmi
- Both direct convention and inverse convention are supported

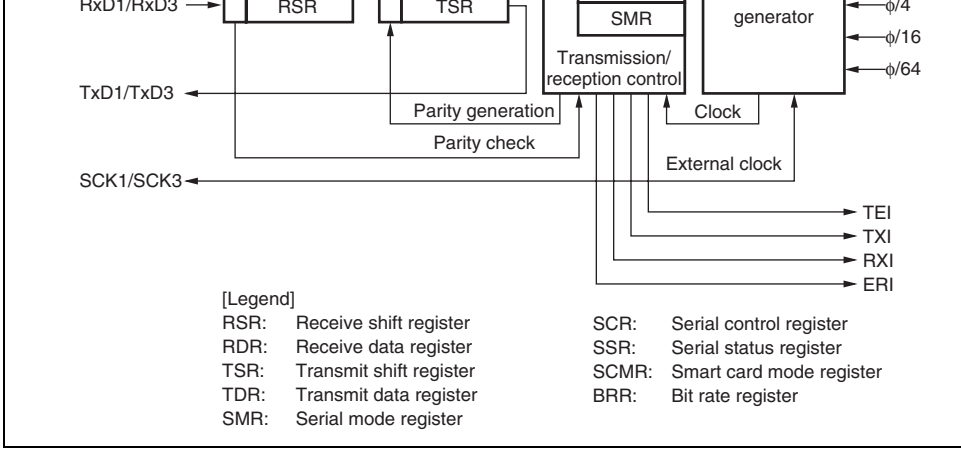


Figure 13.1 Block Diagram of SCI_1 and SCI_3

	TxD1	Output	Channel 1 transmit data output
3	SCK3	Input/Output	Channel 3 clock input/output
	RxD3	Input	Channel 3 receive data input
		Input/Output	Channel 3 transmit/receive data input/output (smart card interface is selected)
	TxD3	Output	Channel 3 transmit data output

Note: * Pin names SCK, RxD, and TxD are used in the text for all channels, omitting th channel designation.

13.3 Register Descriptions

The SCI has the following registers for each channel. Some bits in the serial mode registe serial status register (SSR), and serial control register (SCR) have different functions in d modes—normal serial communication interface mode and smart card interface mode; the the bits are described separately for each mode in the corresponding register sections.

- Receive shift register (RSR)
- Receive data register (RDR)
- Transmit data register (TDR)
- Transmit shift register (TSR)
- Serial mode register (SMR)
- Serial control register (SCR)
- Serial status register (SSR)
- Smart card mode register (SCMR)
- Bit rate register (BRR)

receive operations be performed. After confirming that the RDRF bit in SSR is set to 1, for only once. RDR cannot be written to by the CPU.

13.3.3 Transmit Data Register (TDR)

TDR is an 8-bit register that stores transmit data. When the SCI detects that TSR is emp transfers the transmit data written in TDR to TSR and starts transmission. The double-bu structures of TDR and TSR enables continuous serial transmission. If the next transmit d already been written to TDR when one frame of data is transmitted, the SCI transfers the data to TSR to continue transmission. Although TDR can be read from or written to by t all times, to achieve reliable serial transmission, write transmit data to TDR for only onc confirming that the TDRE bit in SSR is set to 1.

13.3.4 Transmit Shift Register (TSR)

TSR is a shift register that transmits serial data. To perform serial data transmission, the transfers transmit data from TDR to TSR, then sends the data to the TxD pin. TSR cann directly accessed by the CPU.

				1: Clock synchronous mode
6	CHR	0	R/W	Character Length (enabled only in asynchronous 0: Selects 8 bits as the data length. 1: Selects 7 bits as the data length. LSB-first is fi the MSB of TDR is not transmitted in transmiss In clock synchronous mode, a fixed data length o used.
5	PE	0	R/W	Parity Enable (enabled only in asynchronous mo When this bit is set to 1, the parity bit is added to data before transmission, and the parity bit is che reception. For a multiprocessor format, parity bit and checking are not performed regardless of the setting.
4	O/$\overline{E}$	0	R/W	Parity Mode (enabled only when the PE bit is 1 in asynchronous mode) 0: Selects even parity. 1: Selects odd parity.
3	STOP	0	R/W	Stop Bit Length (enabled only in asynchronous m 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 th second stop bit is 0, it is treated as the start bit of transmit frame.
2	MP	0	R/W	Multiprocessor Mode (enabled only in asynchron mode) When this bit is set to 1, the multiprocessor communication function is enabled. The PE bit ar settings are invalid in multiprocessor mode.

				the baud rate, see section 13.3.9, Bit Rate Regi (BRR). n is the decimal display of the value of n (see section 13.3.9, Bit Rate Register (BRR)).

- Bit Functions in Smart Card Interface Mode (when SMIF in SCMR = 1)

Bit	Bit Name	Initial Value	R/W	Description
7	GM	0	R/W	GSM Mode Setting this bit to 1 allows GSM mode operation mode, the TEND set timing is put forward to 11. from the start and the clock output control functi appended. For details, see section 13.7.8, Cloc Control.
6	BLK	0	R/W	Setting this bit to 1 allows block transfer mode o For details, see section 13.7.3, Block Transfer M
5	PE	0	R/W	Parity Enable (valid only in asynchronous mode When this bit is set to 1, the parity bit is added t data before transmission, and the parity bit is ch reception. Set this bit to 1 in smart card interfac
4	O/$\overline{E}$	0	R/W	Parity Mode (valid only when the PE bit is 1 in asynchronous mode) 0: Selects even parity 1: Selects odd parity For details on the usage of this bit in smart card mode, see section 13.7.2, Data Format (Except Transfer Mode).

				Timing and Reception Margin. S is described in s 13.3.9, Bit Rate Register (BRR).
1	CKS1	0	R/W	Clock Select 1 and 0
0	CKS0	0	R/W	These bits select the clock source for the baud ra 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 relation between the bit rate register settir the baud rate, see section 13.3.9, Bit Rate Regist (BRR). n is the decimal display of the value of n i (see section 13.3.9, Bit Rate Register (BRR)).

Note: * etu: Element Time Unit (time taken to transfer one bit)

				When this bit is set to 1, a TXI interrupt request enabled.
6	RIE	0	R/W	Receive Interrupt Enable When this bit is set to 1, RXI and ERI interrupt r are enabled.
5	TE	0	R/W	Transmit Enable When this bit is set to 1, transmission is enablec
4	RE	0	R/W	Receive Enable When this bit is set to 1, reception is enabled.
3	MPIE	0	R/W	Multiprocessor Interrupt Enable (enabled only w 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 RDRF, FER, and ORER status flags in SSR is d On receiving data in which the multiprocessor bi bit is automatically cleared and normal reception resumed. For details, see section 13.5, Multipro Communication Function.
2	TEIE	0	R/W	Transmit End Interrupt Enable When this bit is set to 1, a TEI interrupt request enabled.

1x: External clock

(Inputs a clock with a frequency 16 times the
from the SCK pin.)

Clock synchronous mode:

0x: Internal clock (SCK pin functions as clock out

1x: External clock (SCK pin functions as clock inp

[Legend]

x: Don't care

4	RE	0	R/W	Receive Enable When this bit is set to 1, reception is enabled.
3	MPIE	0	R/W	Multiprocessor Interrupt Enable (enabled only wh bit in SMR is 1 in asynchronous mode) Write 0 to this bit in smart card interface mode.
2	TEIE	0	R/W	Transmit End Interrupt Enable Write 0 to this bit in smart card interface mode.
1	CKE1	0	R/W	Clock Enable 1 and 0
0	CKE0	0	R/W	These bits control the clock output from the SCK GSM mode, clock output can be dynamically swi details, see section 13.7.8, Clock Output Control. When GM in SMR = 0 00: Output disabled (SCK pin functions as I/O po 01: Clock output 1x: Reserved When GM in SMR = 1 00: Output fixed to low 01: Clock output 10: Output fixed to high 11: Clock output

[Legend]

x: Don't care

				Indicates whether TDR contains transmit data. [Setting conditions] • When the TE bit in SCR is 0 • When data is transferred from TDR to TSR ar ready for data write [Clearing conditions] • When 0 is written to TDRE after reading TDR • When a TXI interrupt request is issued allowir write data to TDR
6	RDRF	0	R/(W)*	Receive Data Register Full Indicates that receive data is stored in RDR. [Setting condition] • When serial reception ends normally and rece transferred from RSR to RDR [Clearing conditions] • When 0 is written to RDRF after reading RDF • When an RXI interrupt request is issued allow read data from RDR The RDRF flag is not affected and retains its prev when the RE bit in SCR is cleared to 0.

				When the stop bit is 0 [Clearing condition] When 0 is written to FER after reading FER = 1 In 2-stop-bit mode, only the first stop bit is check
3	PER	0	R/(W)*	Parity Error [Setting condition] When a parity error is detected during reception [Clearing condition] When 0 is written to PER after reading PER = 1
2	TEND	1	R	Transmit End [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 a TXI interrupt request is issued allow to write 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 previ is retained.
0	MPBT	0	R/W	Multiprocessor Bit Transfer MPBT stores the multiprocessor bit to be added transmit frame.

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

				• When 0 is written to TDRE after reading TDRI • When a TXI interrupt request is issued allowin write data to TDR
6	RDRF	0	R/(W)*1	Receive Data Register Full Indicates whether the receive data is stored in RD [Setting condition] • When serial reception ends normally and rece is transferred from RSR to RDR [Clearing conditions] • When 0 is written to RDRF after reading RDR • When an RXI interrupt request is issued allow to read data from RDR The RDRF flag is not affected and retains its prev value when the RE bit in SCR is cleared to 0.
5	ORER	0	R/(W)*1	Overrun Error [Setting condition] When the next serial reception is completed while 1 [Clearing condition] When 0 is written to ORER after reading ORER =
4	ERS	0	R/(W)*1	Error Signal Status [Setting condition] When a low error signal is sampled [Clearing condition] When 0 is written to ERS after reading ERS = 1

				transferred to TDR. [Setting conditions] • When both TE in SCR and ERS are 0 • When ERS = 0 and TDRE = 1 after a specifi passed after the start of 1-byte data transfer timing depends on the register setting as foll When GM = 0 and BLK = 0, 2.5 etu*2 after transmission start When GM = 0 and BLK = 1, 1.5 etu*2 after transmission start When GM = 1 and BLK = 0, 1.0 etu*2 after transmission start When GM = 1 and BLK = 1, 1.0 etu*2 after transmission start [Clearing conditions] • When 0 is written to TDRE after reading TDRE = 1 • When a TXI interrupt request is issued allow to write the next data to TDR
1	MPB	0	R	Multiprocessor Bit 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: 1. Only 0 can be written to clear the flag.
2. etu: Element Time Unit (time taken to transfer one bit)

				0: TDR contents are transmitted with LSB-first. Stores receive data as LSB first in RDR. 1: TDR contents are transmitted with MSB-first. Stores receive data as MSB first in RDR. The SDIR bit is valid only when the 8-bit data form used for transmission/reception; when the 7-bit da format is used, data is always transmitted/receive LSB-first.
2	SINV	0	R/W	Smart Card Data Invert Specifies inversion of the data logic level. The SIN does not affect the logic level of the parity bit. Wh parity bit is inverted, invert the O/$\overline{E}$ bit in SMR. 0: TDR contents are transmitted as they are. Rec is stored as it is in RDR. 1: TDR contents are inverted before being transm Receive data is stored in inverted form in RDR
1	—	1	R	Reserved This bit is always read as 1 and cannot be modifie
0	SMIF	0	R/W	Smart Card Interface Mode Select When this bit is set to 1, smart card interface moc selected. 0: Normal asynchronous or clock synchronous mo 1: Smart card interface mode

Asynchronous mode	$B = \frac{\phi \times 10^6}{64 \times 2^{2n-1} \times (N+1)}$	Error (%) = $\{\frac{\phi \times 10^6}{B \times 64 \times 2^{2n-1} \times (N+$
Clock 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 (%) = $\{\frac{\phi \times 10^6}{B \times S \times 2^{2n+1} \times (N$

[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 table.

SMR Setting

CKS1	CKS0	n
0	0	0
0	1	1
1	0	2
1	1	3

SMR Setting

BCP1	BCP0	S
0	0	3
0	1	6
1	0	3
1	1	2

Table 13.3 shows sample N settings in BRR in normal asynchronous mode. Table 13.4 s maximum bit rate settable for each frequency. Table 13.6 and 13.8 show sample N settin BRR in clock synchronous mode and smart card interface mode, respectively. In smart c interface mode, the number of basic clock cycles S in a 1-bit data transfer time can be se For details, see section 13.7.4, Receive Data Sampling Timing and Reception Margin. T and 13.7 show the maximum bit rates with external clock input.

2400	1	64	0.16	1	80	–0.47	1	110	–
4800	0	129	0.16	0	162	0.15	0	220	0
9600	0	64	0.16	0	80	–0.47	0	110	–
19200	0	32	–1.36	0	40	–0.76	0	54	0
31250	0	19	0.00	0	24	0.00	0	33	0
38400	0	15	1.73	0	19	1.73	0	27	–

Note: Make the settings so that the error does not exceed 1%.

Table 13.4 Maximum Bit Rate for Each Frequency (Asynchronous Mode)

ϕ (MHz)	Maximum Bit Rate (bit/s)	n	N
20	625000	0	0
25	781250	0	0
34	1062500	0	0

Table 13.5 Maximum Bit Rate with External Clock Input (Asynchronous Mode)

ϕ (MHz)	External Input Clock (MHz)	Maximum Bit Rate (bi
20	5.0000	312500
25	6.2500	390625
34	8.0000	531250

5 k	1	249	2	74	2	1
10 k	1	124	1	149	1	2
25 k	0	199	0	239	1	8
50 k	0	99	0	119	0	1
100 k	0	49	0	59	0	8
250 k	0	19	0	23	0	3
500 k	0	9	0	11	0	1
1 M	0	4	0	5		
2.5 M	0	1				
5 M	0	0*				

[Legend]

Blank: Setting prohibited.

—: Can be set, but there will be a degree of error.

*: Continuous transfer or reception is not possible.

Bit Rate (bit/s)	Operating Frequency φ (MHz)											
	20.00			21.4272			25			34		
	n	N	Error (%)	n	N	Error(%)	n	N	Error (%)	n	N	
9600	0	2	–6.65	0	2	0.00	0	3	–12.49	0	4	

Table 13.9 Maximum Bit Rate for Each Frequency (Smart Card Interface Mode, S

φ (MHz)	Maximum Bit Rate (bit/s)	n	N
20.00	26882	0	0
21.4272	28800	0	0
25.00	33602	0	0
34.00	45699	0	0

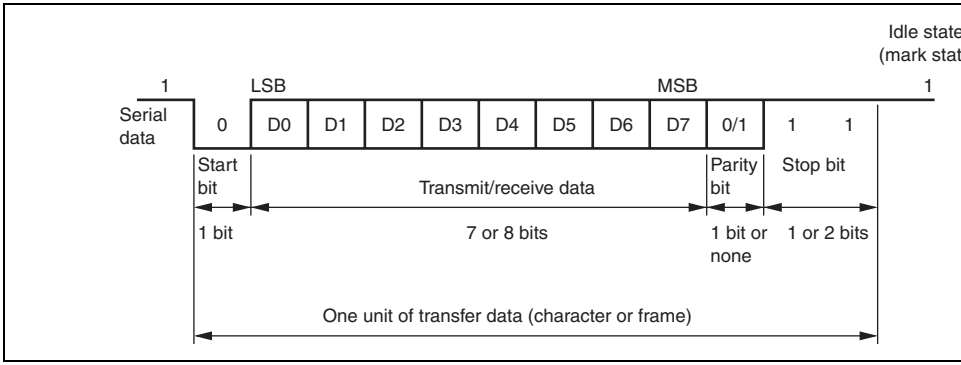


Figure 13.2 Data Format in Asynchronous Communication (Example with 8-Bit Data, Parity, Two Stop Bits)

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

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

M: Reception margin (%)
N: Ratio of bit rate to clock (N = 16)
D: Clock duty (D = 0.5 to 1.0)
L: Frame length (L = 9 to 12)
F: Absolute value of clock rate deviation

Assuming values of F = 0 and D = 0.5 in formula (1), the reception margin is determined formula below.

$$M = \{0.5 - 1/(2 \times 16)\} \times 100 \quad [\%] = 46.875\ \%$$

However, this is only the computed value, and a margin of 20% to 30% should be allow system design.

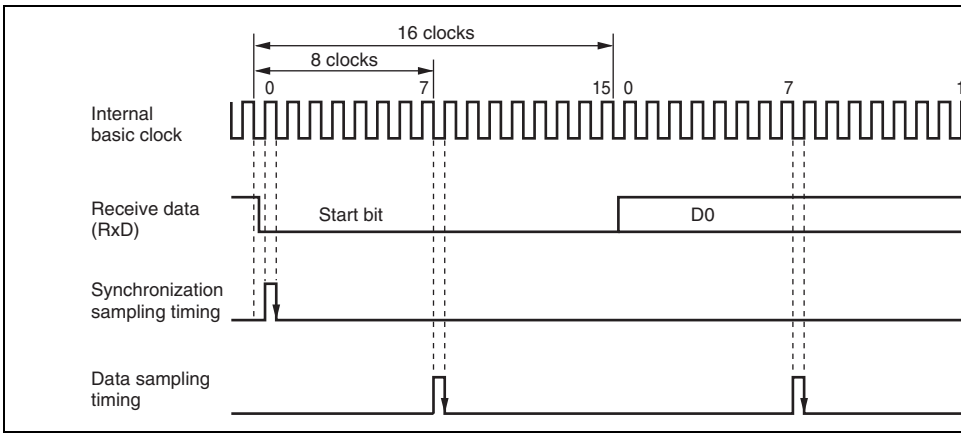


Figure 13.3 Receive Data Sampling Timing in Asynchronous Mode

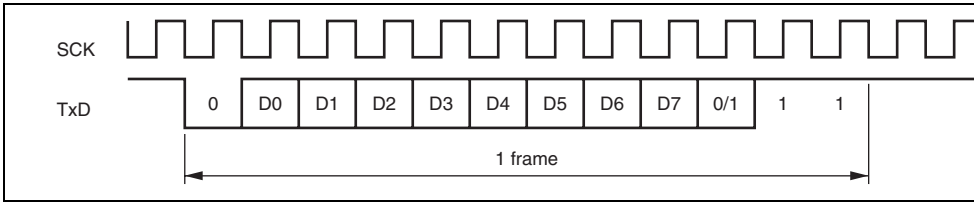
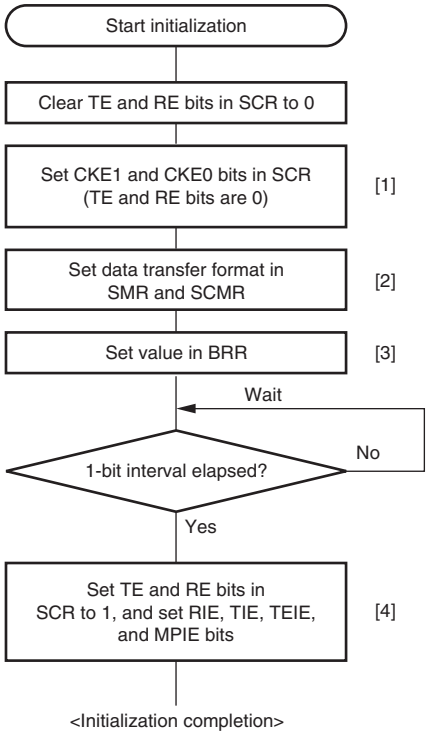


Figure 13.4 Relation between Output Clock and Transmit Data Phase (Asynchronous Mode)



[1] Set the clock selection in SCR. Be sure to clear bits RIE, TIE, TEIE, and MPIE, and bits TE and RE, to 0.

When the clock is selected in asynchronous mode, it is output immediately after SCR settings are made.

[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.

Figure 13.5 Sample SCI Initialization Flowchart

be enabled.

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 T interrupt request is generated.

Figure 13.7 shows a sample flowchart for transmission in asynchronous mode.

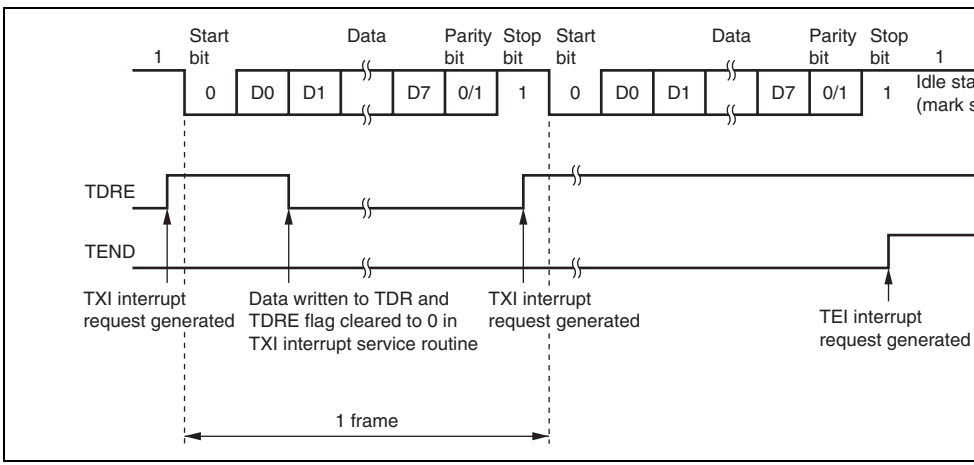
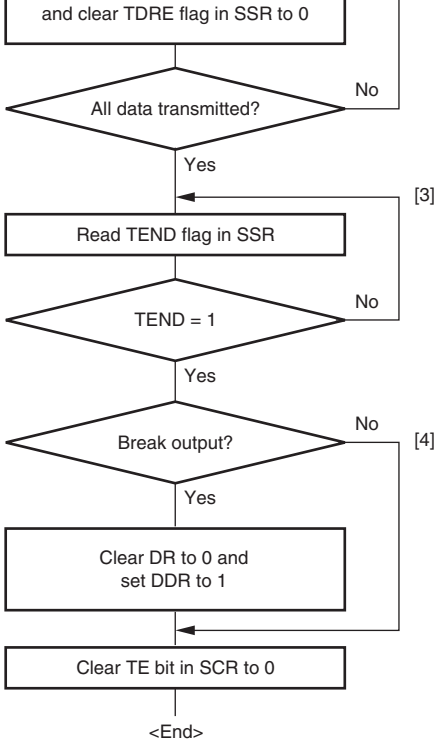


Figure 13.6 Example of Operation in Transmission in Asynchronous Mode (Example with 8-Bit Data, Parity, One Stop Bit)



[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 clear the TDRE flag to 0. However, the TDRE flag is checked and cleared automatically when the DTC is initiated by a transmit data empty interrupt (TXI) request and writes data to TDR.

[4] Break output at the end of serial transmission:
To output a break in serial transmission, set DDR for the port corresponding to the TxD pin to 1, clear DR to 0, then clear the TE bi in SCR to 0.

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 (when the stop bit is 0) is detected, 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 finishes successfully, the RDRF bit in SSR is set to 1, and receive data is transferred to RDR. If the RIE bit in SCR is set to 1 at this time, an RXI interrupt requ generated. Because the RXI interrupt routine reads the receive data transferred to RDI reception of the next receive data has finished, continuous reception can be enabled.

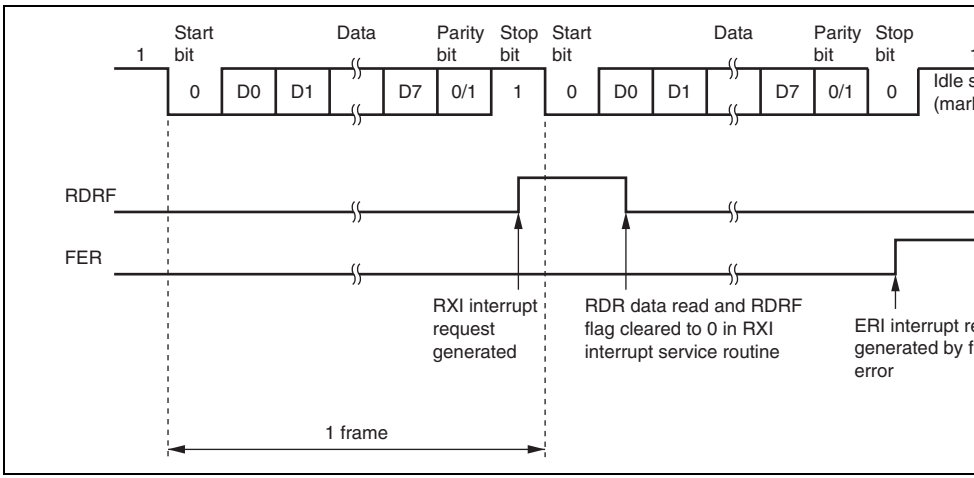
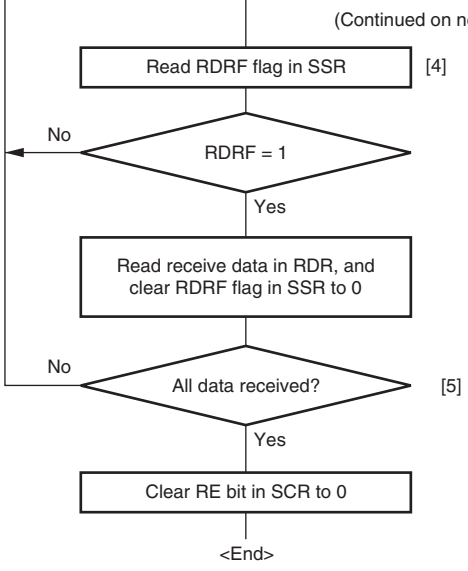


Figure 13.8 Example of SCI Operation in Reception (Example with 8-Bit Data, Parity, One Stop Bit)

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 + fram
1	1	0	1	Lost	Overrun error + parit
0	0	1	1	Transferred to RDR	Framing error + parit
1	1	1	1	Lost	Overrun error + fram parity error

Note: * The RDRF flag retains the state it had before data reception.



1. In the case of a framing error, a break can be detected by reading t value of the input port correspondir the RxD pin.

[4] SCI status check and receive data Read SSR and check that RDRF = then read the receive data in RDR clear the RDRF flag to 0. Transitio the RDRF flag from 0 to 1 can also identified by an RXI interrupt.

[5] Serial reception continuation proce To continue serial reception, before stop bit for the current frame is received, read the RDRF flag, read RDR, and clear the RDRF flag to 0 However, the RDRF flag is cleared automatically when the DTC is initia by an RXI interrupt and reads data RDR.

[Legend]
∨ : Logical add (OR)

Figure 13.9 Sample Serial Reception Flowchart (1)

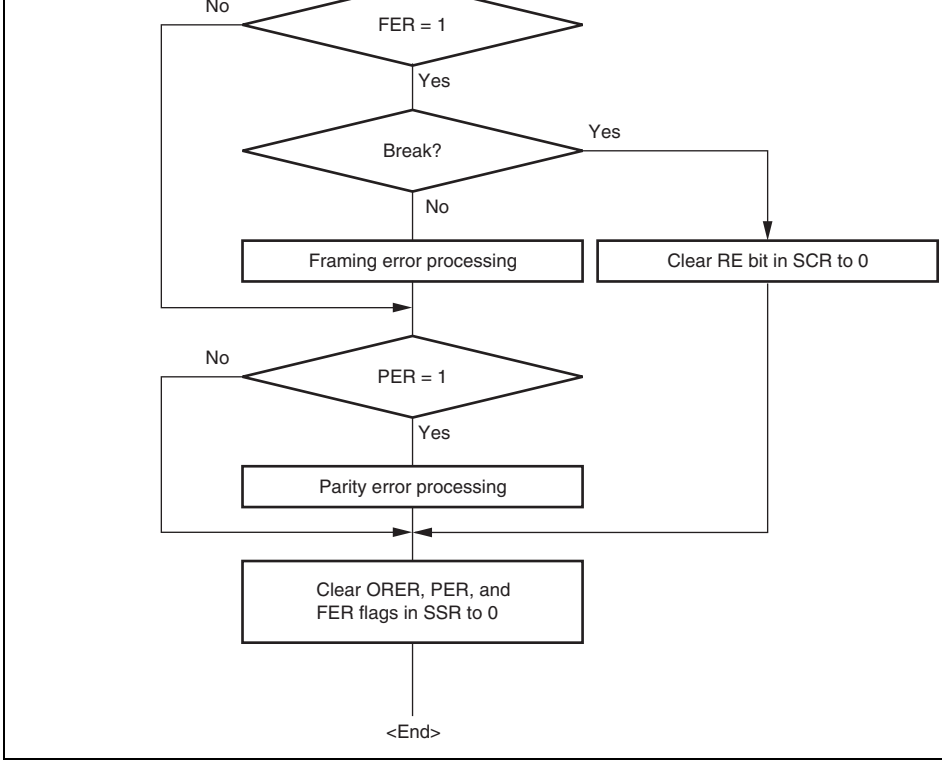


Figure 13.9 Sample Serial Reception Flowchart (2)

transmission cycle, and if the multiprocessor bit is 0, the cycle is a data transmission cycl
13.10 shows an example of inter-processor communication using the multiprocessor form
transmitting station first sends the ID code of the receiving station with which it wants to
serial communication as data with a 1 multiprocessor bit added. It then sends transmit dat
with a 0 multiprocessor bit added. When data with a 1 multiprocessor bit is received, the
station compares that data with its own ID. The station whose ID matches then receives th
sent next. Stations whose ID does not match continue to skip data until data with a 1
multiprocessor bit is again received.

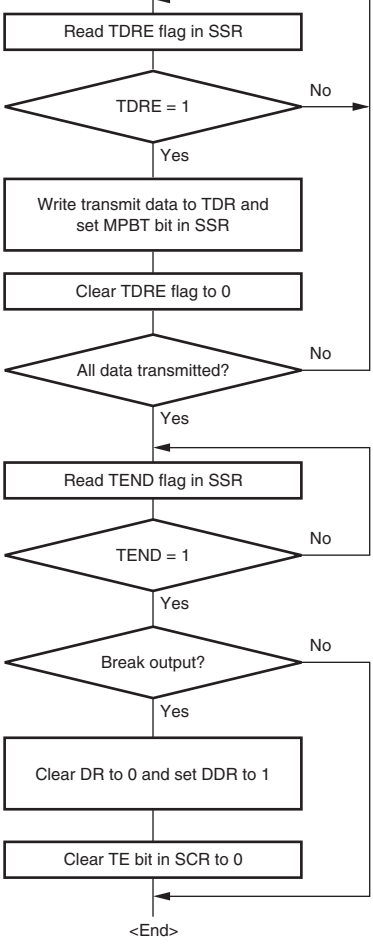
The SCI uses the MPIE bit in SCR to implement this function. When the MPIE bit is set t
transfer of receive data from RSR to RDR, error flag detection, and setting the RDRF, FE
ORER status flags in SSR to 1 are prohibited until data with a 1 multiprocessor bit is rece
reception of a receive character 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 invalid. All other bit s
are the same as those in normal asynchronous mode. The clock used for multiprocessor
communication is the same as that in normal asynchronous mode.

		ID transmission cycle = receiving station specification	Data transmission cycle = Data transmission to 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)



After the TE bit is set to 1, a frame of 1s is output, and transmission is enabled.

[2] SCI status check and transmit data write:
Read SSR and check that the TDRE flag is set to 1, then write transmit data to TDR. Set the MPBT bit in SSR to 0 or 1. Finally, clear the TDRE flag to 0.

[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. However, the TDRE flag is checked and cleared automatically when the DTC is initiated by a transmit data empty interrupt (TXI) request and writes data to TDR.

[4] Break output at the end of serial transmission:
To output a break in serial transmission, set port DDR to 1, clear DR to 0, and then clear the TE bit in SCR to 0.

Figure 13.11 Sample Multiprocessor Serial Transmission Flowchart

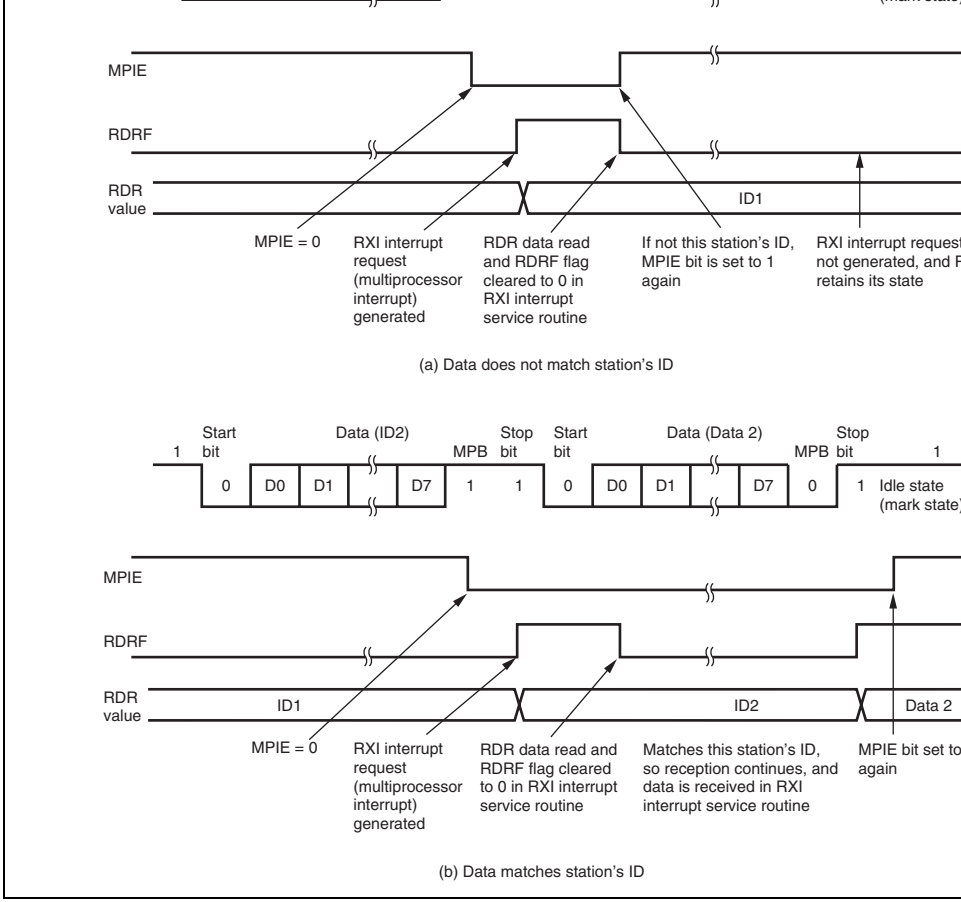
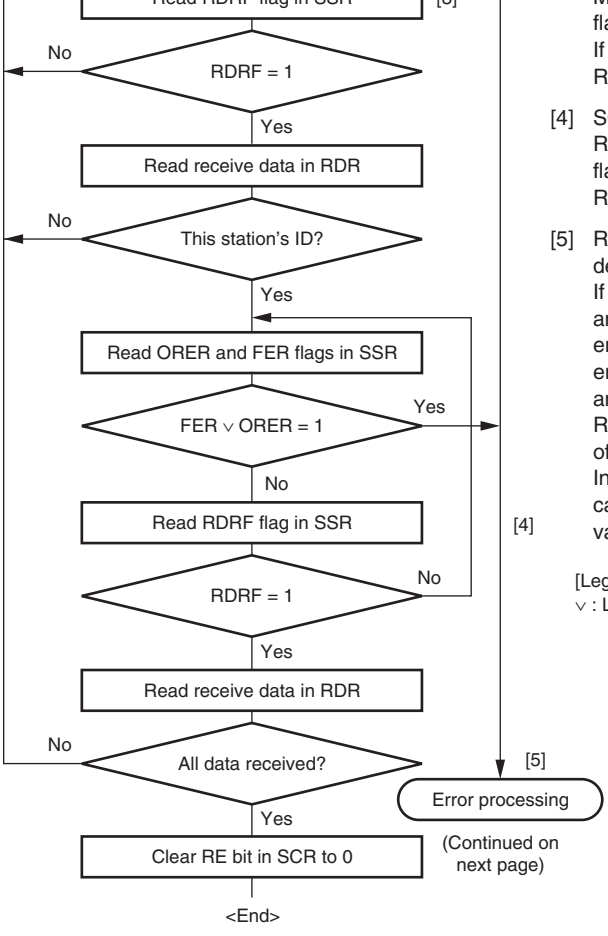


Figure 13.12 Example of SCI Operation in Reception (Example with 8-Bit Data, Multiprocessor Bit, One Stop Bit)



[3] MPIE bit to 1 again, and clear the
flag to 0.
If the data is this station's ID, clear
RDRF flag to 0.

[4] SCI status check and data recepti
Read SSR and check that the RDR
flag is set to 1, then read the data
RDR.

[5] Receive error processing and brea
detection:
If a receive error occurs, read the
and FER flags in SSR to identify th
error. After performing the approp
error processing, ensure that the
and FER flags are all cleared to 0.
Reception cannot be resumed if ei
of these flags is set to 1.
In the case of a framing error, a br
can be detected by reading the Rx
value.

[Legend]
∨ : Logical add (OR)

Figure 13.13 Sample Multiprocessor Serial Reception Flowchart (1)

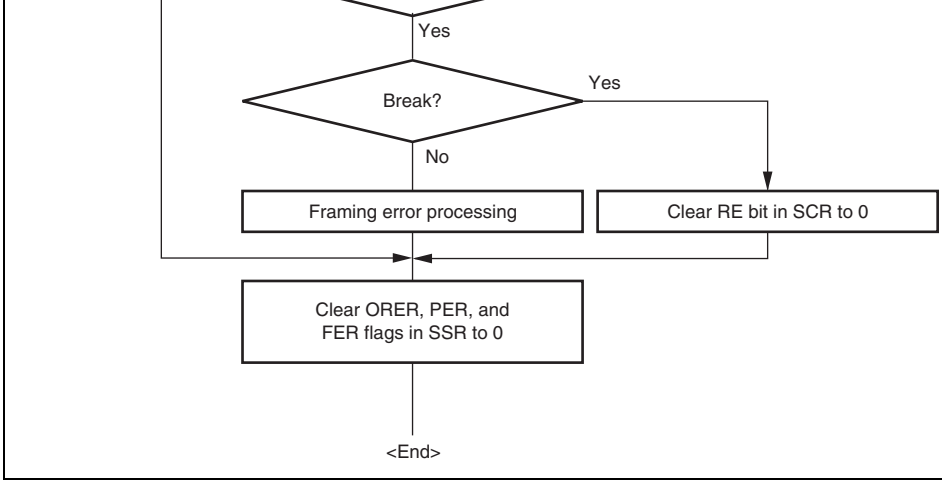


Figure 13.13 Sample Multiprocessor Serial Reception Flowchart (2)

previous receive data can be read during reception, enabling continuous data transfer.

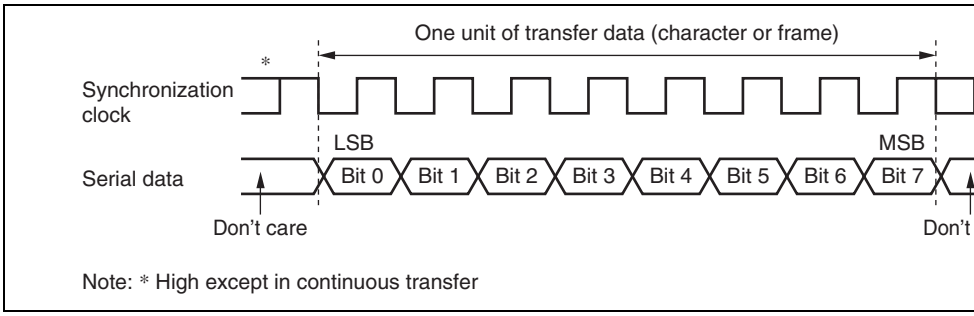
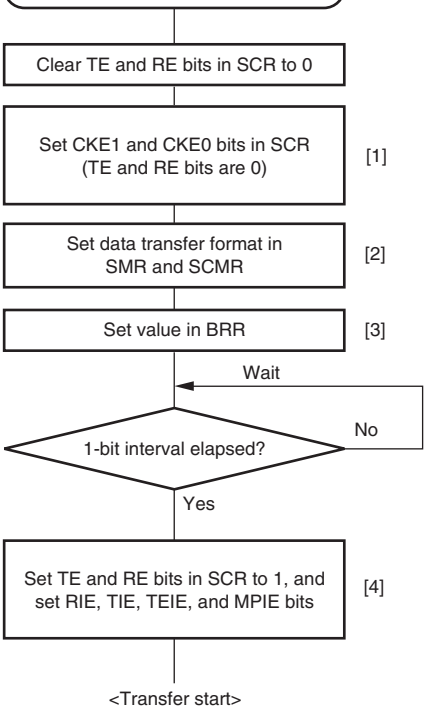


Figure 13.14 Data Format in Synchronous Communication (LSB-First)

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 the and CKE0 bits in SCR. When the SCI is operated on an internal clock, the synchronizatio is output from the SCK pin. Eight synchronization clock pulses are output in the transfer character, and when no transfer is performed the clock is fixed high.



to clear bits RIE, TIE, TEIE, and MPI
TE and RE to 0.

[2] Set the data transfer format in SMR a
SCMR.

[3] Write a value corresponding to the bi
rate to BRR. This step is not necess
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 t
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

the current transmit data has finished, continuous transmission can be enabled.

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 last bit.
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 TxD pin mainta
   output state of the last bit. If the TEIE bit in SCR is set to 1 at this time, a TEI interru
   is generated. The SCK pin is fixed high.

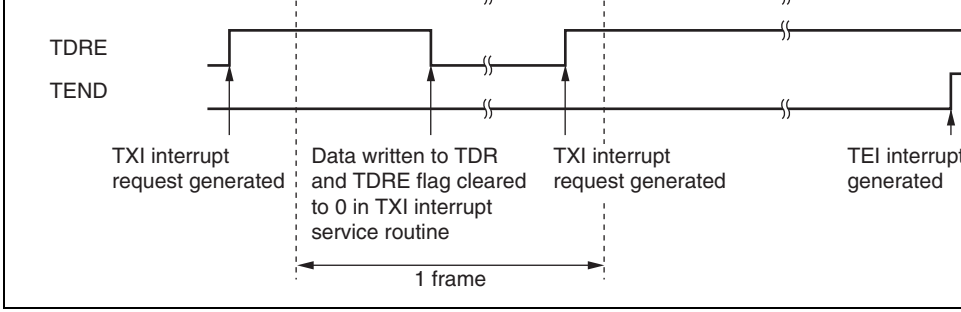
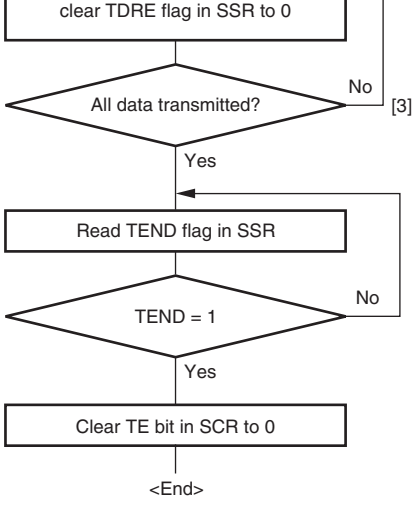


Figure 13.16 Sample SCI Transmission Operation in Clock Synchronous M



confirm that writing is possible, then write data to TDR, and then clear the TDRE flag to 0.
However, the TDRE flag is checked and cleared automatically when the DTC is initiated by a transmit data empty interrupt (TXI) request and writes data to TDR.

Figure 13.17 Sample Serial Transmission Flowchart

3. If reception finishes successfully, the RDRF bit in SSR is set to 1, and receive data is transferred to RDR. If the RIE bit in SCR is set to 1 at this time, an RXI interrupt req generated. Because the RXI interrupt routine reads the receive data transferred to RD reception of the next receive data has finished, continuous reception can be enabled.

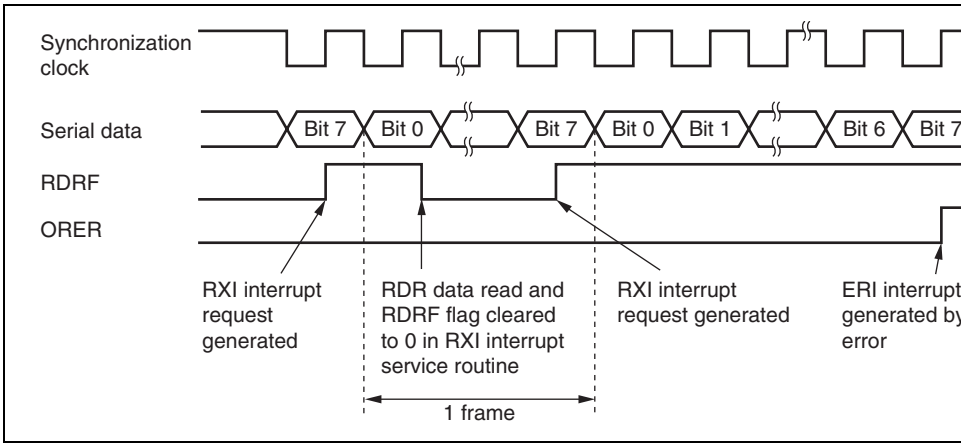
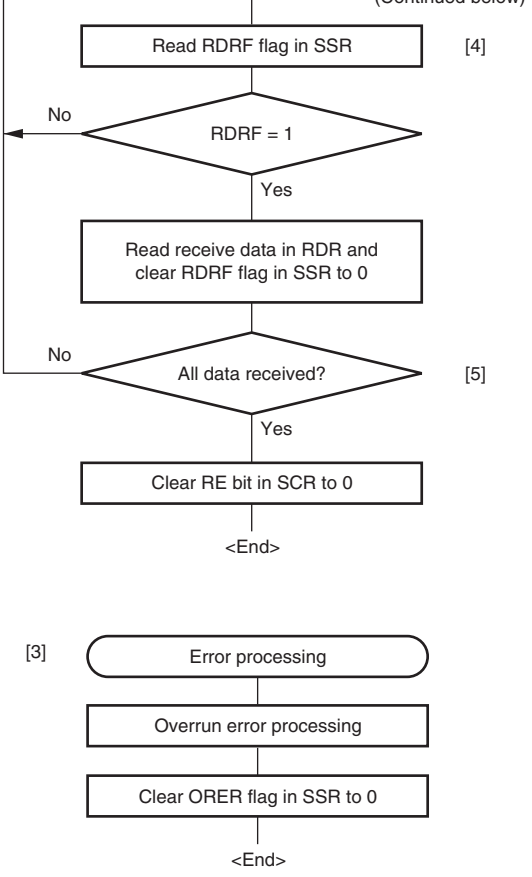


Figure 13.18 Example of SCI Receive Operation in Clock Synchronous Mo

Reception cannot be resumed while a receive error flag is set to 1. Accordingly, clear the FER, PER, and RDRF bits to 0 before resuming reception. Figure 13.19 shows a sample for serial data reception.

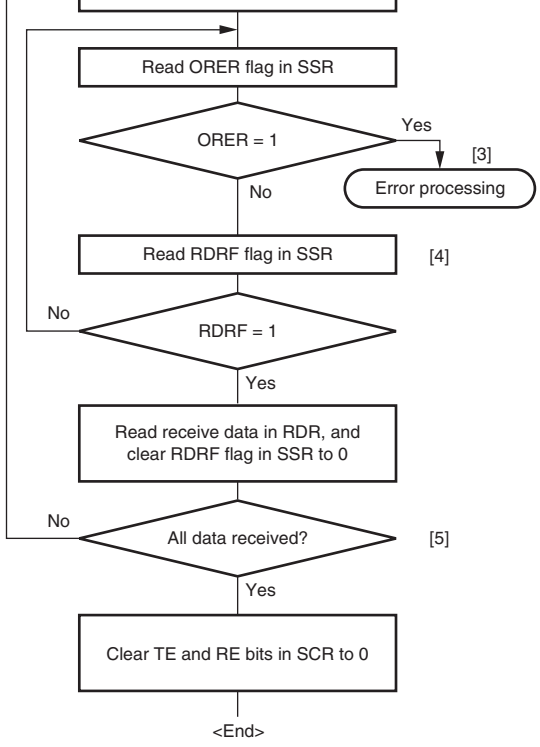


Read SSR and check that the RD flag is set to 1, then read the rece data in RDR and clear the RDRF to 0.
Transition of the RDRF flag from can also be identified by an RXI interrupt.

[5] Serial reception continuation procedure:
To continue serial reception, befo the MSB (bit 7) of the current fran received, reading the RDRF flag, reading RDR, and clearing the RI flag to 0 should be finished.
However, the RDRF flag is cleare automatically when the DTC is initiated by a receive data full inte (RXI) and reads data from RDR.

Figure 13.19 Sample Serial Reception Flowchart

simultaneously set the TE and RE bits to 1 with a single instruction.



Note: 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 simultaneously.

[3] Receive error processing:
If a receive error occurs, read th
ORER flag in SSR, and after
performing the appropriate error
processing, clear the ORER flag
Transmission/reception cannot b
resumed if the ORER flag is set

[4] SCI status check and receive da
read:
Read SSR and check that the R
flag is set to 1, then read the rec
data in RDR and clear the RDRF
to 0. Transition of the RDRF fla
0 to 1 can also be identified by a
interrupt.

[5] Serial transmission/reception
continuation procedure:
To continue serial transmission/
reception, before the MSB (bit 7
the current frame is received, fin
reading the RDRF flag, reading
and clearing the RDRF flag to 0.
before the MSB (bit 7) of the cur
frame is transmitted, read 1 from
TDRE flag to confirm that writing
possible. Then write data to TDR
clear the TDRE flag to 0.
However, the TDRE flag is chec
and cleared automatically when
DTC is initiated by a transmit da
empty interrupt (TXI) request an
writes data to TDR. Similarly, th
RDRF flag is cleared automatica
when the DTC is initiated by a re
data full interrupt (RXI) and read
data from RDR.

Figure 13.20 Sample Flowchart of Simultaneous Serial Transmission and Rece

function as an I/O pin. Pull up the data transmission line to Vcc using a resistor. Setting
and TE bits in SCR to 1 with the IC card not connected enables closed transmission/rece
allowing self diagnosis. To supply the IC card with the clock pulses generated by the SC
the SCK pin output to the CLK pin of the IC card. A reset signal can be supplied via the
port of this LSI.

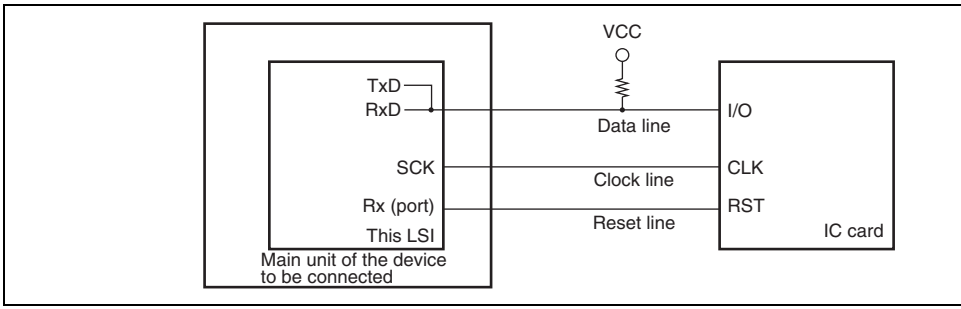


Figure 13.21 Pin Connection for Smart Card Interface

13.7.2 Data Format (Except in Block Transfer Mode)

Figure 13.22 shows the data transfer formats in smart card interface mode.

- One frame contains 8-bit data and a parity bit in asynchronous mode.
- During transmission, at least 2 etu (elementary time unit: time required for transferri
is secured as a guard time after the end of the parity bit before the start of the next fr
- If a parity error is detected during reception, a low error signal is output for 1 etu aft
has passed from the start bit.
- If an error signal is sampled during transmission, the same data is automatically re-t
after two or more etu.

Figure 13.22 Data Formats in Normal Smart Card Interface Mode

For communication with the IC cards of the direct convention and inverse convention typ
follow the procedure below.

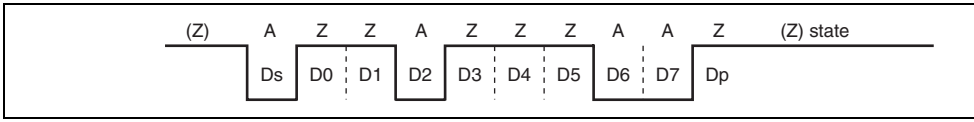


Figure 13.23 Direct Convention (SDIR = SINV = O/$\overline{E}$ = 0)

For the direct convention type, logic levels 1 and 0 correspond to states Z and A, respecti
data is transferred with LSB-first as the start character, as shown in figure 13.23. Therefo
in the start character in the figure is H'3B. When using the direct convention type, write 0
the SDIR and SINV bits in SCMR. Write 0 to the O/$\overline{E}$ bit in SMR in order to use even pa
which is prescribed by the smart card standard.

Figure 13.24 Inverse Convention (SDIR = SINV = O/$\overline{E}$ = 1)

- If a parity error is detected during reception, no error signal is output. Since the PER
SSR is set by error detection, clear the bit before receiving the parity bit of the next f
- During transmission, at least 1 etu is secured as a guard time after the end of the pari
before the start of the next frame.
- Since the same data is not re-transmitted during transmission, the TEND flag in SSR
11.5 etu after transmission start.
- Although the ERS flag in block transfer mode displays the error signal status as in n
smart card interface mode, the flag is always read as 0 because no error signal is tran

$$M = \left| (0.5 - \frac{1}{2N}) - (L - 0.5)\,F - \frac{|D - 0.5|}{N} (1 + F) \right| \times 100\ [\%] \quad \cdots \text{ Formula (1}$$

M: Reception margin (%)
N: Ratio of bit rate to clock (N = 32, 64, 372, 256)
D: Clock duty (D = 0 to 1.0)
L: Frame length (L = 10)
F: Absolute value of clock rate deviation

Assuming values of F = 0, D = 0.5, and N = 372 in formula (1), the reception margin is determined by the formula below.

$$M = (0.5 - 1/2 \times 372) \times 100\ [\%] = 49.866\%$$

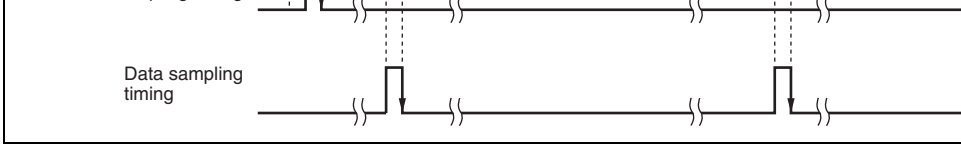


Figure 13.25 Receive Data Sampling Timing in Smart Card Interface Mo
(When Clock Frequency is 372 Times the Bit Rate)

13.7.5 Initialization

Before starting transmitting and receiving data, initialize the SCI using the following pr
Initialization is also necessary before switching from transmission to reception and vice

1. Clear the TE and RE bits in SCR to 0.
2. Clear the error flags ORER, ERS, and PER in SSR to 0.
3. Set the GM, BLK, O/$\overline{E}$, BCP1, BCP0, CKS1, and CKS0 bits in SMR appropriately.
the PE bit to 1.
4. Set the SMIF, SDIR, and SINV bits in SCMR appropriately. When the SMIF bit is s
TxD and RxD pins are changed from port pins to SCI pins, placing the pins into hig
impedance state.
5. Set the value corresponding to the bit rate in BRR.
6. Set the CKE1 and CKE0 bits in SCR appropriately. Clear the TIE, RIE, TE, RE, MP
TEIE bits to 0 simultaneously. When the CKE0 bit is set to 1, the SCK pin is allowe
clock pulses.
7. Set the TIE, RIE, TE, and RE bits in SCR appropriately after waiting for at least 1 bi
Setting prohibited the TE and RE bits to 1 simultaneously except for self diagnosis.

re-transmitted. Figure 13.26 shows the data re-transfer operation during transmission.

1. If an error signal from the receiving end is sampled after one frame of data has been transmitted, the ERS bit in SSR is set to 1. Here, an ERI interrupt request is generated RIE bit in SCR is set to 1. Clear the ERS bit to 0 before the next parity bit is sampled.
2. For the frame in which an error signal is received, the TEND bit in SSR is not set to 1 re-transferred from TDR to TSR allowing automatic data retransmission.
3. If no error signal is returned from the receiving end, the ERS bit in SSR is not set to 1 case, one frame of data is determined to have been transmitted including re-transfer, a TEND bit in SSR is set to 1. Here, a TXI interrupt request is generated if the TIE bit i set to 1. Writing transmit data to TDR starts transmission of the next data.

Figure 13.28 shows a sample flowchart for transmission. All the processing steps are automatically performed using a TXI interrupt request to activate the DTC. In transmissio TEND and TDRE flags in SSR are simultaneously set to 1, thus generating a TXI interru when TIE in SCR is set. This activates the DTC by a TXI request thus allowing transfer o transmit data if the TXI interrupt request is specified as a source of DTC activation befor The TDRE and TEND flags are automatically cleared to 0 at data transfer by the DTC. If occurs, the SCI automatically re-transmits the same data. During re-transmission, TEND as 0, thus not activating the DTC. Therefore, the SCI and DTC automatically transmit the specified number of bytes, including re-transmission in the case of error occurrence. How ERS flag is not automatically cleared; the ERS flag must be cleared by previously setting bit to 1 to enable an ERI interrupt request to be generated at error occurrence.

When transmitting/receiving data using the DTC, be sure to set and enable it prior to mak settings. See section 7, Data Transfer Controller (DTC) for DTC settings.

Figure 13.26 Data Re-transfer Operation in SCI Transmission Mode

Note that the TEND flag is set in different timings depending on the GM bit setting in S which is shown in figure 13.27.

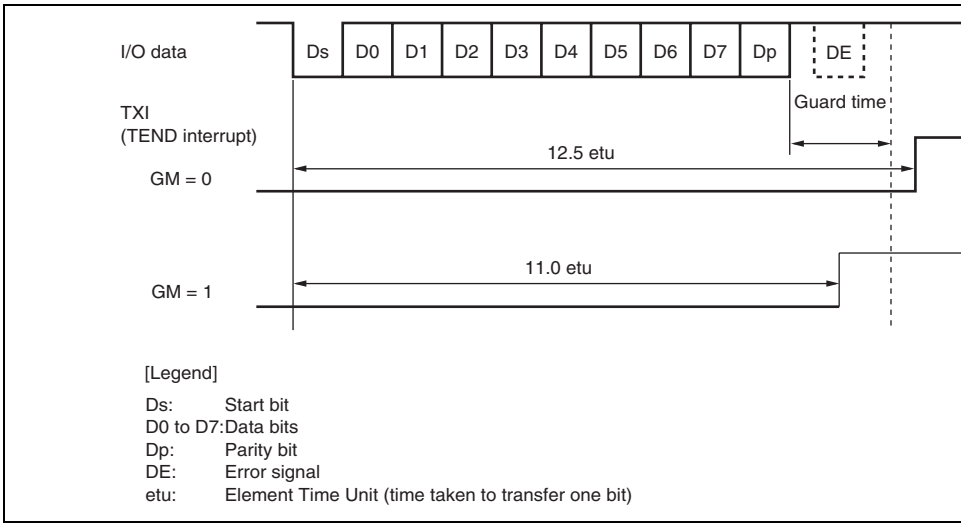


Figure 13.27 TEND Flag Set Timings during Transmission

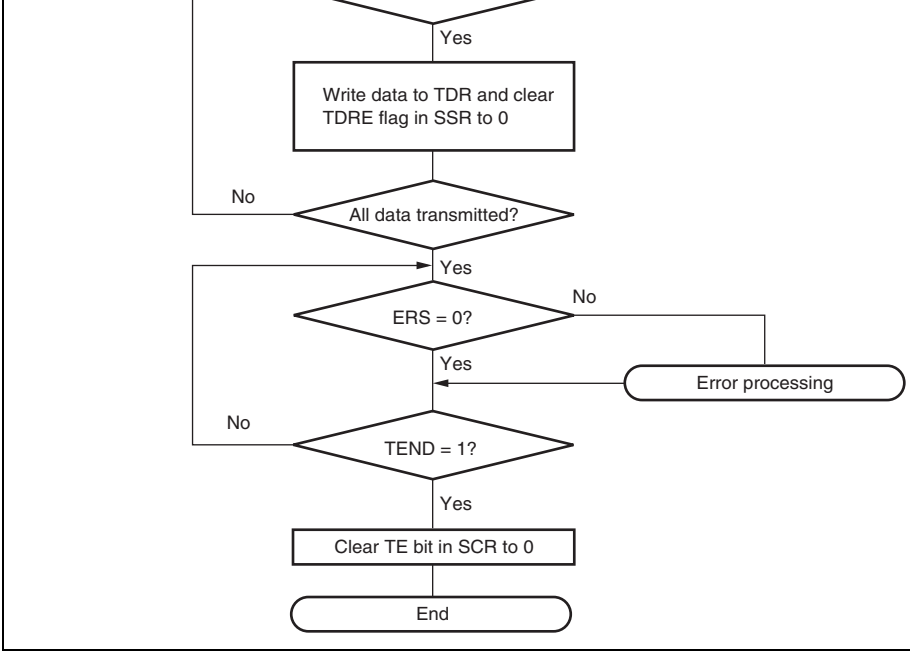


Figure 13.28 Sample Transmission Flowchart

to have been received successfully, and the RDRF bit in SSR is set to 1. Here, an RX request is generated if the RIE bit in SCR is set.

Figure 13.30 shows a sample flowchart for reception. All the processing steps are autom performed using an RXI interrupt request to activate the DTC. In reception, setting the R allows an RXI interrupt request to be generated when the RDRF flag is set to 1. This act DTC by an RXI request thus allowing transfer of receive data if the RXI interrupt reque specified as a source of DTC activate beforehand. The RDRF flag is automatically clear data transfer by DTC. If an error occurs during reception, i.e., either the ORER or PER f to 1, a transmit/receive error interrupt (ERI) request is generated and the error flag must cleared. If an error occurs, DTC is not activated and receive data is skipped, therefore, th of bytes of receive data specified in DTC are transferred. Even if a parity error occurs an set to 1 in reception, receive data is transferred to RDR, thus allowing the data to be read

Note: For operations in block transfer mode, see section 13.4, Operation in Asynchron

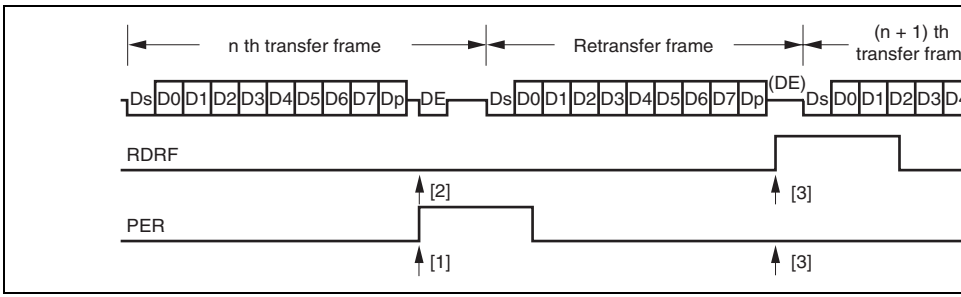


Figure 13.29 Data Re-transfer Operation in SCI Reception Mode

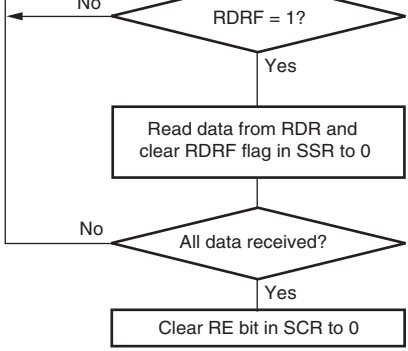


Figure 13.30 Sample Reception Flowchart

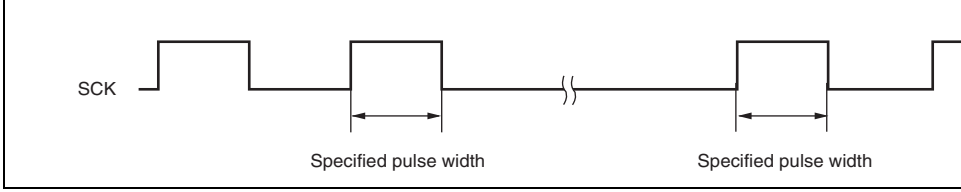


Figure 13.31 Clock Output Fixing Timing

At power-on and transitions to/from software standby mode, use the following procedur
the appropriate clock duty ratio.

At Power-On:

To secure the appropriate clock duty ratio simultaneously with power-on, use the follow
procedure.

1. Initially, port input is enabled in the high-impedance state. To fix the potential level,
   pull-up or pull-down resistor.
2. Fix the SCK pin to the specified output using the CKE1 bit in SCR.
3. Set SMR and SCMR to enable smart card interface mode.
4. Set the CKE0 bit in SCR to 1 to start clock output.

At Transition from Software Standby Mode to Smart Card Interface Mode:

1. Cancel software standby mode.
2. Write 1 to the CKE0 bit in SCR to start clock output. A clock signal with the appropri ratio is then generated.

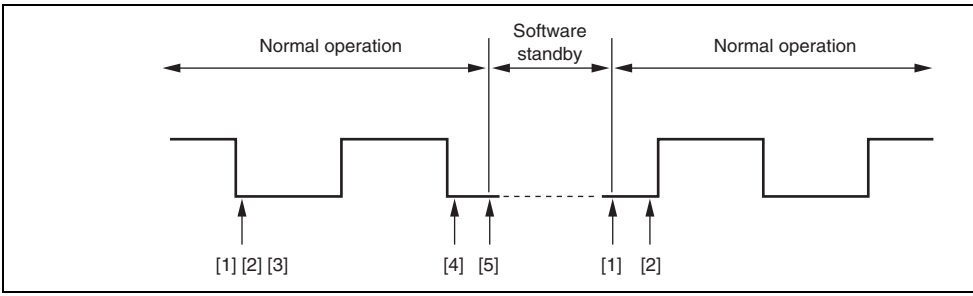


Figure 13.32 Clock Stop and Restart Procedure

allow data transfer. The TDRE flag is automatically cleared to 0 at data transfer by the D

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 inter activate the DTC to allow data transfer. The RDRF flag is automatically cleared to 0 at c transfer by the DTC.

A TEI interrupt is requested when the TEND flag is set to 1 while the TEIE bit is set to interrupt and a TXI interrupt are requested simultaneously, the TXI interrupt has priority acceptance. However, note that if the TDRE and TEND flags are cleared simultaneously TXI interrupt routine, the SCI cannot branch to the TEI interrupt routine later.

Table 13.12 SCI Interrupt Sources

Channel	Name	Interrupt Source	Interrupt Flag	DTC Activation
3	ERI3	Receive error	ORER, FER, PER	Not possible
	RXI3	Receive data full	RDRF	Possible
	TXI3	Transmit data empty	TDRE	Possible
	TEI3	Transmit end	TEND	Not possible
1	ERI1	Receive error	ORER, FER, PER	Not possible
	RXI1	Receive data full	RDRF	Possible
	TXI1	Transmit data empty	TDRE	Possible
	TEI1	Transmit end	TEND	Not possible

	TXI3	Transmit data empty	TEND	Possible
1	ERI1	Receive error, error signal detection	ORER, PER, ERS	Not possible
	RXI1	Receive data full	RDRF	Possible
	TXI1	Transmit data empty	TEND	Possible

Data transmission/reception using the DTC is also possible in smart card interface mode, to in the normal SCI mode. In transmission, the TEND and TDRE flags in SSR are simul set to 1, thus generating a TXI interrupt request. This activates the DTC by a TXI interru thus allowing transfer of transmit data if the TXI interrupt request is specified as a source activation beforehand. The TDRE and TEND flags are automatically cleared to 0 at data by the DTC. If an error occurs, the SCI automatically re-transmits the same data. During transmission, the TEND flag remains as 0, thus not activating the DTC. Therefore, the SC DTC automatically transmit the specified number of bytes, including re-transmission in t error occurrence. However, the ERS flag in SSR, which is set at error occurrence, is not automatically cleared; the ERS flag must be cleared by previously setting the RIE bit in S to enable an ERI interrupt request to be generated at error occurrence.

When transmitting/receiving data using the DTC, be sure to set and enable the DTC prior making SCI settings. For DTC settings, see section 7, Data Transfer Controller (DTC).

In reception, an RXI interrupt request is generated when the RDRF flag in SSR is set to 1 activates the DTC by an RXI interrupt request thus allowing transfer of receive data if the interrupt request is specified as a source of DTC activation beforehand. The RDRF flag is automatically cleared to 0 at data transfer by the DTC. If an error occurs, the RDRF flag i but the error flag is set. Therefore, the DTC is not activated and an ERI interrupt request to the CPU instead; the error flag must be cleared.

When framing error detection is performed, a break can be detected by reading the RxD directly. In a break, the input from the RxD pin becomes all 0s, and so the FER flag in S and the PER flag may also be set. Note that, since the SCI continues the receive operatio after receiving a break, even if the FER flag is cleared to 0, it will be set to 1 again.

13.9.3 Mark State and Break Sending

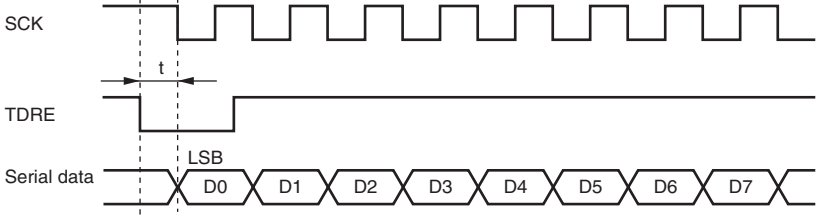
When the TE bit in SCR is 0, the TxD pin is used as an I/O port whose direction (input o and level are determined by DR and DDR of the port. This can be used to set the TxD pi state (high level) or send a break during serial data transmission. To maintain the comm line at mark state until TE is set to 1, set both DDR and DR to 1. Since the TE bit is clea this point, the TxD pin becomes an I/O port, and 1 is output from the TxD pin. To send a during serial transmission, first set DDR to 1 and DR to 0, and then clear the TE bit to 0 TE bit is cleared to 0, the transmitter is initialized regardless of the current transmission TxD pin becomes an I/O port, and 0 is output from the TxD pin.

13.9.4 Receive Error Flags and Transmit Operations (Clock Synchronous Mode

Transmission cannot be started when a receive error flag (ORER, FER, or RER) in SSR even if the TDRE flag in SSR is cleared to 0. Be sure to clear the receive error flags to 0 starting transmission. Note also that the receive error flags cannot be cleared to 0 even if bit in SCR is cleared to 0.

13.9.5 Relation between Writing to TDR and TDRE Flag

Data can be written to TDR irrespective of the TDRE flag status in SSR. However, if the is written to TDR when the TDRE flag is 0, that is, when the previous data has not been transferred to TSR yet, the previous data in TDR is lost. Be sure to write transmit data to after verifying that the TDRE flag is set to 1.



Note: When external clock is supplied, t must be more than four clock cycles.

Figure 13.33 Sample Transmission using DTC in Clock Synchronous Mod

Figure 13.34 shows a sample flowchart for mode transition during transmission. Figures
13.36 show the pin states during transmission.

Before making the transition from the transmission mode using DTC transfer to module
software standby mode, stop all transmit operations (TE = TIE = TEIE = 0). Setting TE
1 after mode cancellation generates a TXI interrupt request to start transmission using th

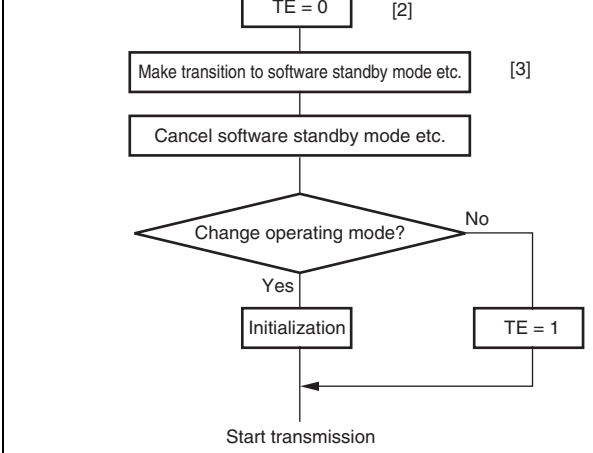


Figure 13.34 Sample Flowchart for Mode Transition during Transmission

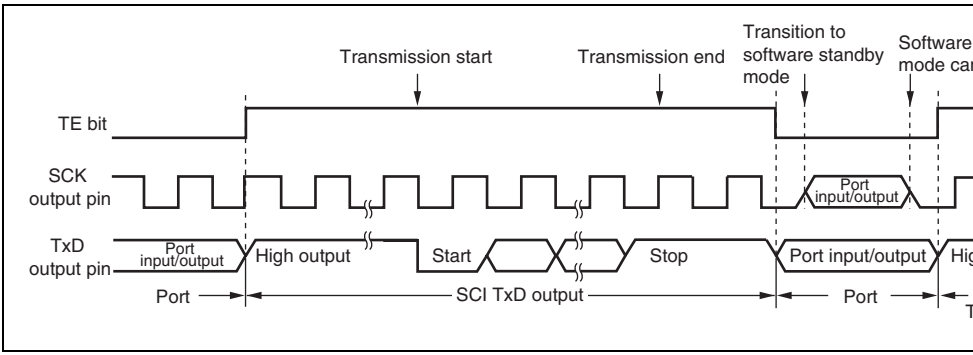


Figure 13.35 Pin States during Transmission in Asynchronous Mode (Internal Clock)

Figure 13.36 Pin States during Transmission in Clock Synchronous Mod (Internal Clock)

Reception: Before making the transition to module stop or software standby mode, stop (RE = 0). RSR, RDR, and SSR are reset. If transition is made during data reception, the received will be invalid.

To receive data in the same reception mode after mode cancellation, set RE to 1, and the reception. To receive data in a different reception mode, initialize the SCI first.

Figure 13.37 shows a sample flowchart for mode transition during reception.

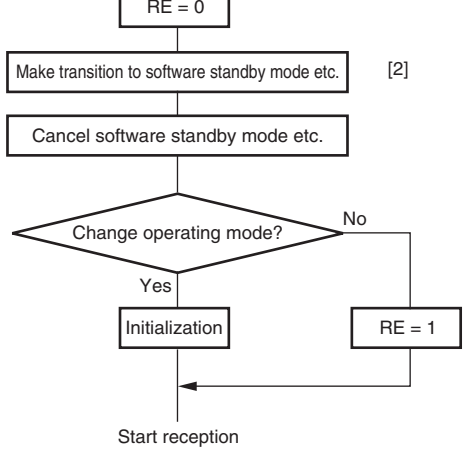


Figure 13.37 Sample Flowchart for Mode Transition during Reception

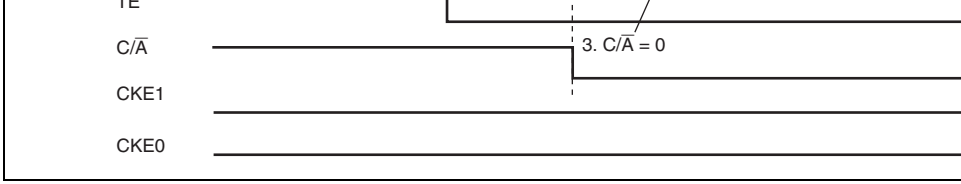


Figure 13.38 Switching from SCK Pins to Port Pins

To prevent the low pulse output that is generated when switching the SCK pins to the po specify the SCK pins for input (pull up the SCK/port pins externally), and follow the pro below with DDR = 1, DR = 1, C/$\overline{A}$ = 1, CKE1 = 0, CKE1 = 0, and TE = 1.

1. End serial data transmission
2. TE bit = 0
3. CKE1 bit = 1
4. C/$\overline{A}$ bit = 0 (switch to port output)
5. CKE1 bit = 0

Figure 13.39 Prevention of Low Pulse Output at Switching from SCK Pins to Po

- CRC code generated for any desired data length in an 8-bit unit
- CRC operation executed on eight bits in parallel
- One of three generating polynomials selectable
- CRC code generation for LSB-first or MSB-first communication selectable

Figure 14.1 is a block diagram of the CRC operation circuit.

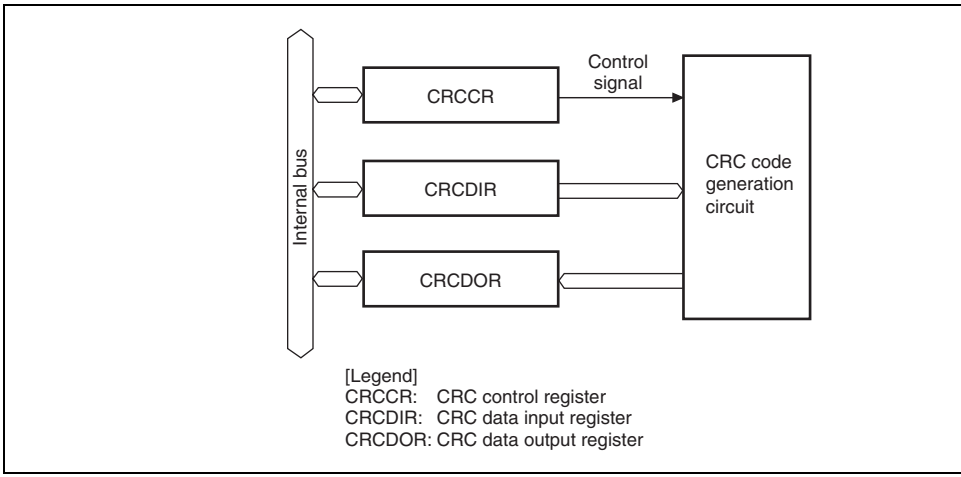


Figure 14.1 Block Diagram of CRC Operation Circuit

CRCCR initializes the CRC operation circuit, switches the operation mode, and selects th
generating polynomial.

Bit	Bit Name	Initial Value	R/W	Description
7	DORCLR	0	W	CRCDOR Clear Setting this bit to 1 clears CRCDOR to H′0000.
6 to 3	—	All 0	R	Reserved The initial value should not be changed.
2	LMS	0	R/W	CRC Operation Switch Selects CRC code generation for LSB-first or MS communication. 0: Performs CRC operation for LSB-first commun The lower byte (bits 7 to 0) is first transmitted w CRCDOR contents (CRC code) are divided int bytes to be transmitted in two parts. 1: Performs CRC operation for MSB-first commu The upper byte (bits 15 to 8) is first transmitted CRCDOR contents (CRC code) are divided int bytes to be transmitted in two parts.
1	G1	0	R/W	CRC Generating Polynomial Select
0	G0	0	R/W	These bits select the polynomial. 00: Reserved 01: $X^8 + X^2 + X + 1$ 10: $X^{16} + X^{15} + X^2 + 1$ 11: $X^{16} + X^{12} + X^5 + 1$

CRCCR are set to G1 = 0 and G0 = 1, respectively, the lower byte of this register contai result.

14.3 CRC Operation Circuit Operation

The CRC operation circuit generates a CRC code for LSB-first/MSB-first communicatio example in which a CRC code for hexadecimal data H'F0 is generated using the $X^{16} + X$ polynomial with the G1 and G0 bits in CRCCR set to B'11 is shown below.

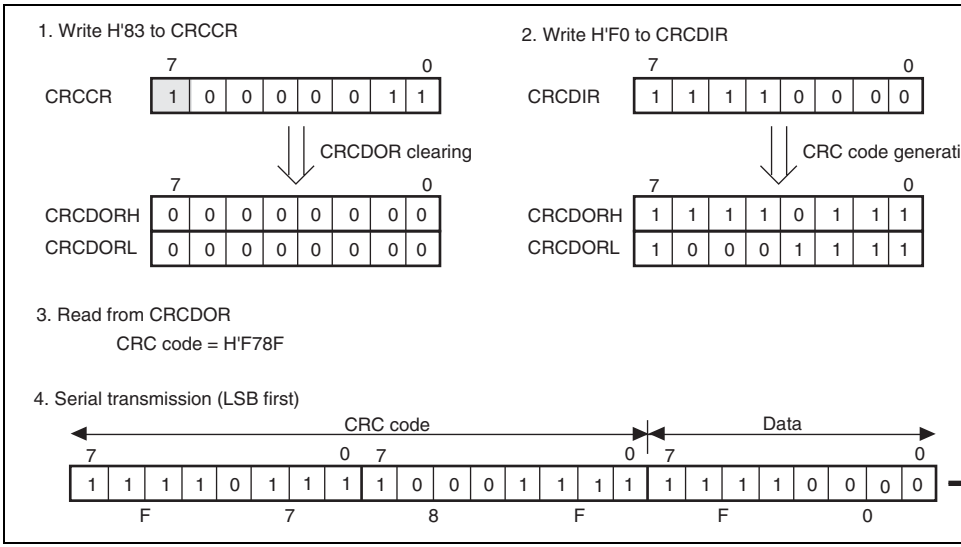


Figure 14.2 LSB-First Data Transmission

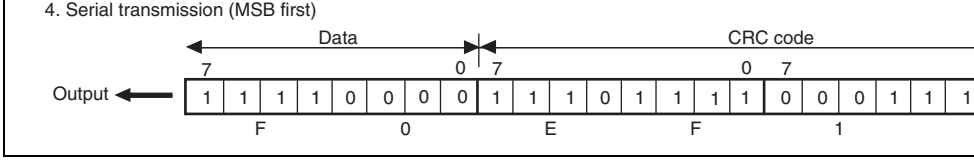


Figure 14.3 MSB-First Data Transmission

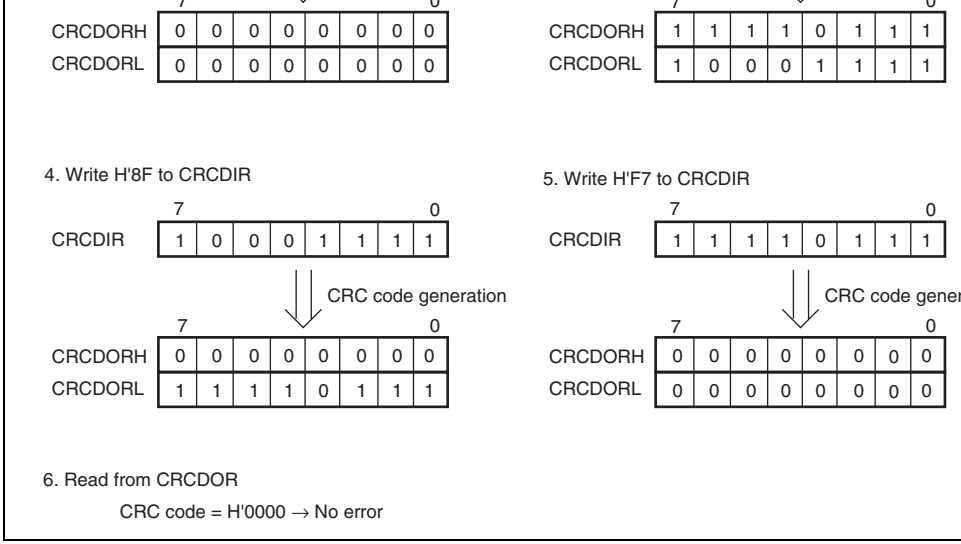


Figure 14.4 LSB-First Data Reception

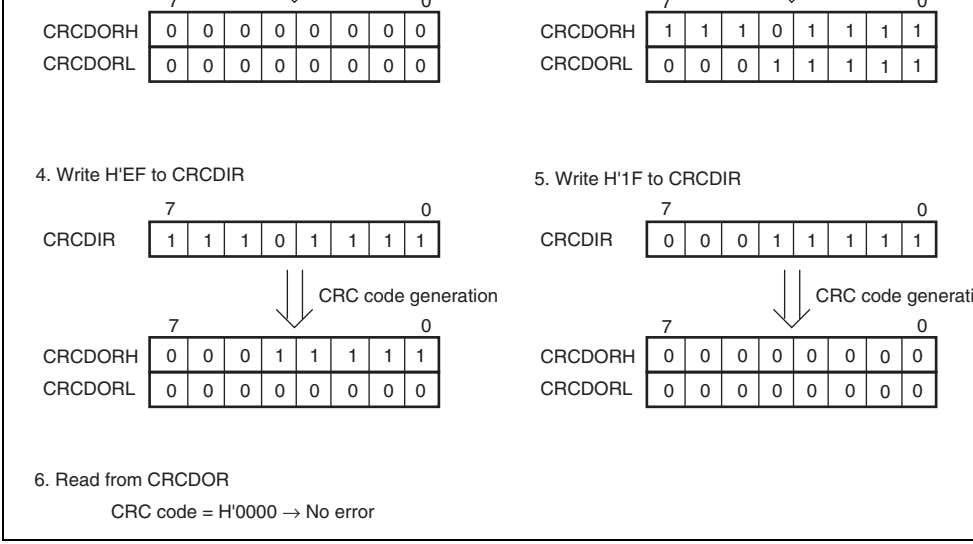


Figure 14.5 MSB-First Data Reception

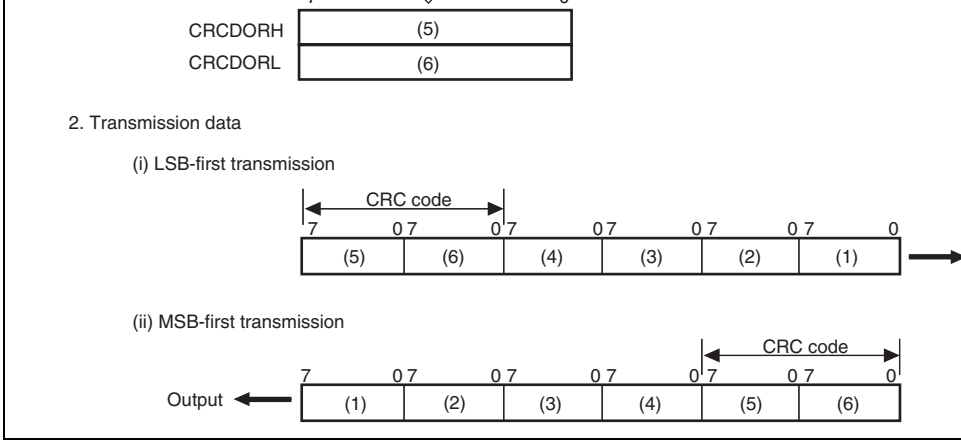


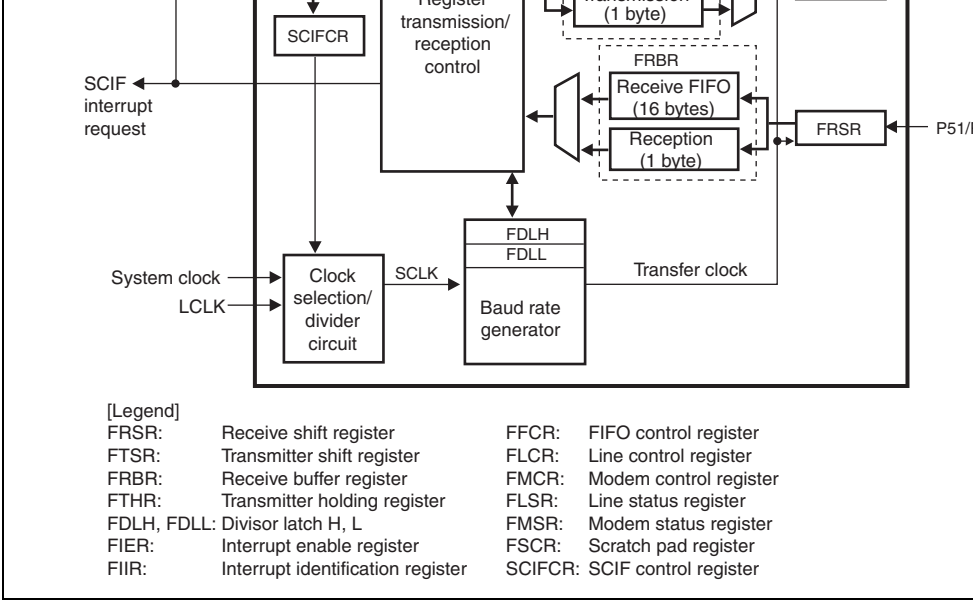
Figure 14.6 LSB-First and MSB-First Transmit Data

In addition, the SCIF can be connected to the LPC interface for direct control from the L

15.1 Features

- Full-duplex communication:

  The transmitter and receiver are independent, enabling transmission and reception to executed simultaneously. Both the transmitter and receiver use 16-stage FIFO buffer enabling continuous transmission and continuous reception of serial data.
- On-chip baud rate generator allows any bit rate to be selected
- Modem control function
- Data length: Selectable from 5, 6, 7, and 8 bits
- Parity: Selectable from even parity, odd parity, and no parity
- Stop bit length: Selectable from 1, 1.5, and 2 bits
- Receive error detection: Parity, overrun, and framing errors
- Break detection



[Legend]

FRSR:	Receive shift register	FFCR:	FIFO control register
FTSR:	Transmitter shift register	FLCR:	Line control register
FRBR:	Receive buffer register	FMCR:	Modem control register
FTHR:	Transmitter holding register	FLSR:	Line status register
FDLH, FDLL:	Divisor latch H, L	FMSR:	Modem status register
FIER:	Interrupt enable register	FSCR:	Scratch pad register
FIIR:	Interrupt identification register	SCIFCR:	SCIF control register

Figure 15.1 Block Diagram of SCIF

DCD	P24	Input	Data carrier detect input
$\overline{\text{DSR}}$	P26	Input	Data set ready input
$\overline{\text{DTR}}$	P27	Output	Data terminal ready output
$\overline{\text{CTS}}$	P64	Input	Transmission permission input
$\overline{\text{RTS}}$	P65	Output	Transmission request output

- Transmitter holding register (FTHR)
- Divisor latch L (FDLL)
- Interrupt enable register (FIER)
- Divisor latch H (FDLH)
- Interrupt identification register (FIIR)
- FIFO control register (FFCR)
- Line control register (FLCR)
- Modem control register (FMCR)
- Line status register (FLSR)
- Modem status register (FMSR)
- Scratch pad register (FSCR)
- SCIF control register (SCIFCR)
- SCIF address register H (SCIFADRH)
- SCIF address register L (SCIFADRL)
- SERIRQ control register 4 (SIRQCR4)

Table 15.2 Register Access

SCIFE Bit in HICR5	**0**		**1**	
Bit 3 in SUBMSTPBL	**0**	**1**	**0**	**1**
SCIFCR	H8S CPU access*[2]	Access disabled	H8S CPU access*[2]	Access d
Other than SCIFCR	H8S CPU access*[2]	Access disabled	LPC access*[1]	LPC acc

Notes: 1. When LPC access is set, writing from the H8S CPU is disabled. The read valu
2. When H8S CPU access is set, writing from the LPC is disabled. The read valu

the DR bit in FLSR is set.

When the FIFO is disabled, the data in FRBR must be read before the next data is receiv
data is received before the remaining data is read, the data is overwritten, resulting in an
error.

When this register is read with the FIFO enabled, the first buffer of the receive FIFO is r
When the receive FIFO becomes full, the subsequent receive data is lost, resulting in an
error.

Bit	Bit Name	Initial Value	R/W	Description
7 to 0	Bit 7 to bit 0	All 0	R	Stores received serial data. The data is 16 bytes when the FIFO is ena

15.3.3 Transmitter Shift Register (FTSR)

FTSR is a register that converts parallel data from the TxDF pin to serial data and then t
the serial data. When one frame transmission of serial data is completed, the next data is
transferred from FTHR. The serial data is transmitted from the LSB (bit 0).

FTSR cannot be written from the H8S CPU/LPC interface.

Bit	Bit Name	Initial Value	R/W	Description
7 to 0	Bit 7 to bit 0	—	W	Stores serial data to be transmitted. The data is 16 bytes when the FIFO is ena

15.3.5 Divisor Latch H, L (FDLH, FDLL)

The FDLH and FDLL are registers used to set the baud rate. They are accessible when th bit in FLCR is 1. Frequency division ranging from 1 to $(2^{16} - 1)$ can be set with these regi The frequency divider circuit stops when both of FDLH and FDLL are 0 (initial value).

- FDLH

Bit	Bit Name	Initial Value	R/W	Description
7 to 0	Bit 7 to bit 0	All 0	R/W	Upper 8 bits of divisor latch

- FDLL

Bit	Bit Name	Initial Value	R/W	Description
7 to 0	Bit 7 to bit 0	All 0	R/W	Lower 8 bits of divisor latch

Baud rate = (Clock frequency input to baud rate generator) / (16 × divisor value)

				0: Modem status interrupt disabled 1: Modem status interrupt enabled
2	ELSI	0	R/W	Receive Line Status Interrupt Enable 0: Receive line status interrupt disabled 1: Receive line status interrupt enabled
1	ETBEI	0	R/W	FTHR Empty Interrupt Enable 0: FTHR empty interrupt disabled 1: FTHR empty interrupt enabled
0	ERBFI	0	R/W	Receive Data Ready Interrupt Enable A character timeout interrupt is included w FIFO is enabled. 0: Receive data ready interrupt disabled 1: Receive data ready interrupt enabled

5, 4	—	All 0	R	Reserved These bits are always read as 0. The initial should not be changed.
3	INTID2	0	R	Interrupt ID2, ID1, ID0
2	INTID1	0	R	These bits Indicate the interrupt of the high
1	INTID0	0	R	priority among the pending interrupts. 000: Modem status 001: FTHR empty 010: Receive data ready 011: Receive line status 110: Character timeout (when the FIFO is e
0	INTPEND	1	R	Interrupt Pending Indicates whether one or more interrupts a pending. 0: Interrupt pending 1: No interrupt pending

						remaining, FIFO trigger level	receive below level.
1	1	0	0	2	Character timeout (with FIFO enabled)	No data is input to or output from the receive FIFO for the 4-character time period while one or more characters remain in the receive FIFO.	FRBR
0	0	1	0	3	FTHR empty	FTHR empty	FIIR re FTHR
0	0	0	0	4 (low)	Modem status	CTS, DSR, RI, DCD	FMSR

				10: 8 bytes 11: 14 bytes
5, 4	—	—	—	Reserved These bits cannot be modified.
3	DMAMODE	0	—	DMA Mode This bit is not supported. The initial value shou be changed.
2	XMITFRST	0	W	Transmit FIFO Reset The transmit FIFO data is cleared when 1 is w However, FTSR data is not cleared. This bit is automatically cleared.
1	RCVRFRST	0	W	Receive FIFO Reset The receive FIFO data is cleared when 1 is wr However, FRSR data is not cleared. This bit is automatically cleared.
0	FIFOE	0	W	FIFO Enable 0: Transmit/receive FIFOs disabled All bytes of these FIFOs are cleared. 1: Transmit/receive FIFOs enabled

				1: FDLL and FDLH access enabled
6	BREAK	0	R/W	Break Control Generates a break by driving the serial outpu TxDF low. The break state is released by clearing this b 0: Break released 1: Break generated
5	STICK PARITY	0	R	Stick Parity This bit is not supported in this LSI. This bit is always read as 0. The initial value not be changed.
4	EPS	0	R/W	Parity Select Selects even or odd parity when the PEN bit 0: Odd parity 1: Even parity
3	PEN	0	R/W	Parity Enable Selects whether to add a parity bit for data transmission and whether to perform a parity data reception. 0: No parity bit added/parity check disabled 1: Parity bit added/parity check enabled

				length. 00: Data length is 5 bits 01: Data length is 6 bits 10: Data length is 7 bits 11: Data length is 8 bits

15.3.10 Modem Control Register (FMCR)

FMCR controls output signals.

Bit	Bit Name	Initial Value	R/W	Description
7 to 5	—	All 0	R	Reserved These bits are always read as 0. The initial va should not be changed.
4	LOOP BACK	0	R/W	Loopback Test The transmit data output is internally connect receive data input, and the transmit data outp (RxDF) becomes 1. The receive data input pi disconnected from external sources. The four control input pins ($\overline{\text{DSR}}$, $\overline{\text{CTS}}$, $\overline{\text{RI}}$, and $\overline{\text{DCD}}$) a disconnected from external sources, and the internally connected to the four modem contro signals ($\overline{\text{DTR}}$, $\overline{\text{RTS}}$, $\overline{\text{OUT1}}$, and $\overline{\text{OUT2}}$), respe The transmit data is received immediately in l mode. Enabling/disabling of interrupts is set b OUT2LOOP bit in SCIFCR and FIER. 0: Loopback function disabled 1: Loopback function enabled

				• Normal operation No effect on operation • Loopback test Internally connected to the $\overline{RI}$ input pin.
1	RTS	0	R/W	Request to Send Controls the $\overline{RTS}$ output. 0: $\overline{RTS}$ output is high level 1: $\overline{RTS}$ output is low level
0	DTR	0	R/W	Data Terminal Ready Controls the $\overline{DTR}$ output. 0: $\overline{DTR}$ output is high level 1: $\overline{DTR}$ output is low level

				[Clearing condition] When FRBR is read or FLSR is read while tl no remaining data that could cause an error FIFO clear. 1: A receive FIFO error [Setting condition] When at least one data error (parity error, fra error, or break interrupt) has occurred in the
6	TEMT	1	R	Transmitter Empty Indicates whether transmit data remains. • When the FIFO is disabled 0: Transmit data remains in FTHR or FTSR. [Clearing condition] Transmit data is written to FTHR. 1: No transmit data remains in FTHR and FT [Setting condition] When no transmit data remains in FTHR and • When the FIFO is enabled 0: Transmit data remains in the transmit FIF FTSR. [Clearing condition] Transmit data is written to FTHR. 1: No transmit data remains in the transmit F FTSR. [Setting condition] When no transmit data remains in the transm and FTSR

				[Setting condition] When the transmit FIFO becomes empty • When the FIFO is disabled 0: Transmit data remains in FTHR. [Clearing condition] Transmit data is written to FTHR 1: No transmit data in FTHR [Setting condition] When data transfer from FTHR to FTSR is completed
4	BI	0	R	Break Interrupt Indicates detection of the receive data brea When the FIFO is enabled, a break interrup in any receive data in the FIFO, and this bi when the receive data is in the first FIFO b Reception of the next data starts after the i receive data becomes mark and a valid sta received. 0: Break signal not detected [Clearing condition] FLSR read 1: Break signal detected [Setting condition] When input receive data stays at space (lo for a reception time exceeding the length o frame

				[Clearing condition] FLSR read 1: A framing error [Setting condition] Invalid stop bit in the receive data
2	PE	0	R	Parity Error This bit indicates a parity error in the receive when the PEN bit in FLCR is 1. When the FI enabled, this error occurs in any receive dat FIFO, and this bit is set when the receive da the first FIFO buffer. 0: No parity error [Clearing condition] FLSR read If this bit is set during an overrun error, read twice. 1: A parity error [Setting condition] Detection of parity error in receive data

				lost. 0: No overrun error [Clearing condition] FLSR read 1: An overrun error [Setting condition] Occurrence of an overrun error
0	DR	0	R	Data Ready Indicates that receive data is stored in FRB FIFO. 0: No receive data [Clearing condition] FRBR is read or all of the FIFO data is rea 1: Receive data remains. [Setting condition] Reception of data

				Indicates the inverted state of the $\overline{\text{DSR}}$ input
4	CTS	0	R	Clear to Send
				Indicates the inverted state of the $\overline{\text{CTS}}$ input
3	DDCD	0	R	Delta Data Carrier Indicator
				Indicates a change in the $\overline{\text{DCD}}$ input signal a DDCD bit is read.
				0: No change in the $\overline{\text{DCD}}$ input signal after F read
				[Clearing condition]
				FMSR read
				1: A change in the $\overline{\text{DCD}}$ input signal after FM
				[Setting condition]
				A change in the $\overline{\text{DCD}}$ input signal
2	TERI	0	R	Trailing Edge Ring Indicator
				Indicates a rise in the $\overline{\text{RI}}$ input signal after th bit is read.
				0: No change in the $\overline{\text{RI}}$ input signal after FM
				[Clearing condition]
				FMSR read
				1: A rise in the $\overline{\text{RI}}$ input signal after FMSR re
				[Setting condition]
				A rise in the $\overline{\text{RI}}$ input pin

				[Setting condition]
				A change in the $\overline{DSR}$ input signal
0	DCTS	0	R	Delta Clear to Send Indicator
				Indicates a change in the $\overline{CTS}$ input signal DCTS bit is read.
				0: No change in the $\overline{CTS}$ input signal after read
				[Clearing condition]
				FMSR read
				1: A change in the $\overline{CTS}$ input signal after F
				[Setting condition]
				A change in the $\overline{CTS}$ input signal

15.3.13 Scratch Pad Register (FSCR)

FSCR is not used for SCIF control, but is used to temporarily store program data.

Bit	Bit Name	Initial Value	R/W	Description
7 to 0	Bit 7 to bit 0	All 0	R/W	Temporarily stores program data.

				Do not change the initial value.
4	OUT2LOOP	0	R/W	Enables or disables interrupts during a loopb test. 0: Interrupt enabled 1: Interrupt disabled
3	CKSEL1	0	R/W	These bits select the clock (SCLK) to be inp baud rate generator. 00: LCLK divided by 18 01: System clock divided by 11 10: Reserved for LCLK (not selectable) 11: Reserved for system clock (not selectab
2	CKSEL0	0	R/W	
1	SCIFRST	0	R/W	Resets the baud rate generator, FRSR, and 0: Normal operation 1: Reset
0	REGRST	0	R/W	Resets registers (except SCIFCR) accessibl the H8S CPU or LPC interface. 0: Normal operation 1: Reset

and P-05 pins are set to PORT.

CKSEL1, CKSEL0	LCLK (33 MHz) divided by 18		System Clock (34 MHz) divided by 1	
Baud rate	FDLH, FDLL (Hex)	Error (%)	FDLH, FDLL (Hex)	Error (%)
50	0900	-0.54%	H'0F18	-0.01%
75	0600	-0.54%	H'0A10	-0.01%
110	0417	-0.51%	H'06DC	0.01%
300	0180	-0.54%	H'0284	-0.01%
600	00C0	-0.54%	H'0142	-0.01%
1200	0060	-0.54%	H'00A1	-0.01%
1800	0040	-0.54%	H'006B	0.30%
2400	0030	-0.54%	H'0050	0.62%
4800	0018	-0.54%	H'0028	0.62%
9600	000C	-0.54%	H'0014	0.62%
14400	0008	-0.54%	H'000D	—
19200	0006	-0.54%	H'000A	0.62%
38400	0003	-0.54%	H'0005	0.62%
57600	0002	-0.54%	H'0003	—
115200	0001	-0.54%	H'0002	—

enabling continuous data transmission and reception.

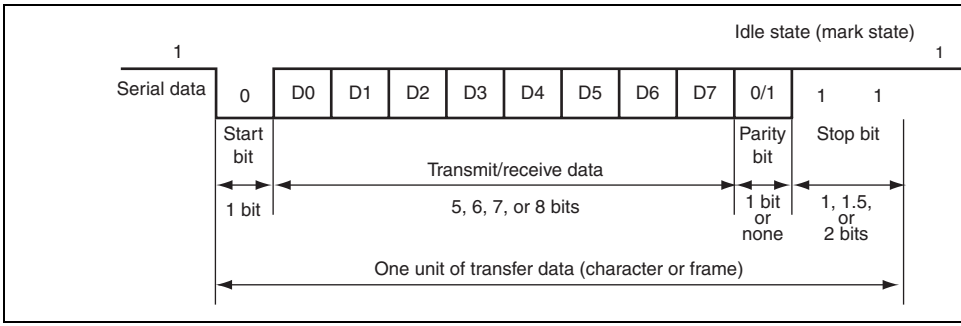


Figure 15.2 Data Format in Serial Transmission/Reception (Example with 8-Bit Data, Parity and 2 Stop Bits)

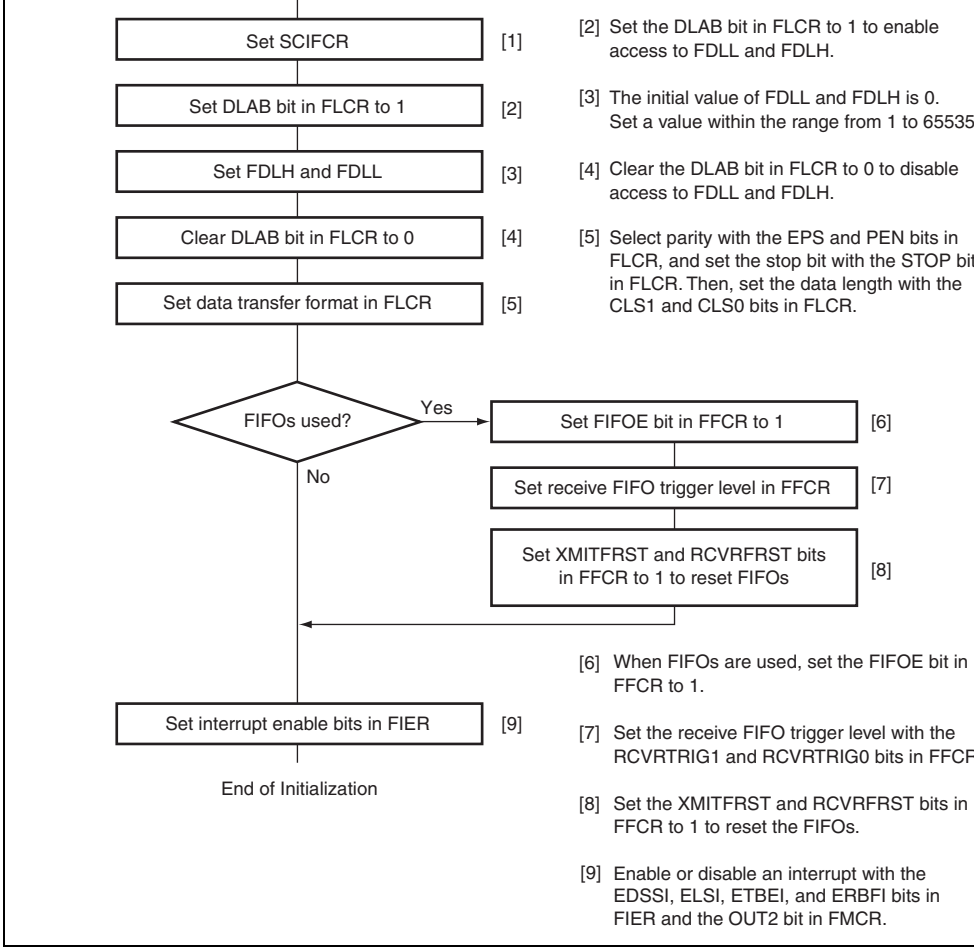
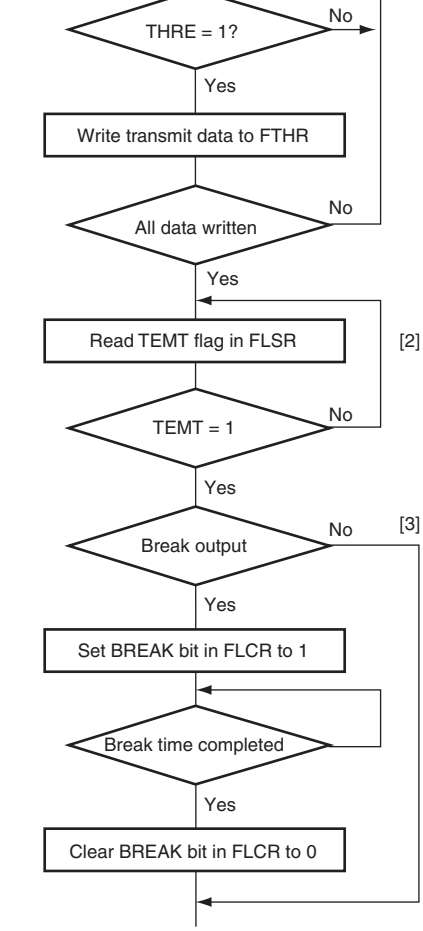
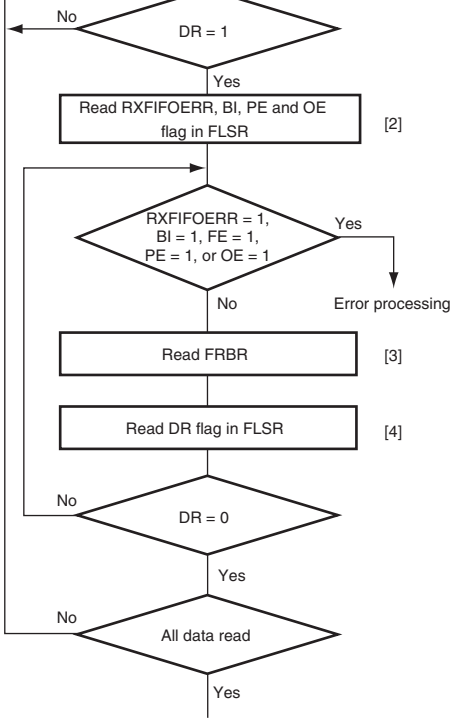


Figure 15.3 Example of Initialization Flowchart



[3] To output a break at the end of serial transmission
BREAK bit in FLCR to 1. After completion of the br
clear the BREAK bit in FLCR to 0 to clear the brea

Figure 15.4 Example of Data Transmission Flowchart

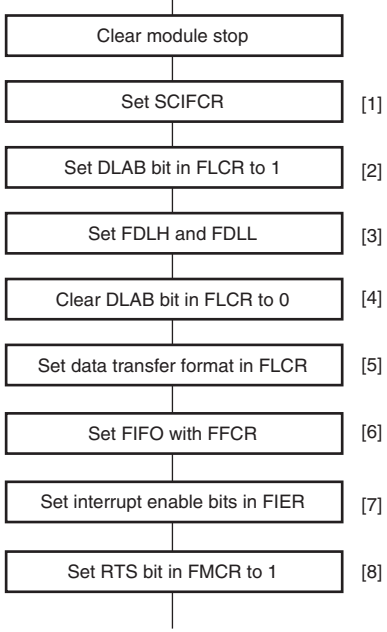


set to 1, a receive line status interrupt occurs.

[3] Read the receive data in FRBR.

[4] Check the DR flag in FLSR. When the DR flag is cleared to 0 and all data has been read, data recepti is complete.

Figure 15.5 Example of Data Reception Flowchart



[2] Set the DLAB bit in FLCR to 1 to enable access
FDLL and FDLH.

[3] The initial value of FDLL and FDLH is 0. Set a v
within the range from 1 to 65535.

[4] Clear the DLAB bit in FLCR to 0 to disable acce
FDLL and FDLH.

[5] Select parity with the EPS and PEN bits in FLCR
set the stop bit with the STOP bit in FLCR. Then
the data length with the CLS1 and CLS0 bits in
Set the FIFOE bit in FFCR to 1 to enable the FI

[6] Set the receive FIFO trigger level with the RCVR
and RCVRTRIG0 bits in FFCR. Select the best t
level to prevent an overflow of the receive FIFO.

[7] Set the EDSSI and ERBFI bits in FIER to 1 to er
modem status interrupt and receive data ready i

[8] Set the RTS bit in FMCR to 1.

Figure 15.6 Example of Initialization Flowchart

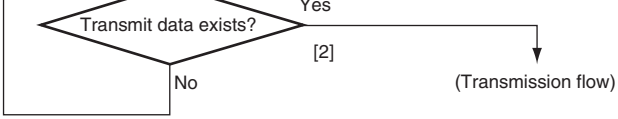


Figure 15.7 Example of Data Transmission/Reception Standby Flowchart

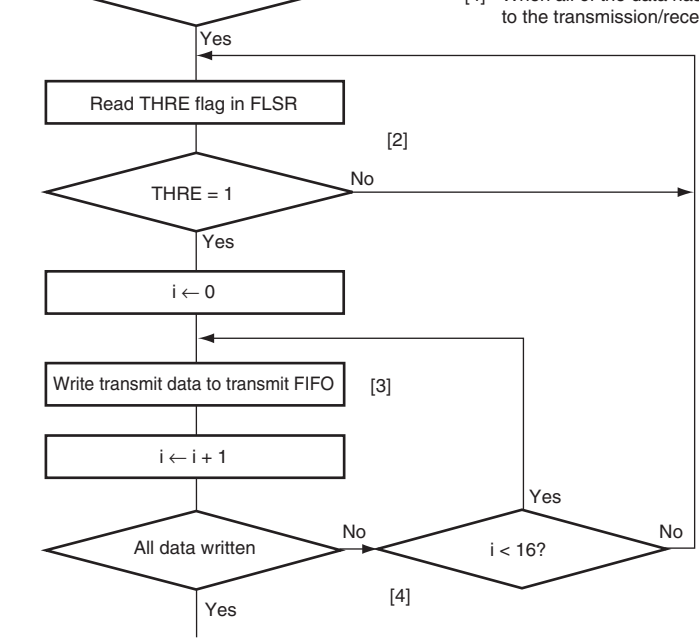


Figure 15.8 Example of Data Transmission Flowchart

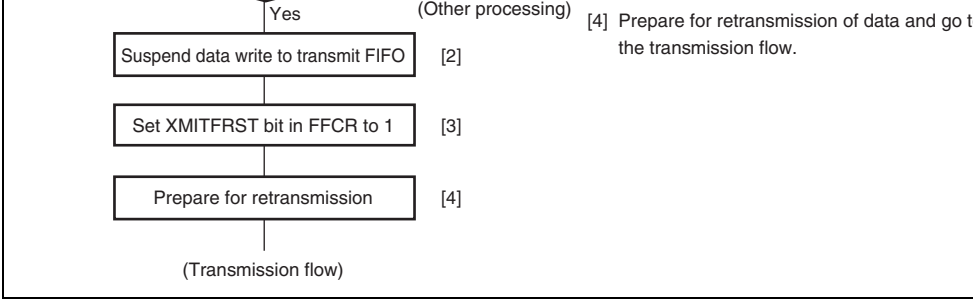
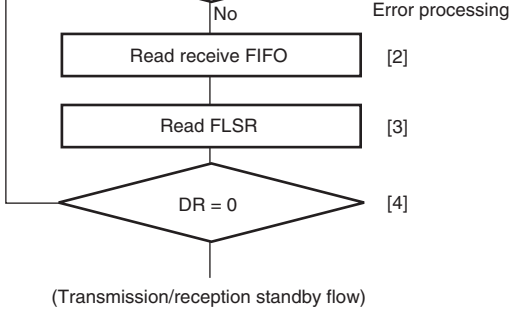
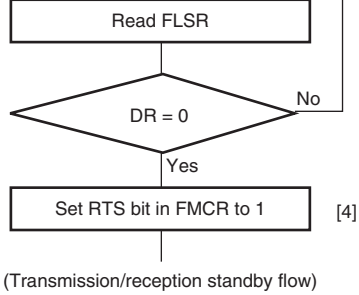


Figure 15.9 Example of Data Transmission Suspension Flowchart



[4] Check the DR flag in FLSR. When the D
is cleared and all of the data has been re
reception is complete.

Figure 15.10 Example of Data Reception Flowchart



[4] Set the RTS bit in FMCR to 1, and then go to the transmission/reception standby flow.

Figure 15.11 Example of Data Reception Suspension Flowchart

LPC Interface I/O Address						SC
Bits 15 to 3	**Bit 2**	**Bit 1**	**Bit 0**	**R/W**	**Condition**	**Re**
SCIFADR (bits 15 to 3)	0	0	0	R	FLCR[7] = 0	FR
				W	FLCR[7] = 0	FT
				R/W	FLCR[7] = 1	FD
SCIFADR (bits 15 to 3)	0	0	1	R/W	FLCR[7] = 0	FIE
				R/W	FLCR[7] = 1	FD
SCIFADR (bits 15 to 3)	0	1	0	R	—	FII
				W	—	FF
SCIFADR (bits 15 to 3)	0	1	1	R/W	—	FL
SCIFADR (bits 15 to 3)	1	0	0	R/W	—	FM
SCIFADR (bits 15 to 3)	1	0	1	R	—	FL
SCIFADR (bits 15 to 3)	1	1	0	R	—	FM
SCIFADR (bits 15 to 3)	1	1	1	R/W	—	FS

SCSIRQ3	Initialized	Retained	Retained	Retai
SCSIRQ2	Initialized	Retained	Retained	Retai
SCSIRQ1	Initialized	Retained	Retained	Retai
SCSIRQ0	Initialized	Retained	Retained	Retai

Receive line status	Overrun error, parity error, framing error, break interrupt
Receive data ready	Acceptance of receive data, FIFO trigger level
Character timeout (when FIFO is enabled)	No data is input to or output from the receive FIFO for the 4-character time period while one or more characters remain in the receive FIFO.
FTHR empty	FTHR empty
Modem status	CTS, DSR, RI, DCD

Table 15.9 shows the interrupt source, vector address, and interrupt priority.

Table 15.9 Interrupt Source, Vector Address, and Interrupt Priority

Interrupt Origin of Interrupt Source	Interrupt Name	Vector Number	Vector Address
SCIF	SCIF (SCIF interrupt)	82	H'000148

15.6 Usage Note

15.6.1 Power-Down Mode When LCLK is Selected for SCLK

To switch to software standby mode when LCLK divided by 18 has been selected for SC
the shutdown function of the LPC interface to stop LCLK.

for IPMI applications.

- Five serial pin multiplexed modes
  - Mode 0: Each COM port is used for its respective serial communication module: for SCIF, COM2 for SCI_1 and COM3 for SCI_3 (default mode)
  - Mode 1: COM1 snoop mode with use of SCI_1 and internal registers
  - Mode 2: SCIF-and-SCI_1 bridge mode in which internal registers provide softwa control.
  - Mode 3: COM port switched mode in which COM1 is connected to SCI_1 and C connected to SCIF. Internal registers provide flow control for SCI_1.
  - Mode 4: SCIF-and-SCI_3 bridge mode providing the same functionality as mode

Please refer to section 13, Serial Communication Interface (SCI) for details on SCI_1 ar and section 15, Serial Communication Interface with FIFO (SCIF), for details on SCIF.

DCD	Input	Data carrier detect	P24
$\overline{\text{DSR}}$	Input	Data set ready	P26
$\overline{\text{DTR}}$	Output	Data terminal ready	P27
$\overline{\text{CTS}}$	Input	Transmission permission	P64
$\overline{\text{RTS}}$	Output	Transmission request	P65

Bit	Bit Name	Initial Value	R/W	Description
7	DCD1	—	R	Monitors the state of the $\overline{\text{DCD}}$ line in modes 4.
6	RI1	—	R	Monitors the state of the $\overline{\text{RI}}$ line in modes 1,
5	DSR1	—	R	Monitors the state of the $\overline{\text{DSR}}$ line in modes 4.
4	SME	0	R/W	Serial Pin Multiplex Enable 0: Pin multiplexing disabled 1: Pin multiplexing enabled
3	—	0	R	Reserved This bit is always read as 0. The write value always be 0.
2	SM2	0	R/W	Serial Pin Multiplexed Mode Select
1	SM1	0	R/W	These bits select a serial pin multiplexed mo
0	SM0	0	R/W	selection is only enabled when SME bit is 1. 000: Serial multiplexed mode 0 001: Serial multiplexed mode 1 010: Serial multiplexed mode 2 011: Serial multiplexed mode 3 100: Serial multiplexed mode 4 101: Reserved (Do not modify) 110: Reserved (Do not modify) 111: Reserved (Do not modify)

				and 4. 0: 0 is output 1: 1 is output
5	RTS1	1	R/W	Controls the output on the $\overline{\text{RTS}}$ pin of COM1. Controls the input on the $\overline{\text{CTS}}$ pin of SCIF in mo 0: 0 is output 1: 1 is output
4	CTS3	—	R	Monitors the state of the $\overline{\text{RTS}}$ pin input of SCIF i 4.
3	—	—	R	Reserved
2	RTS3	1	R/W	Controls the output on the $\overline{\text{CTS}}$ pin of SCIF. 0: 0 is output 1: 1 is output
1,0	—	—	R/W	Reserved

RxD1 and TxD1 of SCI_1 are cross-connected to COM2. RxD3 and TxD3 of SCI_3 are connected to COM3.

Figure 16.1 illustrates the pin connection in serial pin multiplexed mode 0.

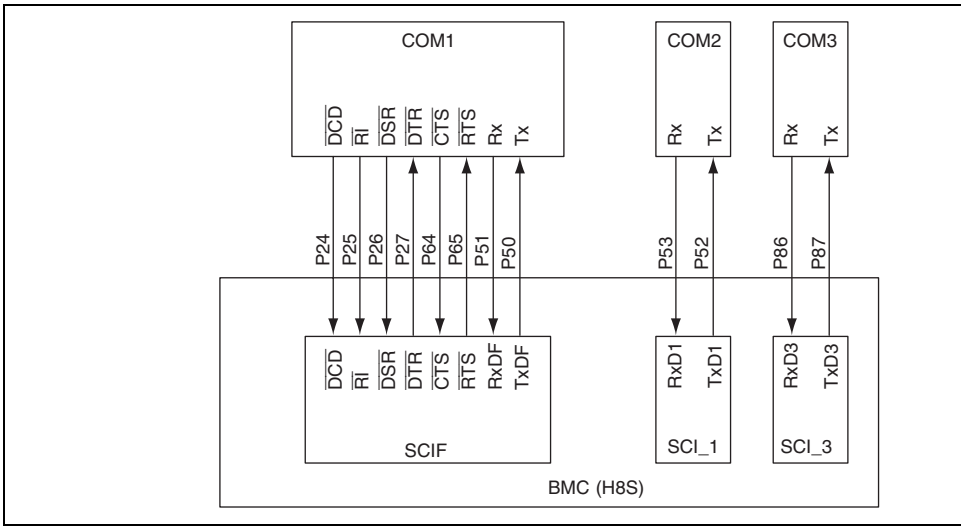


Figure 16.1 Serial Pin Multiplexed Mode 0

SMR0 register. The pin state of $\overline{CTS}$ of COM1 is reflected in bit CTS1 of the SMR1 regi

Figure 16.2 illustrates the pin connection in serial pin multiplexed mode 1.

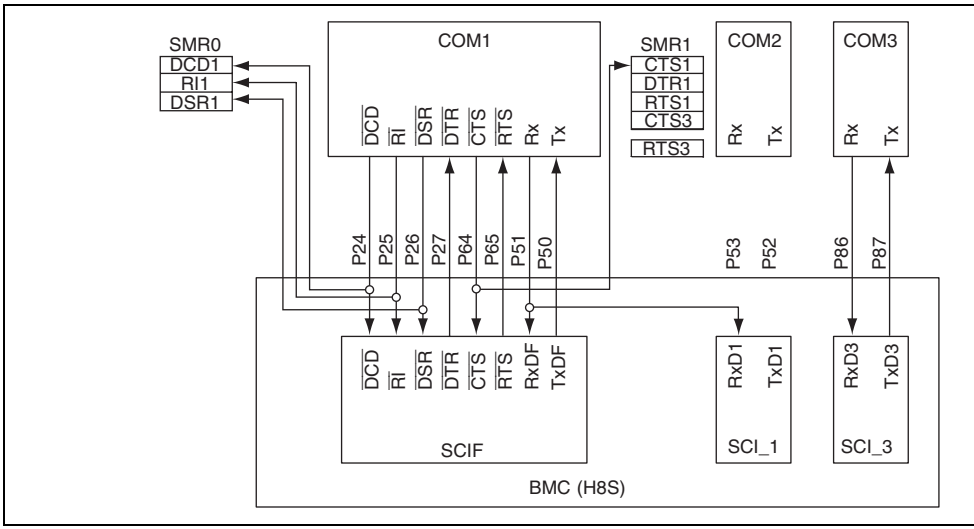


Figure 16.2 Serial Pin Multiplexed Mode 1

The value written to bit RTS1 of the SMR1 register is reflected in the $\overline{CTS}$ input of SCI
state of $\overline{RTS}$ of SCIF is reflected in bit CTS1 of the SMR1 register.

Figure 16.3 illustrates the pin connection in serial pin multiplexed mode 2.

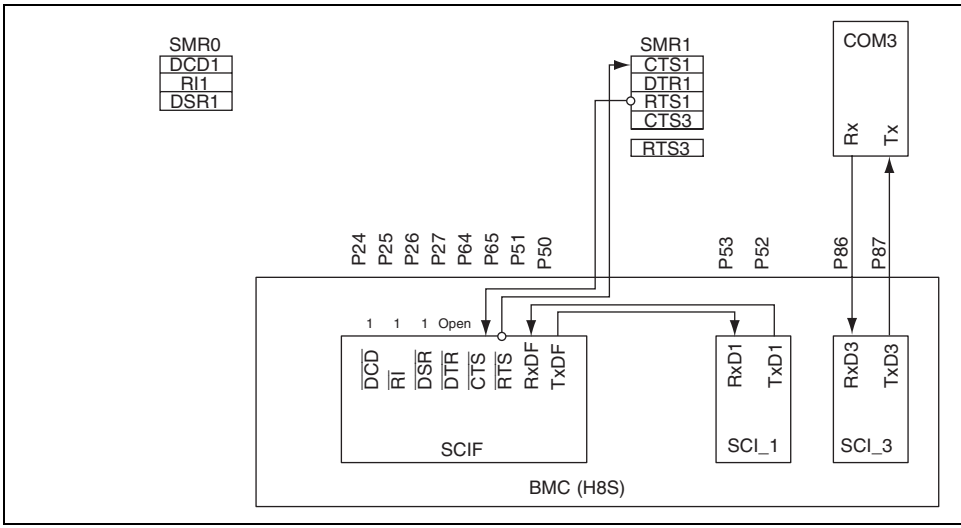


Figure 16.3 Serial Pin Multiplexed Mode 2

The states of $\overline{DCD}$/$\overline{RI}$/$\overline{DSR}$ of COM1 are reflected in bits DCD1/RI1/DSR1 of the SMR0 and $\overline{CTS}$ of COM1 is reflected in bit CTS1 of the SMR1 register.

The values written to bits DTR1/RTS1 of the SMR1 register are output to $\overline{DTR}$/$\overline{RTS}$ of C

Figure 16.4 illustrates the pin connection in serial pin multiplexed mode 3.

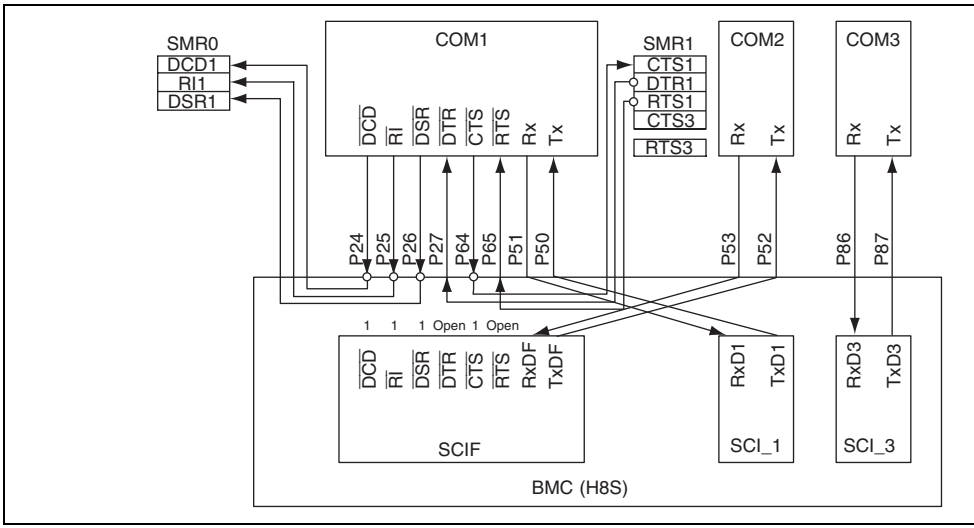


Figure 16.4 Serial Pin Multiplexed Mode 3

The states of $\overline{DCD}$/$\overline{RI}$/$\overline{DSR}$ of COM1 are reflected in bits DCD1/RI1/DSR1 of the SMR and CTS of COM1 is reflected to CTS1 bit of SMR1 register.

The values written to bits DTR1/RTS1 of the SMR1 register are output to $\overline{DTR}$/$\overline{RTS}$ of The value written to bit RTS3 of SMR1 is reflected in $\overline{CTS}$ of SCIF, and the state of $\overline{RT}$ is reflected in bit CTS3 of SMR1, allowing SCI_3 and SCIF to communicate each other virtual flow control.

Figure 16.5 illustrates the pin connection in serial pin multiplexed mode 4.

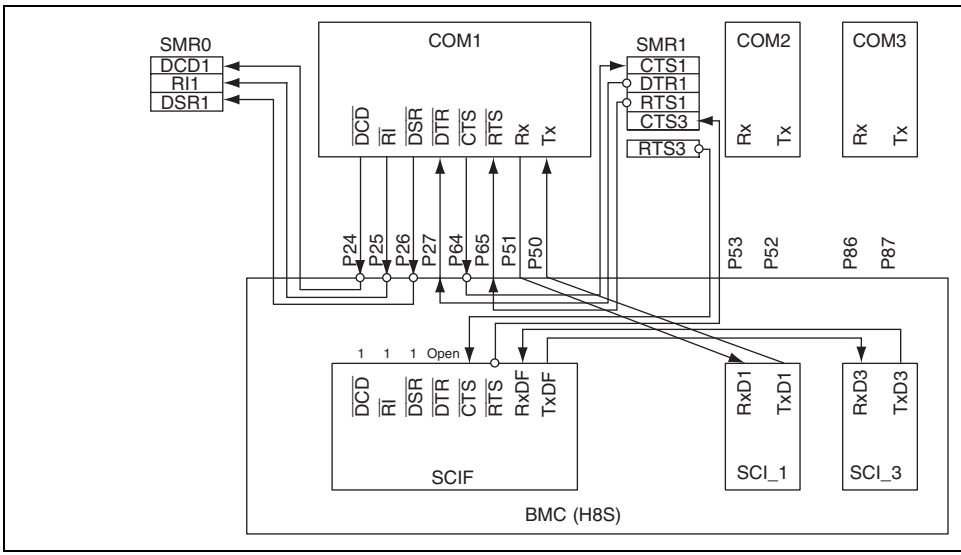


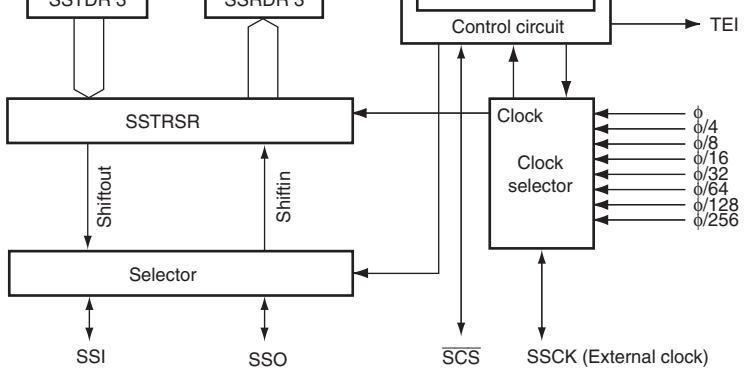
Figure 16.5 Serial Pin Multiplexed Mode 4

- Choice of SSU mode and clock synchronous mode
- Choice of master mode and slave mode
- Choice of standard mode and bidirectional mode
- Synchronous serial communication with devices with different clock polarity and clo
- Choice of 8/16/32-bit width of transmit/receive data
- Full-duplex communication capability

  The shift register is incorporated, enabling transmission and reception to be executed simultaneously.
- Consecutive serial communication
- Choice of LSB-first or MSB-first transfer
- Choice of clock sources

  $\phi/4$, $\phi/8$, $\phi/16$, $\phi/32$, $\phi/64$, $\phi/128$, $\phi/256$, or an external clock
- Five interrupt sources

  Transmit-end, transmit-data-register-empty, receive-data-full, overrun-error, and con
- Module stop mode can be set.



[Legend]

SSCRH:	SS control register H
SSCRL:	SS control register L
SSCR2:	SS control register 2
SSMR:	SS mode register
SSER:	SS enable register
SSSR:	SS status register
SSTDR0 to SSTDR3:	SS transmit data registers 0 to 3
SSRDR0 to SSRDR3:	SS receive data registers 0 to 3
SSTRSR:	SS shift register

Figure 17.1 Block Diagram of SSU

SCS	I/O	SSU chip select input/output

17.3 Register Descriptions

The SSU has the following registers.

- SS control register H (SSCRH)
- SS control register L (SSCRL)
- SS mode register (SSMR)
- SS enable register (SSER)
- SS status register (SSSR)
- SS control register 2 (SSCR2)
- SS transmit data register 0 (SSTDR0)
- SS transmit data register 1 (SSTDR1)
- SS transmit data register 2 (SSTDR2)
- SS transmit data register 3 (SSTDR3)
- SS receive data register 0 (SSRDR0)
- SS receive data register 1 (SSRDR1)
- SS receive data register 2 (SSRDR2)
- SS receive data register 3 (SSRDR3)
- SS shift register (SSTRSR)

				in SSSR is set, this bit is automatically cleared 0: Slave mode is selected. 1: Master mode is selected.
6	BIDE	0	R/W	Bidirectional Mode Enable Selects that both serial data input pin and outp used or one of them is used. However, transmi and reception are not performed simultaneous bidirectional mode is selected. For details, sec 17.4.3, Relationship between Data Input/Outpu and Shift Register. 0: Standard mode (two pins are used for data i output) 1: Bidirectional mode (one pin is used for data output)
5	—	0	R/W	Reserved This bit is always read as 0. The initial value sh be changed.

				1: Serial data output is changed to high.
3	SOLP	1	R/W	SOL Bit Write Protect When changing the output level of serial data SOL bit to 1 or clear the SOL bit to 0 after cle SOLP bit to 0 using the MOV instruction. 0: Output level can be changed by the SOL b 1: Output level cannot be changed by the SO bit is always read as 1.
2	SCKS	0	R/W	SSCK Pin Select Selects that the SSCK pin functions as a port clock pin. When the SSCK pin is used as a se pin, this bit must be set to 1. 0: Functions as an I/O port. 1: Functions as a serial clock.
1	CSS1	0	R/W	$\overline{\text{SCS}}$ Pin Select
0	CSS0	0	R/W	Select that the $\overline{\text{SCS}}$ pin functions as a port or or output. However, when MSS = 0, the $\overline{\text{SCS}}$ functions as an input pin regardless of the CS CSS0 settings. 00: I/O port 01: Function as $\overline{\text{SCS}}$ input 10: Function as $\overline{\text{SCS}}$ automatic input/output (f $\overline{\text{SCS}}$ input before and after transfer and o low level during transfer) 11: Function as $\overline{\text{SCS}}$ automatic output (outpu level before and after transfer and outputs level during transfer)

				0: SSU mode
				1: Clock synchronous mode
5	SRES	0	R/W	Software Reset
				Setting this bit to 1 forcibly resets the SSU inte sequencer. After that, this bit is automatically c The ORER, TEND, TDRE, RDRF, and CE bits and the TE and RE bits in SSER are also initia Values of other bits for SSU registers are held.
				To stop transfer, set this bit to 1 to reset the SS internal sequencer.
4 to 2	—	All 0	R/W	Reserved
				These bits are always read as 0. The initial val should not be changed.
1	DATS1	0	R/W	Transmit/Receive Data Length Select
0	DATS0	0	R/W	Select serial data length.
				00: 8 bits
				01: 16 bits
				10: 32 bits
				11: Setting prohibited

				1: MSB first
6	CPOS	0	R/W	Clock Polarity Select Selects the SSCK clock polarity. 0: High output in idle mode, and low output in mode 1: Low output in idle mode, and high output in mode
5	CPHS	0	R/W	Clock Phase Select (Only for SSU Mode) Selects the SSCK clock phase. 0: Data changes at the first edge. 1: Data is latched at the first edge.
4, 3	—	All 0	R/W	Reserved These bits are always read as 0. The initial va should not be changed.
2	CKS2	0	R/W	Transfer Clock Rate Select
1	CKS1	0	R/W	Select the transfer clock rate (prescaler divisi
0	CKS0	0	R/W	when an internal clock is selected. 000: Reserved 100: ϕ/32 001: ϕ/4 101: ϕ/64 010: ϕ/8 110: ϕ/128 011: ϕ/16 111: ϕ/256

				When this bit is set to 1, reception is enabled.
5, 4	—	All 0	R/W	Reserved These bits are always read as 0. The initial val should not be changed.
3	TEIE	0	R/W	Transmit End Interrupt Enable When this bit is set to 1, a TEI interrupt reques enabled.
2	TIE	0	R/W	Transmit Interrupt Enable When this bit is set to 1, a TXI interrupt reques enabled.
1	RIE	0	R/W	Receive Interrupt Enable When this bit is set to 1, an RXI interrupt reque an OEI interrupt request are enabled.
0	CEIE	0	R/W	Conflict Error Interrupt Enable When this bit is set to 1, a CEI interrupt reques enabled.

				error occurs, indicating abnormal termination. stores 1-frame receive data before an overrun occurs and loses data to be received later. W = 1, consecutive serial reception cannot be co Serial transmission cannot be continued, eith [Setting condition] When one byte of the next reception is compl RDRF = 1 [Clearing condition] When writing 0 after reading ORER = 1
5, 4	—	All 0	R/W	Reserved These bits are always read as 0. The initial va should not be changed.
3	TEND	1	R	Transmit End [Setting condition] • When the last bit of transmit data is transr while the TENDSTS bit in SSCR2 is clear and the TDRE bit is set to 1 • After the last bit of transmit data is transm the TENDSTS bit in SSCR2 is set to 1 an TDRE bit is set to 1 [Clearing conditions] • When writing 0 after reading TEND = 1 • When writing data to SSTDR

1	RDRF	0	R/W	Receive Data Register Full Indicates whether or not SSRDR contains rece [Setting condition] • When receive data is transferred from SST SSRDR after successful serial data recepti [Clearing conditions] • When writing 0 after reading RDRF = 1 • When reading receive data from SSRDR
0	CE	0	R/W	Conflict/Incomplete Error Indicates that a conflict error has occurred when 0 is externally input to the $\overline{SCS}$ pin with S = 0 (SSU mode) and MSS = 1 (master mode). If the $\overline{SCS}$ pin level changes to 1 with SSUMS mode) and MSS = 0 (slave mode), an incomple occurs because it is determined that a master has terminated the transfer. Data reception do continue while the CE bit is set to 1. Serial tran also does not continue. Reset the SSU interna sequencer by setting the SRES bit in SSCRL t before resuming transfer after incomplete erro [Setting condition] • When a low level is input to the $\overline{SCS}$ pin in mode (the MSS bit in SSCRH is set to 1) • When the $\overline{SCS}$ pin is changed to 1 during t slave mode (the MSS bit in SSCRH is clea [Clearing condition] • When writing 0 after reading CE = 1

				serial data differ according to the register sett details, 14.4.3, Relationship between Data Inp Pins and Shift Register. 0: CMOS output 1: NMOS open drain output
6	SSCKOS	0	R/W	SSCK Pin Open Drain Select Selects whether the SSCK pin is used as a C an NMOS open drain output. 0: CMOS output 1: NMOS open drain output
5	SCSOS	0	R/W	$\overline{\text{SCS}}$ Pin Open Drain Select Selects whether the $\overline{\text{SCS}}$ pin is used as a CM NMOS open drain output. 0: CMOS output 1: NMOS open drain output
4	TENDSTS	0	R/W	Selects the timing of setting the TEND bit (val and master mode). 0: Sets the TEND bit when the last bit is being transmitted 1: Sets the TEND bit after the last bit is transr

				= 1, TE = 1, and RE = 0, the SSO pin output while the SCS pin is driven low
1, 0	—	All 0	R/W	Reserved These bits are always read as 0. The initial val should not be changed.

17.3.7 SS Transmit Data Registers 0 to 3 (SSTDR0 to SSTDR3)

SSTDR is an 8-bit register that stores transmit data. When 8-bit data length is selected by DATS1 and DATS0 in SSCRL, SSTDR0 is valid. When 16-bit data length is selected, SS and SSTDR1 are valid. When 32-bit data length is selected, SSTDR0 to SSTDR3 are vali

When the SSU detects that SSTRSR is empty, it transfers the transmit data written in SST SSTRSR and starts serial transmission. If the next transmit data has already been written SSTDR during serial transmission, the SSU performs consecutive serial transmission.

Although SSTDR can always be read from or written to by the CPU and DMAC, to achie reliable serial transmission, write transmit data to SSTDR after confirming that the TDRE SSSR is set to 1.

SSRDR is a read-only register, therefore, cannot be written to by the CPU.

17.3.9 SS Shift Register (SSTRSR)

SSTRSR is a shift register that transmits and receives serial data.

When data is transferred from SSTDR to SSTRSR, bit 0 of transmit data is bit 0 in the S contents (MLS = 0: LSB first communication) and is bit 7 in the SSTDR contents (MLS first communication). The SSU transfers data from the LSB (bit 0) in SSTRSR to the SS perform serial data transmission.

In reception, the SSU sets serial data that has been input via the SSI pin in SSTRSR fron (bit 0). When 1-byte data has been received, the SSTRSR contents are automatically tra SSRDR. SSTRSR cannot be directly accessed by the CPU.

17.4.2 Relationship of Clock Phase, Polarity, and Data

The relationship of clock phase, polarity, and transfer data depends on the combination o
CPOS and CPHS bits in SSMR. Figure 17.2 shows the relationship. When SSUMS = 1, t
setting is invalid although the CPOS setting is valid.

Setting the MLS bit in SSMR selects that MSB or LSB first communication. When MLS
is transferred from the LSB to the MSB. When MLS = 1, data is transferred from the MS
LSB.

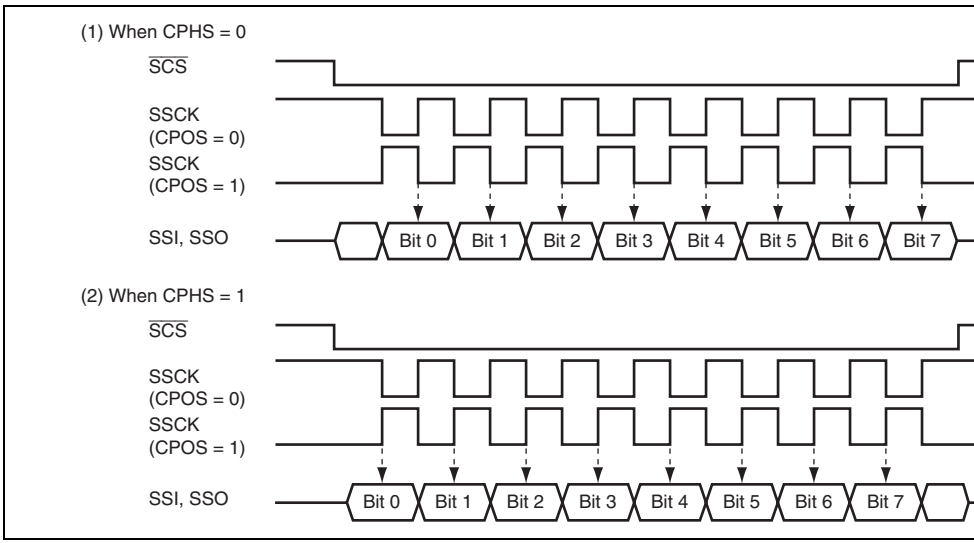


Figure 17.2 Relationship of Clock Phase, Polarity, and Data

The SSU transmits and receives serial data from the SSO pin regardless of master or sla
when operating with BIDE = 1 (bidirectional mode) (see figures 17.3 (3) and (4)).

However, even if both the TE and RE bits are set to 1, transmission and reception are no
performed simultaneously. Either the TE or RE bit must be selected.

The SSU transmits serial data from the SSO pin and receives serial data from the SSI pi
operating with SSUMS = 1. The SSCK pin outputs the internal clock when MSS = 1 an
as an input pin when MSS = 0 (see figures 17.3 (5) and (6)).

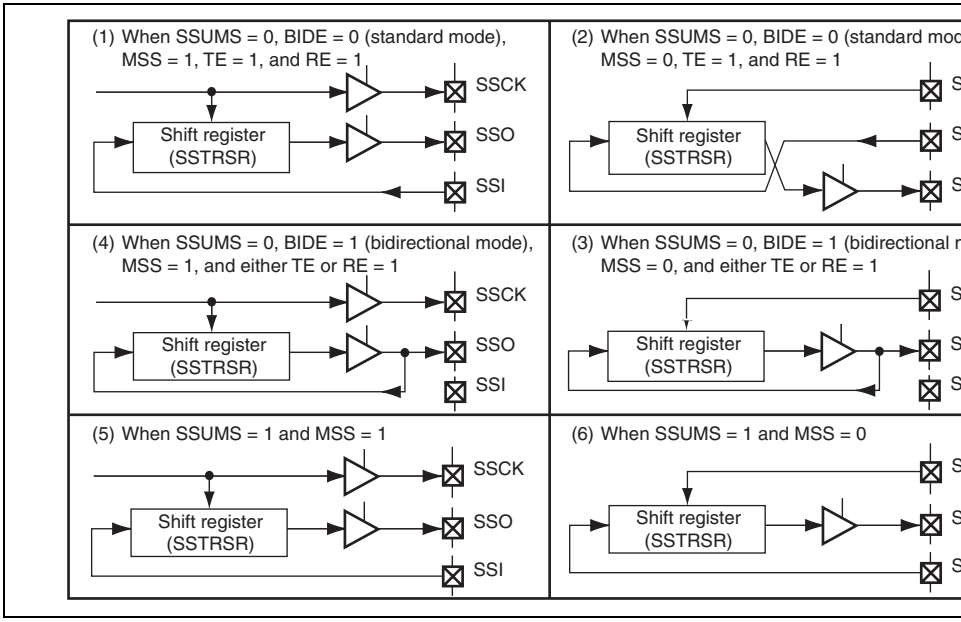


Figure 17.3 Relationship between Data Input/Output Pins and the Shift Reg

SSU communication mode	0	0	0	0	1	—	In
				1	0	Output	–
					1	Output	In
			1	0	1	Input	–
				1	0	—	Cl
					1	Input	Cl
SSU (bidirectional) communication mode	0	1	0	0	1	—	In
				1	0	—	Cl
			1	0	1	—	In
				1	0	—	Cl
Clock synchronous communication mode	1	0	0	0	1	Input	–
				1	0	—	Cl
					1	Input	Cl
			1	0	1	Input	–
				1	0	—	Cl
					1	Input	Cl

[Legend]

—: Not used as SSU pin (can be used as I/O port)

communication mode		1	Input
	1	0	—
		1	Output

[Legend]

—: Not used as SSU pin (can be used as I/O port)

Table 17.4 Communication Modes and Pin States of $\overline{\text{SCS}}$ Pin

Communication Mode	Register Setting				Pi
	SSUMS	MSS	CSS1	CSS0	$\overline{\text{SC}}$
SSU communication mode	0	0	×	×	Inp
		1	0	0	—
			0	1	Inp
			1	0	Au inp
			1	1	Ou
Clock synchronous communication mode	1	×	×	×	—

[Legend]

×: Don't care

—: Not used as SSU pin (can be used as I/O port)

the TE and RE bits in SSER to 0 to set the initial values.

Note: Before changing operating modes and communications formats, clear both the TE bits to 0. Although clearing the TE bit to 0 sets the TDRE bit to 1, clearing the RE does not change the values of the RDRF and ORER bits and SSRDR. Those bits r previous values.

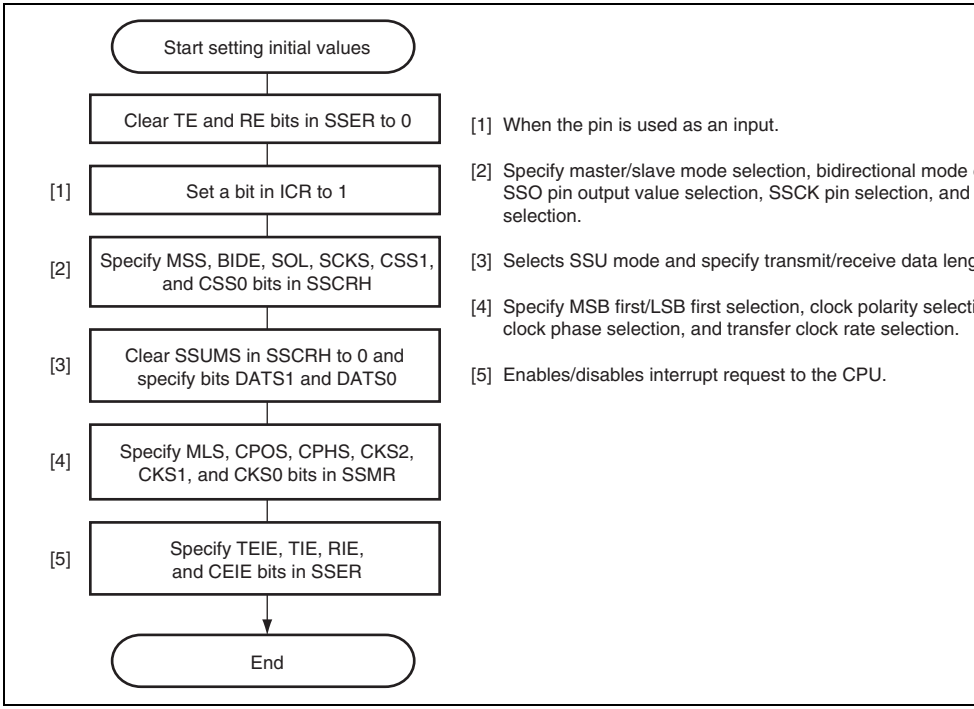
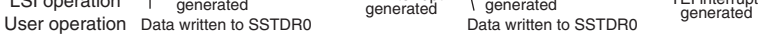


Figure 17.4 Example of Initial Settings in SSU Mode

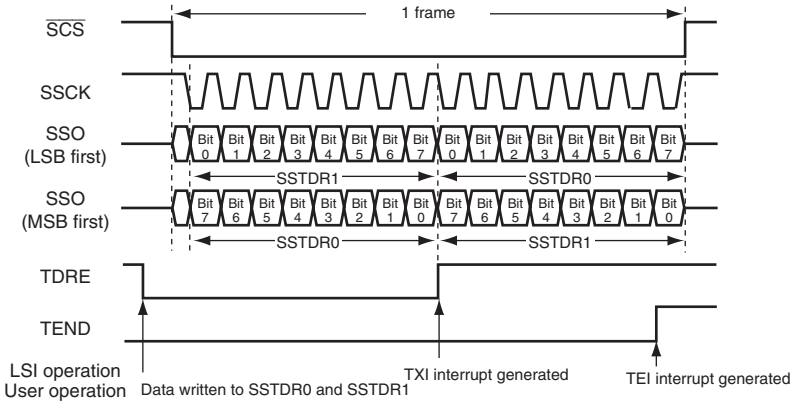
the SSTDR contents are transferred to SSTRSR. After that, the SSU sets the TDRE bit t
starts transmission. At this time, if the TIE bit in SSER is set to 1, a TXI interrupt is gen

When 1-frame data has been transferred with TDRE = 0, the SSTDR contents are transf
SSTRSR to start the next frame transmission. When the 8th bit of transmit data has beer
transferred with TDRE = 1, the TEND bit in SSSR is set to 1 and the state is retained. A
if the TEIE bit is set to 1, a TEI interrupt is generated. After transmission, the output lev
SSCK pin is fixed high when CPOS = 0 and low when CPOS = 1.

While the ORER bit in SSSR is set to 1, transmission is not performed. Check that the C
is cleared to 0.



(2) When 16-bit data length is selected (SSTDR0 and SSTDR1 are valid) with CPOS = 0 and CPHS = 0



(3) When 32-bit data length is selected (SSTDR0, SSTDR1, SSTDR2, and SSTDR3 are valid) with CPOS = 0 and CPHS = 0

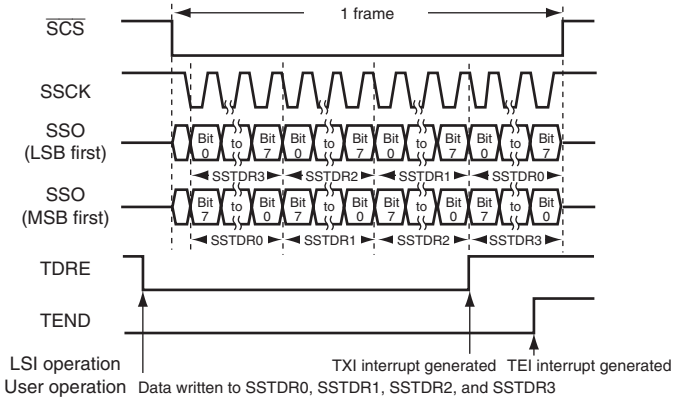
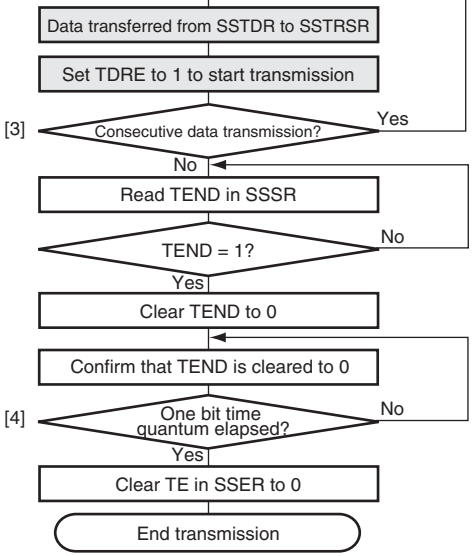


Figure 17.5 Example of Transmission Operation (SSU Mode)



To end data transmission, confirm that the TEND b
to 0. After completion of transmitting the last bit, cl
bit to 0.

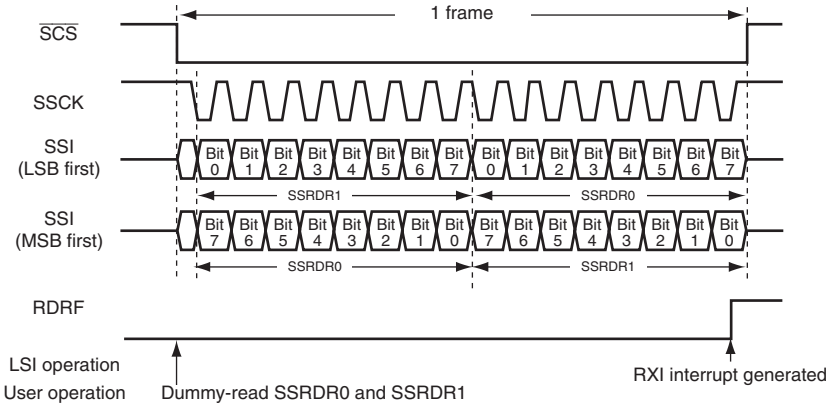
Note: Hatching boxes represent SSU internal op

Figure 17.6 Flowchart Example of Data Transmission (SSU Mode)

The RDRF bit is automatically cleared to 0 by reading SSRDR.

When the RDRF bit has been set to 1 at the 8th rising edge of the transfer clock, the ORE
SSSR is set to 1. This indicates that an overrun error (OEI) has occurred. At this time, dat
reception is stopped. While the ORER bit in SSSR is set to 1, reception is not performed.
resume the reception, clear the ORER bit to 0.

(2) When 16-bit data length is selected (SSRDR0 and SSRDR1 are valid) with CPOS = 0 and CPHS



(3) When 32-bit data length is selected (SSRDR0, SSRDR1, SSRDR2, and SSRDR3 are valid) with CPOS = 0 and CPHS = 0

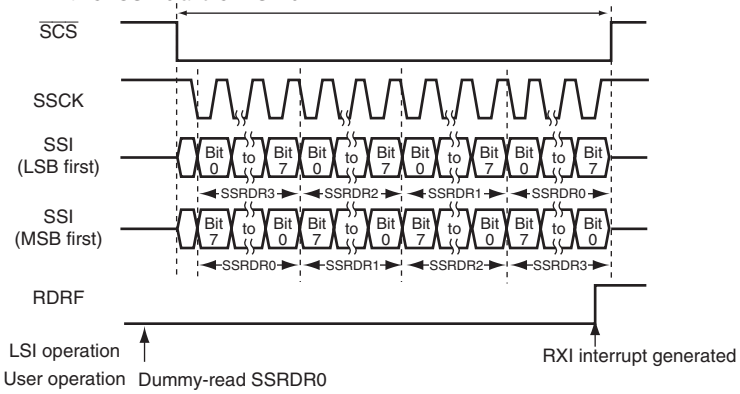
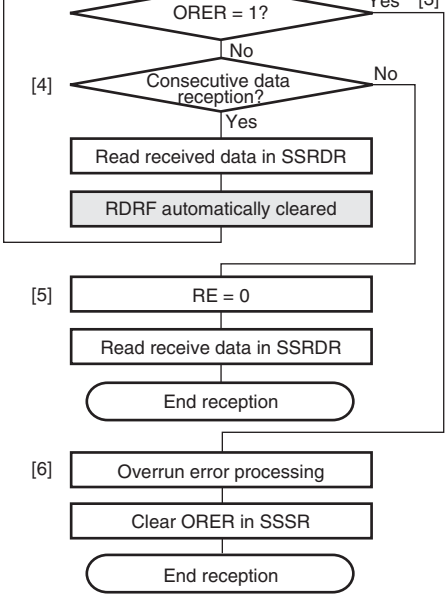


Figure 17.7 Example of Reception Operation (SSU Mode)



Note: Hatching boxes represent SSU internal operations.

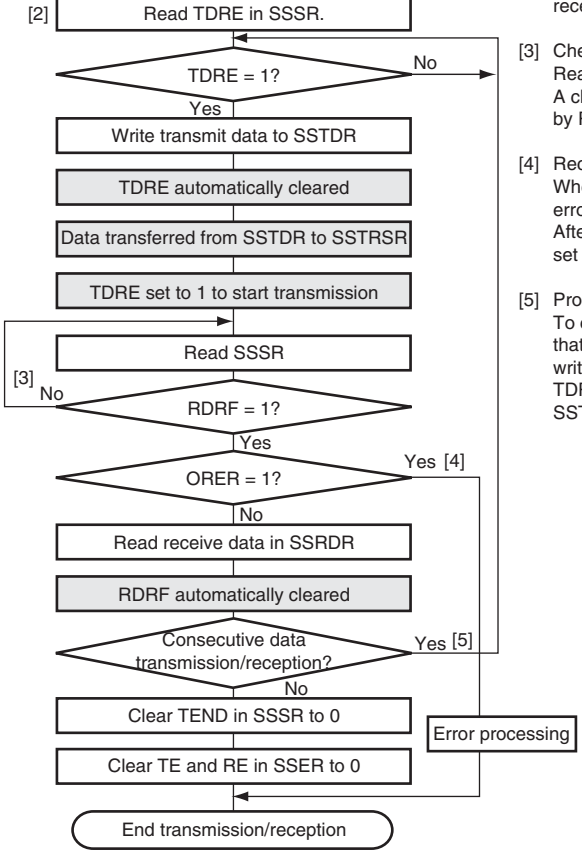
When continuing single reception, wait for time o
while the RDRF flag is set to 1 and then read rec
in SSRDR.
The next single reception starts after reading rec
in SSRDR.

[5] To complete reception:
To complete reception, read receive data after cl
RE bit to 0. When reading SSRDR without clearin
bit, reception is resumed.

Figure 17.8 Flowchart Example of Data Reception (SSU Mode)

(4) Data Transmission/Reception

Figure 17.9 shows a flowchart example of simultaneous transmission/reception. The data
transmission/reception is performed combining the data transmission and data reception a
mentioned above. The data transmission/reception is started by writing transmit data to S
with TE = RE = 1.



[3] Check the SSU state:
Read SSSR confirming that the RDRF bit is 1.
A change of the RDRF bit (from 0 to 1) can be
by RXI interrupt.

[4] Receive error processing:
When a receive error occurs, execute the desi
error processing after reading the ORER bit in
After that, clear the ORER bit to 0. While the C
set to 1, transmission or reception is not resun

[5] Procedure for consecutive data transmission/r
To continue serial data transmission/reception
that the TDRE bit is 1 meaning that SSTDR is
written to. After that, data can be written to SS
TDRE bit is automatically cleared to 0 by writin
SSTDR.

Note: Hatching boxes represent SSU internal operations.

Figure 17.9 Flowchart Example of Simultaneous Transmission/Reception (SSU

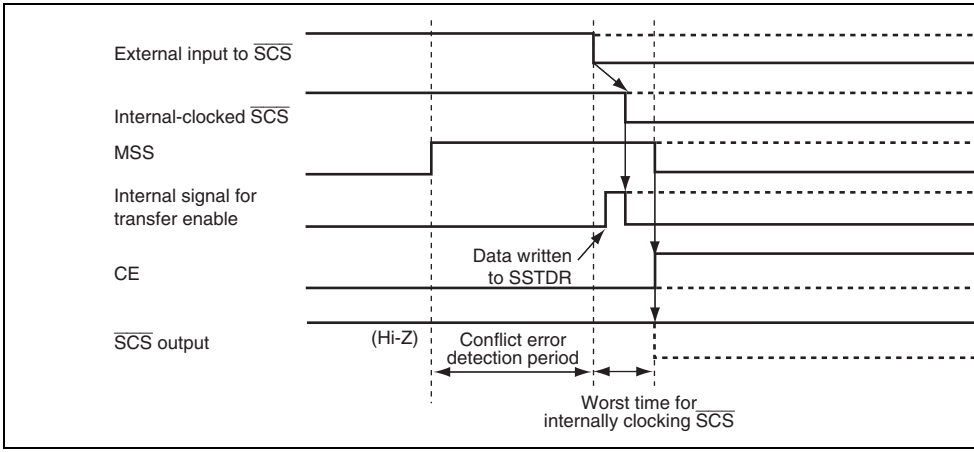


Figure 17.10 Conflict Error Detection Timing (Before Transfer)

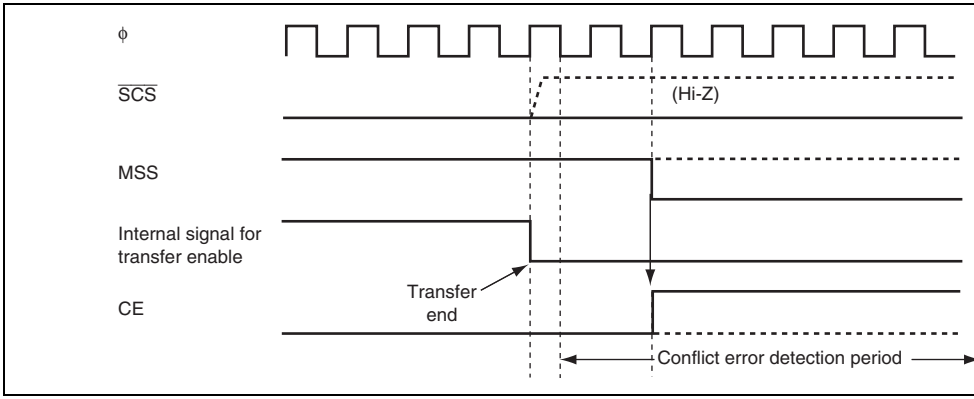


Figure 17.11 Conflict Error Detection Timing (After Transfer End)

bits to 0. Although clearing the TE bit to 0 sets the TDRE bit to 1, clearing the R
does not change the values of the RDRF and ORER bits and SSRDR. Those bits
previous values.

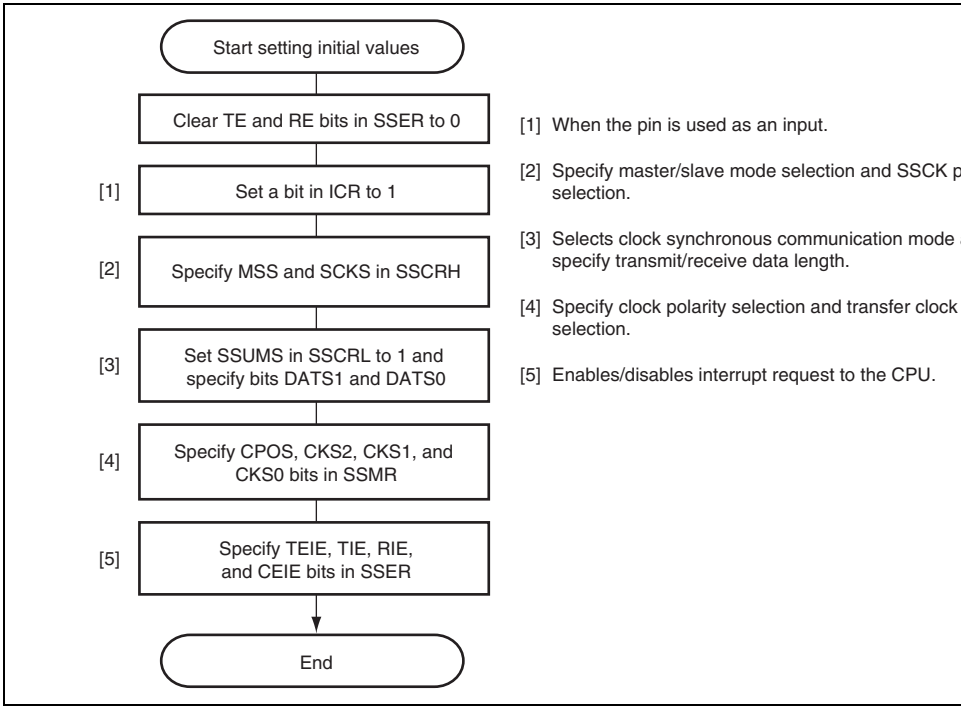


Figure 17.12 Example of Initial Settings in Clock Synchronous Communicatio

When 1-frame data has been transferred with TDRE = 0, the SSTDR contents are transfe
SSTRSR to start the next frame transmission. When the 8th bit of transmit data has been
transferred with TDRE = 1, the TEND bit in SSSR is set to 1 and the state is retained. At
if the TEIE bit is set to 1, a TEI interrupt is generated.

While the ORER bit in SSSR is set to 1, transmission is not performed. Check that the OR
is cleared to 0.

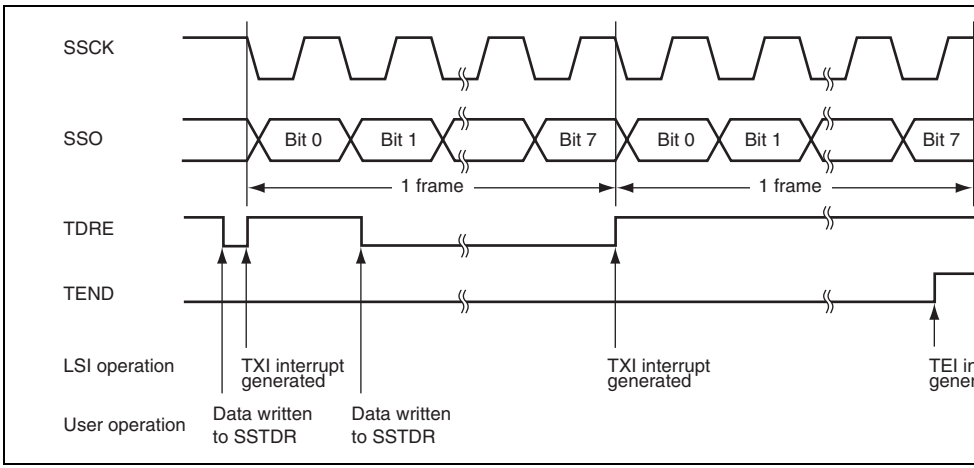
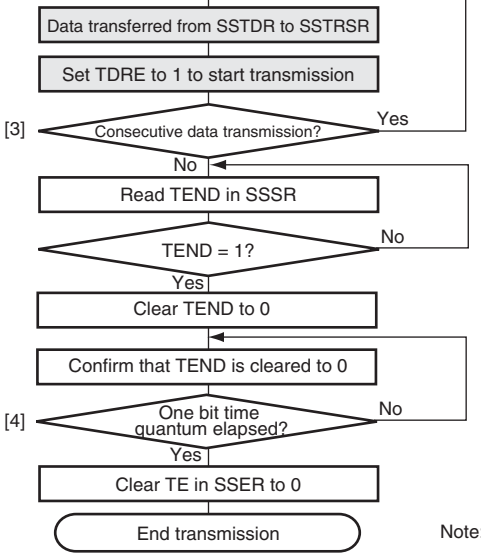


Figure 17.13 Example of Transmission Operation (Clock Synchronous Communication Mode)



To end data transmission, confirm that the TEND b
to 0. After completion of transmitting the last bit, cl
bit to 0.

Note: Hatching boxes represent SSU internal operations.

Figure 17.14 Flowchart Example of Transmission Operation (Clock Synchronous Communication Mode)

bit is automatically cleared to 0 by reading SSRDR.

When the RDRF bit has been set to 1 at the 8th rising edge of the transfer clock, the ORE
SSSR is set to 1. This indicates that an overrun error (OEI) has occurred. At this time, dat
reception is stopped. While the ORER bit in SSSR is set to 1, reception is not performed.
resume the reception, clear the ORER bit to 0.

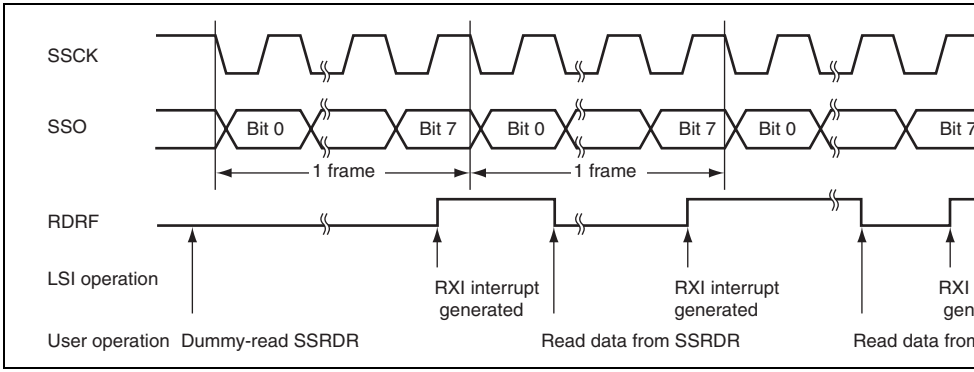
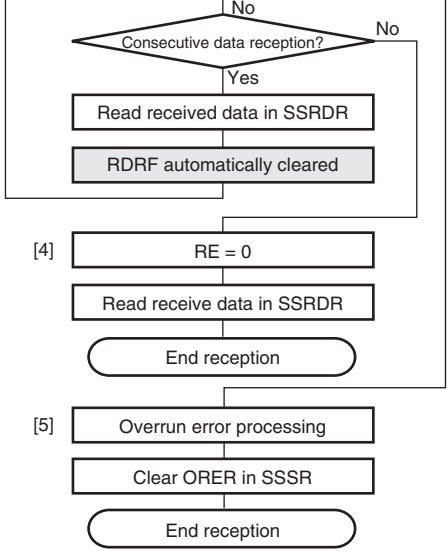


Figure 17.15 Example of Reception Operation (Clock Synchronous Communication Mode)

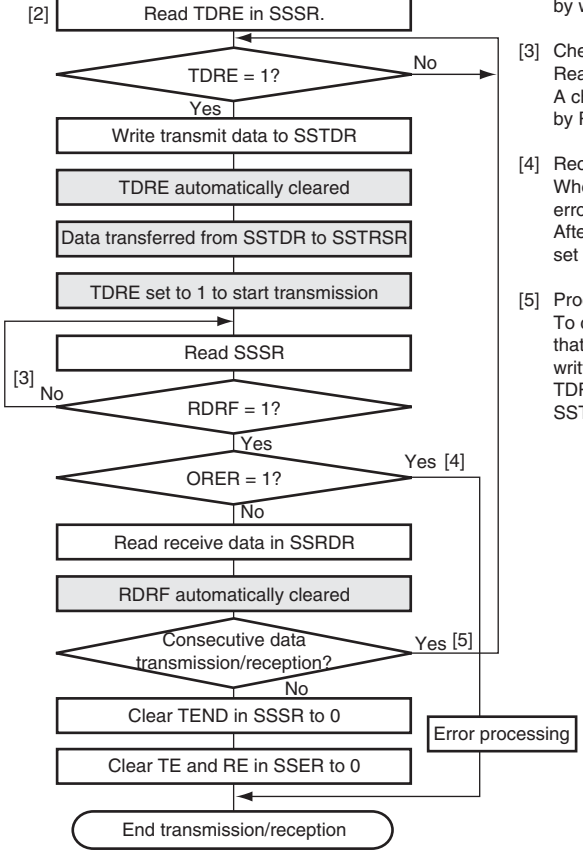


Note: Hatching boxes represent SSU internal operations.

Figure 17.16 Flowchart Example of Data Reception (Clock Synchronous Communication Mode)

(4) Data Transmission/Reception

Figure 17.17 shows a flowchart example of simultaneous transmission/reception. The da transmission/reception is performed combining the data transmission and data reception mentioned above. The data transmission/reception is started by writing transmit data to S with TE = RE = 1.



[3] Check the SSU state:
Read SSSR confirming that the RDRF bit is 1.
A change of the RDRF bit (from 0 to 1) can be r
by RXI interrupt.

[4] Receive error processing:
When a receive error occurs, execute the desig
error processing after reading the ORER bit in S
After that, clear the ORER bit to 0. While the OR
set to 1, transmission or reception is not resume

[5] Procedure for consecutive data transmission/re
To continue serial data transmission/reception,
that the TDRE bit is 1 meaning that SSTDR is r
written to. After that, data can be written to SST
TDRE bit is automatically cleared to 0 by writing
SSTDR.

Note: Hatching boxes represent SSU internal operations.

Figure 17.17 Flowchart Example of Simultaneous Transmission/Reception (Clock Synchronous Communication Mode)

When an interrupt condition shown in table 17.5 is satisfied, an interrupt is requested. C
interrupt source by CPU or DTC data transfer.

Table 17.5 Interrupt Sources

Abbreviation	Interrupt Source	Symbol	Interrupt Condition	DTC Act
SSERI	Overrun error	OEI	(RIE = 1) • (ORER = 1)	—
	Conflict error	CEI	(CEIE = 1) • (CE = 1)	—
SSRXI	Receive data register full	RXI	(RIE = 1) • (RDRF = 1)	Yes
SSTXI	Transmit data register empty	TXI	(TIE = 1) • (TDRE = 1)	Yes
	Transmit end	TEI	(TEIE = 1) • (TEND = 1)	Yes

17.6 Usage Note

17.6.1 Setting of Module Stop Mode

The SSU can be enabled/disabled by the module stop control register setting and is disal
initial value. Canceling module stop mode enables to access the SSU registers. For detai
section 28, Power-Down Modes.

— Clocked synchronous serial format: non-addressing format without acknowledge
master operation only

- Conforms to Philips I^2C bus interface (I^2C bus format)
- Two ways of setting slave address (I^2C bus format)
- Start and stop conditions generated automatically in master mode (I^2C bus format)
- Selection of acknowledge output levels when receiving (I^2C bus format)
- Automatic loading of acknowledge bit when transmitting (I^2C bus format)
- Wait function in master mode (I^2C bus format)
  - A wait can be inserted by driving the SCL pin low after data transfer, excluding acknowledgement.
  - The wait can be cleared by clearing the interrupt flag.
- Wait function (I^2C bus format)
  - A wait request can be generated by driving the SCL pin low after data transfer.
  - The wait request is cleared when the next transfer becomes possible.
- Interrupt sources
  - Data transfer end (including when a transition to transmit mode with I^2C bus forn when ICDR data is transferred, or during a wait state)
  - Address match: when any slave address matches or the general call address is rec slave receive mode with I^2C bus format (including address reception after loss of arbitration)
  - Arbitration loss
  - Start condition detection (in master mode)
  - Stop condition detection (in slave mode)
- Selection of 32 internal clocks (in master mode)
- Direct bus drive

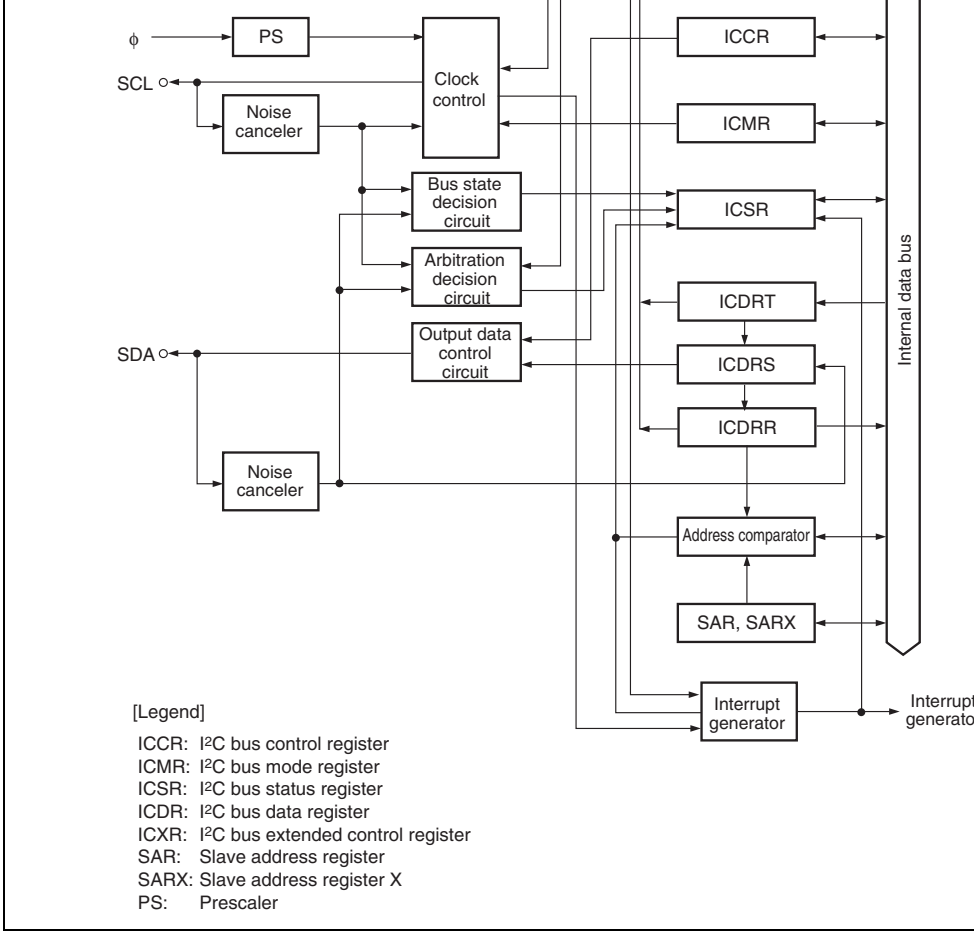


Figure 18.1 Block Diagram of I²C Bus Interface

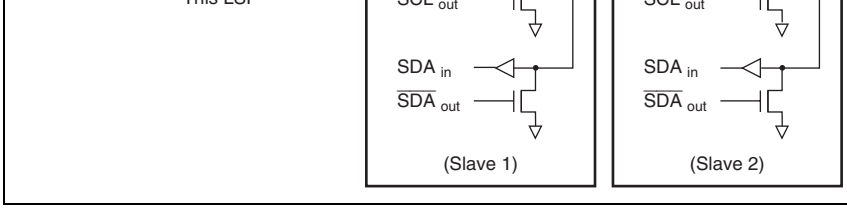


Figure 18.2 I²C Bus Interface Connections (Example: This LSI as Master

	SDA1	Input/Output	Data input/output pin of channel IIC
2	SCL2	Input/Output	Clock input/output pin of channel IIC
	SDA2	Input/Output	Data input/output pin of channel IIC
3	SCL3	Input/Output	Clock input/output pin of channel IIC
	SDA3	Input/Output	Data input/output pin of channel IIC
4	SCL4	Input/Output	Clock input/output pin of channel IIC
	SDA4	Input/Output	Data input/output pin of channel IIC
5	SCL5	Input/Output	Clock input/output pin of channel IIC
	SDA5	Input/Output	Data input/output pin of channel IIC

Note: * In the text, the channel subscript is omitted, and only SCL and SDA are used.

- I²C bus mode register (ICMR)
- I²C bus transfer rate select register (IICX3)
- I²C bus control register (ICCR)
- I²C bus status register (ICSR)
- I²C bus extended control register (ICXR)
- I²C SMbus control register (ICSMBCR)

18.3.1 I²C Bus Data Register (ICDR)

ICDR is an 8-bit readable/writable register that is used as a transmit data register when transmitting and a receive data register when receiving. ICDR is divided internally into a register (ICDRS), receive buffer (ICDRR), and transmit buffer (ICDRT). Data transfers three registers are performed automatically in accordance with changes in the bus state, affect the status of internal flags such as ICDRE and ICDRF.

In master transmit mode with the I²C bus format, writing transmit data to ICDR should b performed after start condition detection. When the start condition is detected, previous is ignored. In slave transmit mode, writing should be performed after the slave addresses and the TRS bit is automatically changed to 1.

If IIC is in transmit mode (TRS = 1) and the next data is in ICDRT (the ICDRE flag is 0 transferred automatically from ICDRT to ICDRS, following transmission of one frame o using ICDRS. When the ICDRE flag is 1 and the next transmit data writing is waited, da transferred automatically from ICDRT to ICDRS by writing to ICDR. If IIC is in receiv (TRS = 0), no data is transferred from ICDRT to ICDRS. Note that data should not be w ICDR in receive mode.

Reading receive data from ICDR is performed after data is transferred from ICDRS to IC

ICDR can be written to and read from only when the ICE bit is set to 1 in ICCR. The initi
of ICDR is undefined.

18.3.2 Slave Address Register (SAR)

SAR sets the slave address and selects the communication format. When the LSI is in sla
with the I^2C bus format selected, if the FS bit is set to 0 and the upper 7 bits of SAR matc
upper 7 bits of the first frame received after a start condition, the LSI operates as the slav
specified by the master device. SAR can be accessed only when the ICE bit in ICCR is cl
0.

Bit	Bit Name	Initial Value	R/W	Description
7	SVA6	All 0	R/W	Slave Addresses 6 to 0
6	SVA5			Set a slave address.
5	SVA4			
4	SVA3			
3	SVA2			
2	SVA1			
1	SVA0			
0	FS	0	R/W	Format Select Selects the communication format together with t bit in SARX. Refer to table 18.2. This bit should be set to 0 when general call addr recognition is performed.

7	SVAX6	All 0	R/W	Second Slave Addresses 6 to 0
6	SVAX5			Set the second slave address.
5	SVAX4			
4	SVAX3			
3	SVAX2			
2	SVAX1			
1	SVAX0			
0	FSX	1	R/W	Format Select X Selects the communication format together with in SAR. Refer to table 18.2.

		• General call address recognized
1	0	I^2C bus format • SAR slave address ignored • SARX slave address recognized • General call address ignored
	1	Clocked synchronous serial format • SAR and SARX slave addresses ignored • General call address ignored

- I^2C bus format: addressing format with acknowledge bit
- Clocked synchronous serial format: non-addressing format without acknowledge bit, master mode only

				Set this bit to 0 when the I²C bus format is used.
6	WAIT	0	R/W	Wait Insertion Bit This bit is valid only in master mode with the I²C format. 0: Data and the acknowledge bit are transferred consecutively with no wait inserted. 1: After the fall of the clock for the final data bit (clock), the IRIC flag is set to 1 in ICCR, and a begins (with SCL at the low level). When the I is cleared to 0 in ICCR, the wait ends and the acknowledge bit is transferred. For details, refer to section 18.4.7, IRIC Setting and SCL Control.
5	CKS2	All 0	R/W	Transfer Clock Select
4	CKS1			These bits are used only in master mode.
3	CKS0			These bits select the required transfer clock rate with bits IICX5 (channel 5), IICX4 (channel 4), a (channel 3) in the IICX3 register and bits IICX2 (2), IICX1 (channel 1), and IICX0 (channel 0) in t register. Refer to table 18.3.

I²C Bus Format	Clocked Synchronous Serial Format
B'000: 9 bits	B'000: 8 bits
B'001: 2 bits	B'001: 1 bits
B'010: 3 bits	B'010: 2 bits
B'011: 4 bits	B'011: 3 bits
B'100: 5 bits	B'100: 4 bits
B'101: 6 bits	B'101: 5 bits
B'110: 7 bits	B'110: 6 bits
B'111: 8 bits	B'111: 7 bits

				This bit selects a clock rate to be applied to the transfer rate. 0: ϕ/2 1: ϕ/4
2	IICX5	0	R/W	IIC Transfer Rate Select 5, 4, 3
1	IICX4	0	R/W	These bits are used to control IIC_5 to IIC_3 op
0	IICX3	0	R/W	These bits select the transfer rate in master mo together with the CKS2 to CKS0 bits in ICMR. F transfer rate, see table 18.3.

	1	0	0	φ/80	250.0 kHz	312.5 kHz	425.0 kH
			1	φ/100	200.0 kHz	250.0 kHz	340.0 kH
		1	0	φ/112	178.6 kHz	223.2 kHz	303.6 kH
			1	φ/128	156.3 kHz	195.3 kHz	265.6 kH
1	0	0	0	φ/56	357.1 kHz	446.4 kHz*	607.1 kH
			1	φ/80	250.0 kHz	312.5 kHz	425.0 kH
		1	0	φ/96	208.3 kHz	260.4 kHz	354.2 kH
			1	φ/128	156.3 kHz	195.3 kHz	265.6 kH
	1	0	0	φ/160	125.0 kHz	156.3 kHz	212.5 kH
			1	φ/200	100.0 kHz	125.0 kHz	170.0 kH
		1	0	φ/224	89.3 kHz	111.6 kHz	151.8 kH
			1	φ/256	78.1 kHz	97.7 kHz	132.8 kH

Note: * The correct operation cannot be guaranteed since the value is outside the I^2C interface specifications (high-speed mode: max. 400 kHz).
(n = 0 to 5)

	1	0	0	φ/160	125.0 kHz	156.3 kHz	212.5 k
			1	φ/200	100.0 kHz	125.0 kHz	170.0 k
		1	0	φ/224	89.3 kHz	111.6 kHz	151.8 k
			1	φ/256	78.1 kHz	97.7 kHz	132.8 k
1	0	0	0	φ/112	178.6 kHz	223.2 kHz	303.6 k
			1	φ/160	125.0 kHz	156.3 kHz	212.5 k
		1	0	φ/190	104.2 kHz	130.2 kHz	177.1 k
			1	φ/256	78.1 kHz	97.7 kHz	132.8 k
	1	0	0	φ/320	62.5 kHz	78.1 kHz	106.3 k
			1	φ/400	50.0 kHz	62.5 kHz	85.0 kH
		1	0	φ/448	44.6 kHz	55.8 kHz	75.9 kH
			1	φ/512	39.1 kHz	48.8 kHz	66.4 kH

Note: * The correct operation cannot be guaranteed since the value is outside the I^2C interface specifications (high-speed mode: max. 400 kHz).
(n = 0 to 5)

				reception, they are connected to the SCL and S and the I²C bus can be driven. ICMR and ICDR accessed.
6	IEIC	0	R/W	I²C Bus Interface Interrupt Enable 0: Disables interrupts from the I²C bus interface to CPU. 1: Enables interrupts from the I²C bus interface to CPU.
5	MST	0	R/W	Master/Slave Select
4	TRS	0	R/W	Transmit/Receive Select 00: Slave receive mode 01: Slave transmit mode 10: Master receive mode 11: Master transmit mode Both these bits will be cleared by hardware when lose in a bus contention in master mode of the I²C format. In slave receive mode with I²C bus format bit in the first frame immediately after the start co automatically sets these bits in receive mode or tr mode by hardware. Modification of the TRS bit during transfer is defe transfer is completed, and the changeover is mad completion of the transfer.

				MST clearing condition 2) [TRS clearing conditions] (1) When 0 is written by software (except for TR condition 3) (2) When 0 is written in TRS after reading TRS = TRS setting condition 3) (3) When lost in bus contention in I^2C bus format mode [TRS setting conditions] (1) When 1 is written by software (except for TR condition 3) (2) When 1 is written in TRS after reading TRS = TRS clearing condition 3) (3) When 1 is received as the R/$\overline{W}$ bit after the fi address matching in I^2C bus format slave mo
3	ACKE	0	R/W	Acknowledge Bit Decision Selection 0: The value of the acknowledge bit is ignored, continuous transfer is performed. The value received acknowledge bit is not indicated by bit in ICSR, which is always 0. 1: If the acknowledge bit is 1, continuous transf halted. Depending on the receiving device, the acknowl may be significant, in indicating completion of pr of the received data, for instance, or may be fixe have no significance.

- Writing to the BBSY flag is disabled.

[BBSY setting condition]

- When the SDA level changes from high to low the condition of SCL = high, assuming that the condition has been issued.

[BBSY clearing conditions]

- When the SDA level changes from low to high the condition of SCL = high, assuming that the condition has been issued.

To issue a start/stop condition, use the MOV instr

The I^2C bus interface must be set in master transm before the issue of a start condition. Set MST to 1 TRS to 1 before writing 1 in BBSY and 0 in SCP.

The BBSY flag can be read to check whether the (SCL, SDA) is busy or free.

Note: * Even if the BBSY bit is written to, the value of the flag does not change.

- When a start condition is detected in the bus l after a start condition is issued (when the ICD set to 1 because of first frame transmission)
- When a wait is inserted between the data and acknowledge bit when the WAIT bit is 1 (fall o transmit/receive clock)
- At the end of data transfer (rise of the 9th transmit/receive clock)
- When a slave address is received after bus m is lost
- If 1 is received as the acknowledge bit (when bit in ICSR is set to 1) when the ACKE bit is 1
- When the AL flag is set to 1 after bus masters while the ALIE bit is 1

I^2C bus format slave mode:

- When the slave address (SVA or SVAX) matc (when the AAS or AASX flag in ICSR is set to the end of data transfer up to the subsequent retransmission start condition or stop conditio detection (rise of the 9th clock)
- When the general call address is detected (wl is received for R/$\overline{W}$ bit, and ADZ flag in ICSR 1) and at the end of data reception up to the subsequent retransmission start condition or s condition detection (rise of the 9th receive clo
- When 1 is received as an acknowledge bit wh ACKE bit is 1 (when the ACKB bit is set to 1)
- When a stop condition is detected while the S is 0 (when the STOP or ESTP flag in ICSR is

- When transmitting the data in the ICDR registe
(when data is transferred from ICDRT to ICDR
transmit mode and the ICDRE flag is set to 1,
is transferred from ICDRS to ICDRR in receive
and the ICDRF flag is set to 1.)

[Clearing conditions]

- When 0 is written in IRIC after reading IRIC =
- When ICDR is accessed by DTC * (This may
clearing condition. For details, see the descrip
the DTC operation on the next page.

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

Even when the IRIC flag and IRTR flag are set, the ICDRE or ICDRF flag may not be s
IRIC and IRTR flags are not cleared at the end of the specified number of transfers in co
transfer using the DTC. The ICDRE or ICDRF flag is cleared, however, since the specif
number of ICDR reads or writes have been completed.

Tables 18.4 and 18.5 show the relationship between the flags and the transfer states.

													ICDF
1	1	1	0	0	—	0	0	0	0	0	—	0↓	ICDF the a
1	1	1	0	0	—	0	0	0	0	0	—	1	Tran end ICDF
1	1	1	0	0	—	0	0	0	0	0	—	0↓	ICDF the a or af cond detec
1	1	1	0	0	1↑	0	0	0	0	0	—	1↑	Auto trans ICDF with state
1	0	1	0	0	1↑	0	0	0	0	—	1↑	—	Rece with
1	0	1	0	0	—	0	0	0	0	—	0↓	—	ICDF the a
1	0	1	0	0	—	0	0	0	0	—	1	—	Rece with
1	0	1	0	0	—	0	0	0	0	—	0↓	—	ICDF the a
1	0	1	0	0	1↑	0	0	0	0	—	1↑	—	Auto trans ICDF ICDF abov
0↓	0↓	1	0	0	—	0	1↑	0	0	—	—	—	Arbit
1	—	0↓	0	0	—	0	0	0	0	—	—	0↓	Stop detec

[Legend]

0: 0-state retained 1: 1-state retained —: Previous state retained

0↓: Cleared to 0 1↑: Set to 1

0	1↑/0 *1	1	0	0	1↑	1↑	—	0	0	0	1↑	1	S fi (
0	1	1	0	0	—	—	—	—	0	1↑	—	—	T e a
0	1	1	0	0	1↑/0 *1	—	—	—	0	0	—	1↑	T e I
0	1	1	0	0	—	—	0↓	0↓	0	0	—	0↓	I w s
0	1	1	0	0	—	—	—	—	0	0	—	1	T e I
0	1	1	0	0	—	—	0↓	0↓	0	0	—	0↓	I w s
0	1	1	0	0	1↑/0 *2	—	0	0	0	0	—	1↑	A c fr I tl s
0	0	1	0	0	1↑/0 *2	—	—	—	—	—	1↑	—	P w
0	0	1	0	0	—	—	0↓	0↓	0↓	—	0↓	—	I w s

*3 *3

[Legend]

0: 0-state retained 1: 1-state retained —: Previous state retained

0↓: Cleared to 0 1↑: Set to 1

Notes: 1. Set to 1 when 1 is received as a R/$\overline{W}$ bit following an address.

2. Set to 1 when the AASX bit is set to 1.

3. When ESTP=1, STOP is 0, or when STOP=1, ESTP is 0.

				• When 0 is written in ESTP after reading EST • When the IRIC flag in ICCR is cleared to 0
6	STOP	0	R/(W)*	Normal Stop Condition Detection Flag This bit is valid in I^2C bus format slave mode. [Setting condition] When a stop condition is detected after frame tra completed. [Clearing conditions] • When 0 is written in STOP after reading STO • When the IRIC flag is cleared to 0
5	IRTR	0	R/(W)*	I^2C Bus Interface Continuous Transfer Interrupt F Flag Indicates that the I^2C bus interface has issued an request to the CPU, and the source is completion reception/transmission of one frame in continuou transmission/reception for which DTC activation possible. When the IRTR flag is set to 1, the IRIC also set to 1 at the same time. [Setting conditions] I^2C bus format slave mode: • When the ICDRE or ICDRF flag in ICDR is s when AASX = 1 I^2C bus format master mode or clocked synchron format mode: • When the ICDRE or ICDRF flag is set to 1 [Clearing conditions] • When 0 is written after reading IRTR = 1 • When the IRIC flag is cleared to 0 while ICE

				• When a start condition is detected • In master mode
3	AL	0	R/(W)*	Arbitration Lost Flag Indicates that arbitration was lost in master mode. [Setting conditions] When ALSL=0 • If the internal SDA and SDA pin disagree at th SCL in master transmit mode • If the internal SCL line is high at the fall of SCL master mode When ALSL=1 • If the internal SDA and SDA pin disagree at th SCL in master transmit mode • If the SDA pin is driven low by another device the I^2C bus interface drives the SDA pin low, a start condition instruction was executed in mas transmit mode [Clearing conditions] • When ICDR is written to (transmit mode) or re (receive mode) • When 0 is written in AL after reading AL = 1

				[Clearing conditions] • When ICDR is written to (transmit mode) or r (receive mode) • When 0 is written in AAS after reading AAS = • In master mode
1	ADZ	0	R/(W)*	General Call Address Recognition Flag In I^2C bus format slave receive mode, this flag is the first frame following a start condition is the g address (H'00). [Setting condition] When the general call address (one frame includ R/$\overline{W}$ bit is H'00) is detected in slave receive mod = 0 or FSX = 0 [Clearing conditions] • When ICDR is written to (transmit mode) or r (receive mode) • When 0 is written in ADZ after reading ADZ = • In master mode If a general call address is detected while FS=1 FSX=0, the ADZ flag is set to 1; however, the ge address is not recognized (AAS flag is not set to

ACKE=1 in transmit mode

- When 0 is written to the ACKE bit

Receive mode:

0: Returns 0 as acknowledge data after data rece

1: Returns 1 as acknowledge data after data rece

When this bit is read, the value loaded from the bu
(returned by the receiving device) is read in transm
(when TRS = 1). In reception (when TRS = 0), the
set by internal software is read.

When this bit is written, acknowledge data that is
after receiving is rewritten regardless of the TRS v
the ICSR register bit is written using bit-manipulat
instructions, the acknowledge data should be re-s
the acknowledge data setting is rewritten by the A
reading value.

Write the ACKE bit to 0 to clear the ACKB flag to
transmission is ended and a stop condition is issu
master mode, or before transmission is ended and
released to issue a stop condition by a master dev

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

				the stop condition is detected (STOP = 1 or E in slave mode. 1: Disables IRIC flag setting and interrupt gener when the stop condition is detected.
6	HNDS	0	R/W	Handshake Receive Operation Select Enables or disables continuous receive operatio receive mode. 0: Enables continuous receive operation 1: Disables continuous receive operation When the HNDS bit is cleared to 0, receive oper performed continuously after data has been rec successfully while ICDRF flag is 0. When the HNDS bit is set to 1, SCL is fixed to th level after data has been received successfully ICDRF flag is 0; thus disabling the next data to b transferred. The bus line is released and next re operation is enabled by reading the receive data

- When data is received successfully and transf
  from ICDRS to ICDRR.

(1) When data is received successfully while ICD
(at the rise of the 9th clock pulse).

(2) When ICDR is read successfully in receive m
data was received while ICDRF = 1.

[Clearing conditions]

- When ICDR (ICDRR) is read.
- When 0 is written to the ICE bit.

When ICDRF is set due to the condition (2) above
is temporarily cleared to 0 when ICDR (ICDRR) is
however, since data is transferred from ICDRS to
immediately, ICDRF is set to 1 again.

Note that ICDR cannot be read successfully in tra
mode (TRS = 1) because data is not transferred f
ICDRS to ICDRR. Be sure to read data from ICDF
receive mode (TRS = 0).

- When the start condition is detected from the state in I^2C bus format or serial format.
- When data is transferred from ICDRT to ICDR
  1. When data is transmitted completely while = 0 (at the rise of the 9th clock pulse).
  2. When data is written to ICDR completely mode after data was transmitted while ICE

[Clearing conditions]

- When data is written to ICDR (ICDRT).
- When the stop condition is detected in I^2C bus or serial format.
- When 0 is written to the ICE bit.

Note that if the ACKE bit is set to 1 in I^2C bus form enabling acknowledge bit decision, ICDRE is not data is transmitted completely while the acknowle 1.

When ICDRE is set due to the condition (2) above is temporarily cleared to 0 when data is written to (ICDRT); however, since data is transferred from ICDRS immediately, ICDRF is set to 1 again. Do data to ICDR when TRS = 0 because the ICDRE value is invalid during the time.

				interface outputs at the rise of SCL and the SCL driven low by another device. 1: If the SDA pin state disagrees with the data tha interface outputs at the rise of SCL and the SDA driven low by another device in idle state or afte start condition instruction was executed.
1	FNC1	0	R/W	Function Bit
0	FNC0	0	R/W	These bits cancel some restrictions on usage. For refer to section 18.6, Usage Notes. 00: Restrictions on operation remaining in effect 01: Setting prohibited 10: Setting prohibited 11: Restrictions on operation canceled

Bit	Bit Name	Value	R/W	Description
7	SMB5E	0	R/W	SMBus Enable
6	SMB4E	0	R/W	These bits enable/disable to support the SMBus,
5	SMB3E	0	R/W	combination with bits FSEL1 and FSEL0. Bits SM
4	SMB2E	0	R/W	SMB4E, SMB3E, SMB2E, SMB1E, and SMB0E
3	SMB1E	0	R/W	IIC_5, IIC_4, IIC_3, IIC_2, IIC_1, and IIC_0, resp
2	SMB0E	00	R/W	0: Disables to support the SMBus
				1: Enables to support the SMBus
1	FSEL1	0	R/W	Frequency Selection
0	FSEL0	0	R/W	These bits must be specified to match the systen frequency in order to support the SMBus. For de setting, see table 18.6.

1	0	Min.	300	240*	176*
		Max.	550	440	324
	1	Min.	500	400	294*
		Max.	950	760	559

Notes: n = 0 to 5

* Since the value is outside the SMBus specification, it should not be set.

Table 18.7 ISCMBCR Setting

System Clock	SMBnE	FSEL1	FSEL0
20 MHz	1	1	0
20 to 34 MHz	1	1	1

n = 0 to 5

Figure 18.5 shows the I²C bus timing.

The symbols used in figures 18.3 to 18.5 are explained in table 18.8.

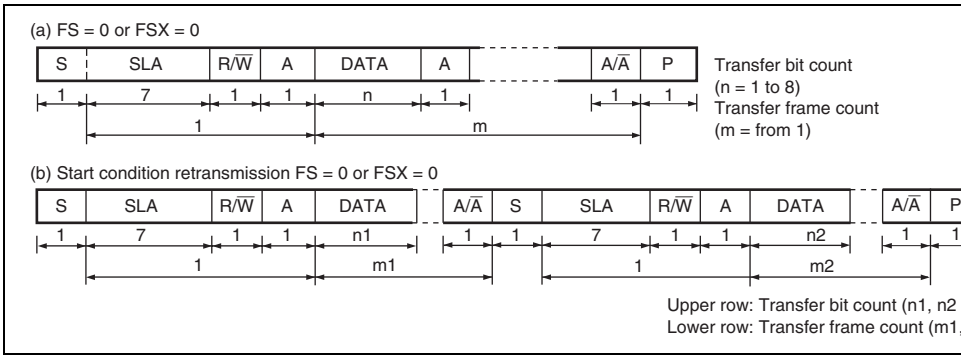


Figure 18.3 I²C Bus Data Formats (I²C Bus Formats)

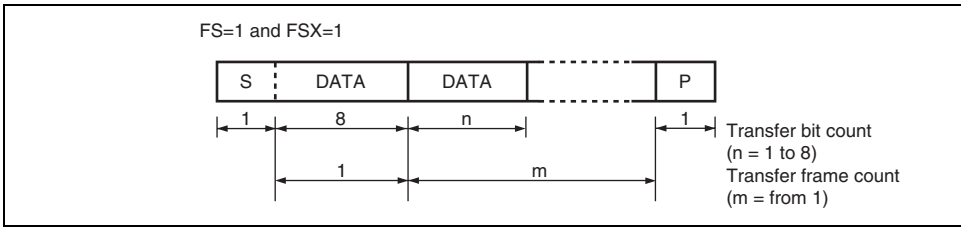


Figure 18.4 I²C Bus Data Formats (Serial Formats)

S	Start condition. The master device drives SDA from high to low while SCL is h
SLA	Slave address. The master device selects the slave device.
R/$\overline{W}$	Indicates the direction of data transfer: from the slave device to the master de when R/$\overline{W}$ is 1, or from the master device to the slave device when R/W is 0
A	Acknowledge. The receiving device drives SDA low to acknowledge a transfe slave device returns acknowledge in master transmit mode, and the master d returns acknowledge in master receive mode.)
DATA	Transferred data. The bit length of transferred data is set with the BC2 to BC0 ICMR. The MSB first or LSB first is switched with the MLS bit in ICMR.
P	Stop condition. The master device drives SDA from low to high while SCL is h

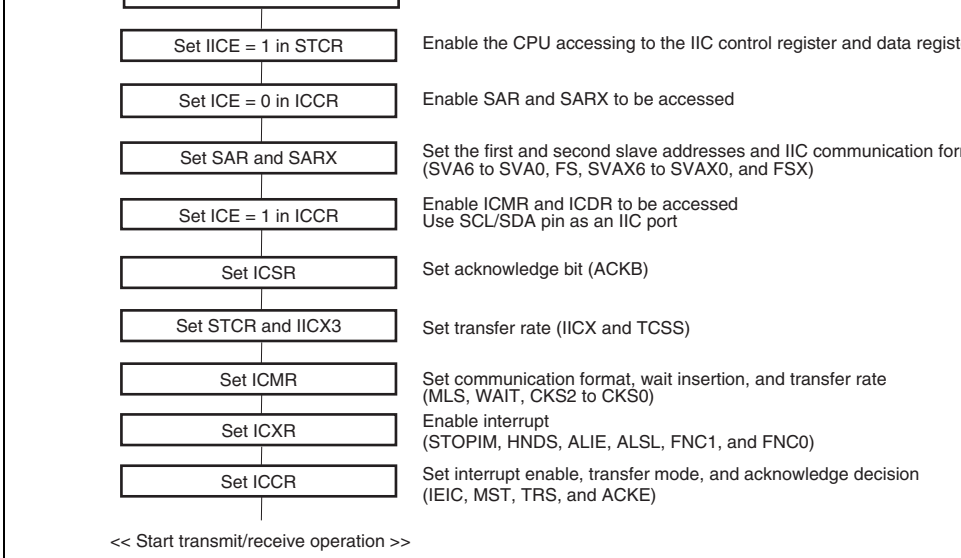


Figure 18.6 Sample Flowchart for IIC Initialization

Note: Be sure to modify the ICMR register after transmit/receive operation has been c
If the ICMR register is modified during transmit/receive operation, bit counter B
BC0 will be modified erroneously, thus causing incorrect operation.

18.4.3 Master Transmit Operation

In I^2C bus format master transmit mode, the master device outputs the transmit clock an
data, and the slave device returns an acknowledge signal.

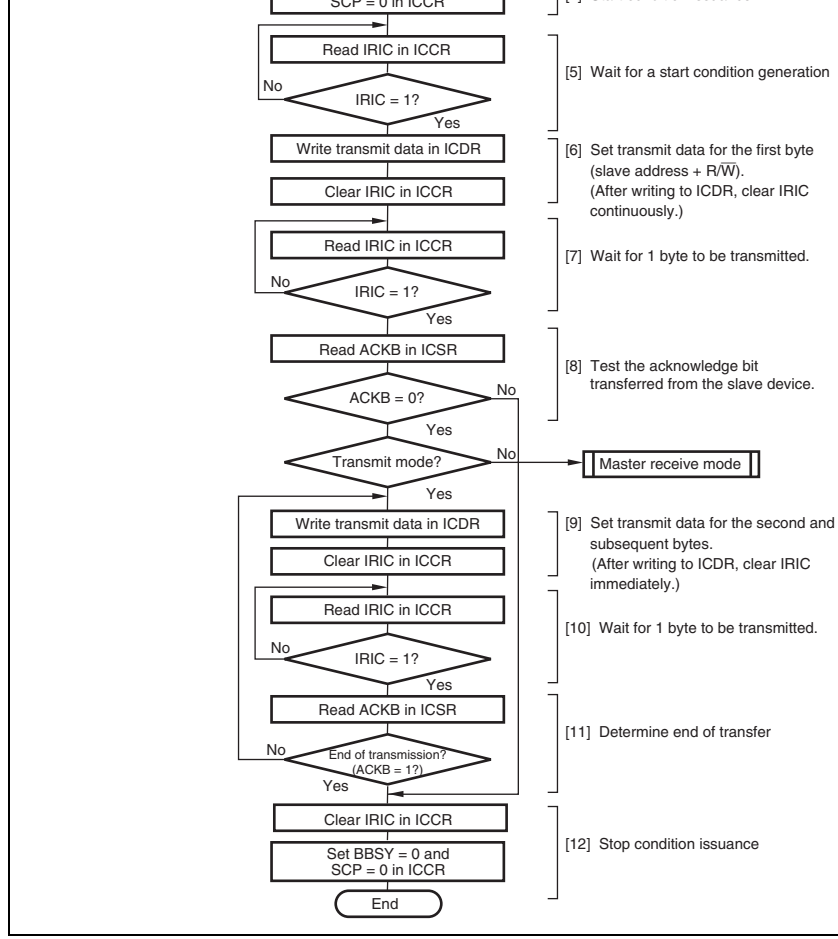


Figure 18.7 Sample Flowchart for Operations in Master Transmit Mode

6. Write the data (slave address + R/W) to ICDR.

   With the I^2C bus format (when the FS bit in SAR or the FSX bit in SARX is 0), the f
   data following the start condition indicates the 7-bit slave address and transmit/recei
   direction (R/$\overline{W}$).

   To determine the end of the transfer, the IRIC flag is cleared to 0. After writing to IC
   IRIC continuously so no other interrupt handling routine is executed. If the time for
   transmission of one frame of data has passed before the IRIC clearing, the end of tra
   cannot be determined. The master device sequentially sends the transmission clock a
   data written to ICDR. The selected slave device (i.e. the slave device with the match
   address) drives SDA low at the 9th transmit clock pulse and returns an acknowledge
7. When one frame of data has been transmitted, the IRIC flag is set to 1 at the rise of t
   transmit clock pulse. After one frame has been transmitted, SCL is automatically fix
   synchronization with the internal clock until the next transmit data is written.
8. Read the ACKB bit in ICSR to confirm that ACKB is cleared to 0. When the slave d
   not acknowledged (ACKB bit is 1), operate step [12] to end transmission, and retry t
   transmit operation.
9. Write the transmit data to ICDR.

   As indicating the end of the transfer, the IRIC flag is cleared to 0. Perform the ICDR
   the IRIC flag clearing sequentially, just as in step [6]. Transmission of the next fram
   performed in synchronization with the internal clock.
10. When one frame of data has been transmitted, the IRIC flag is set to 1 at the rise of t
    transmit clock pulse. After one frame has been transmitted, SCL is automatically fix
    synchronization with the internal clock until the next transmit data is written.
11. Read the ACKB bit in ICSR.

    Confirm that the slave device has been acknowledged (ACKB bit is 0). When there i
    to be transmitted, go to step [9] to continue the next transmission operation. When th
    device has not acknowledged (ACKB bit is set to 1), operate step [12] to end transm

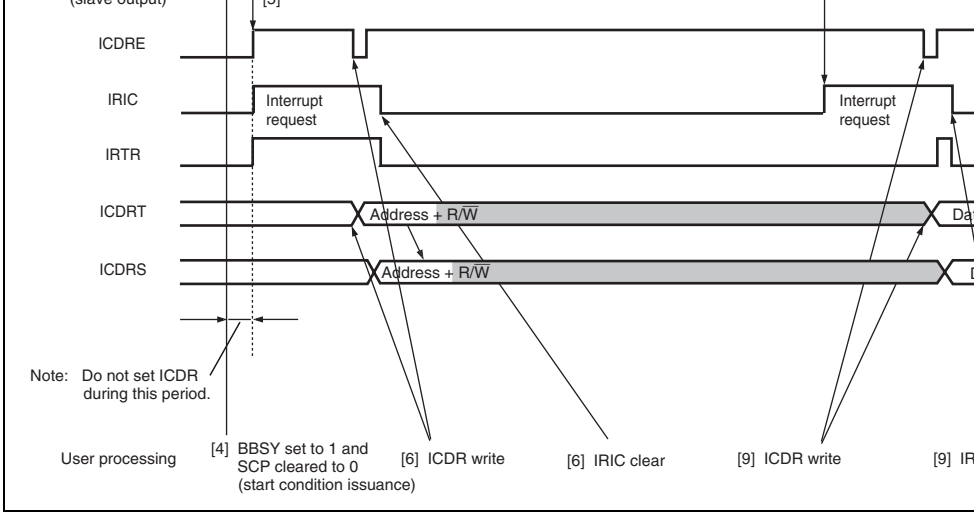


Figure 18.8 Operation Timing Example in Master Transmit Mode (MLS = WAI

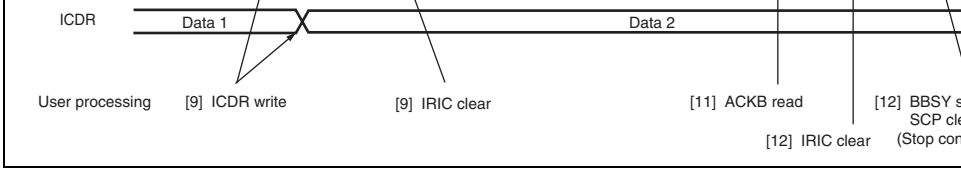


Figure 18.9 Stop Condition Issuance Operation Timing Example in Master Trans (MLS = WAIT = 0)

18.4.4 Master Receive Operation

In I^2C bus format master receive mode, the master device outputs the receive clock, rece and returns an acknowledge signal. The slave device transmits data.

The master device transmits data containing the slave address and R/$\overline{W}$ (1: read) in the f following the start condition issuance in master transmit mode, selects the slave device, switches the mode for receive operation.

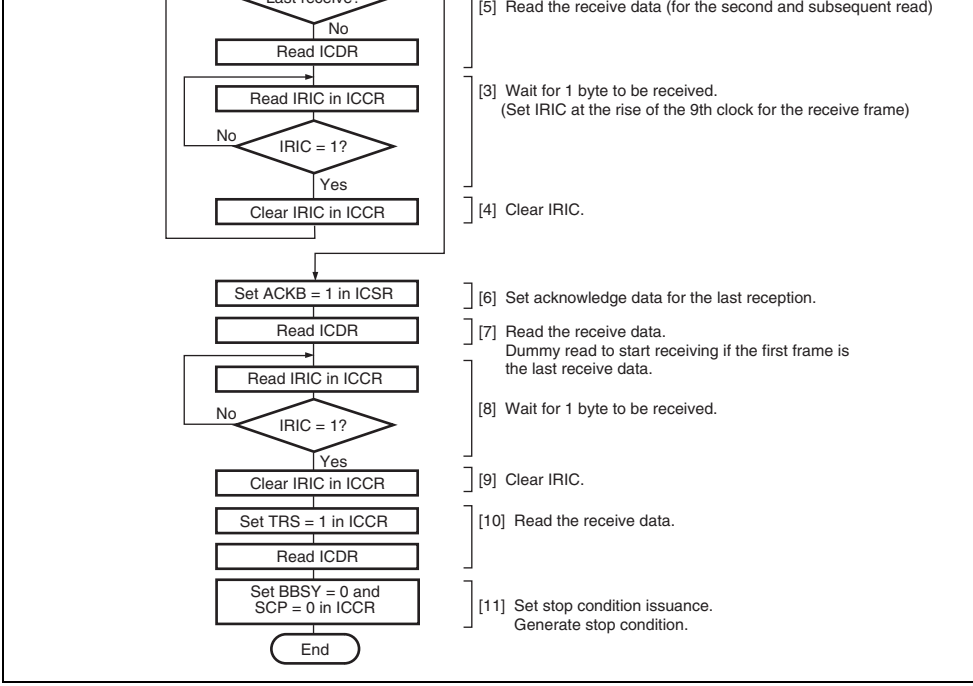


Figure 18.10 Sample Flowchart for Operations in Master Receive Mode (HNDS

sequentially transferred to ICDRS in synchronization with the rise of the receive clo

3. The master device drives SDA low to return the acknowledge data at the 9th receive pulse. The receive data is transferred to ICDRR from ICDRS at the rise of the 9th cl setting the ICDRF, IRIC, and IRTR flags to 1. If the IEIC bit has been set to 1, an i request is sent to the CPU.

   The master device drives SCL low from the fall of the 9th receive clock pulse to the reading.
4. Clear the IRIC flag to determine the next interrupt.

   Go to step [6] to halt reception operation if the next frame is the last receive data.
5. Read ICDR receive data. This clears the ICDRF flag to 0. The master device outputs receive clock continuously to receive the next data.

   Data can be received continuously by repeating steps [3] to [5].
6. Set the ACKB bit to 1 so as to return the acknowledge data for the last reception.
7. Read ICDR receive data. This clears the ICDRF flag to 0. The master device outputs receive clock to receive data.
8. When one frame of data has been received, the ICDRF, IRIC, and IRTR flags are se rise of the 9th receive clock pulse.
9. Clear the IRIC flag to 0.
10. Read ICDR receive data after setting the TRS bit. This clears the ICDRF flag to 0.
11. Clear the BBSY bit and SCP bit to 0 in ICCR. This changes SDA from low to high v is high, and generates the stop condition.

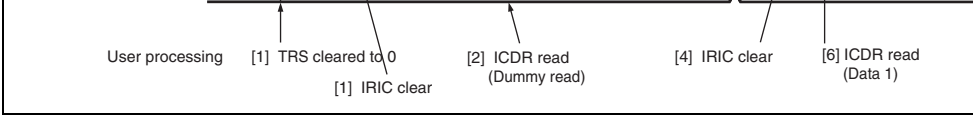


Figure 18.11 Master Receive Mode Operation Timing Example (MLS = WAIT = 0, HNDS = 1)

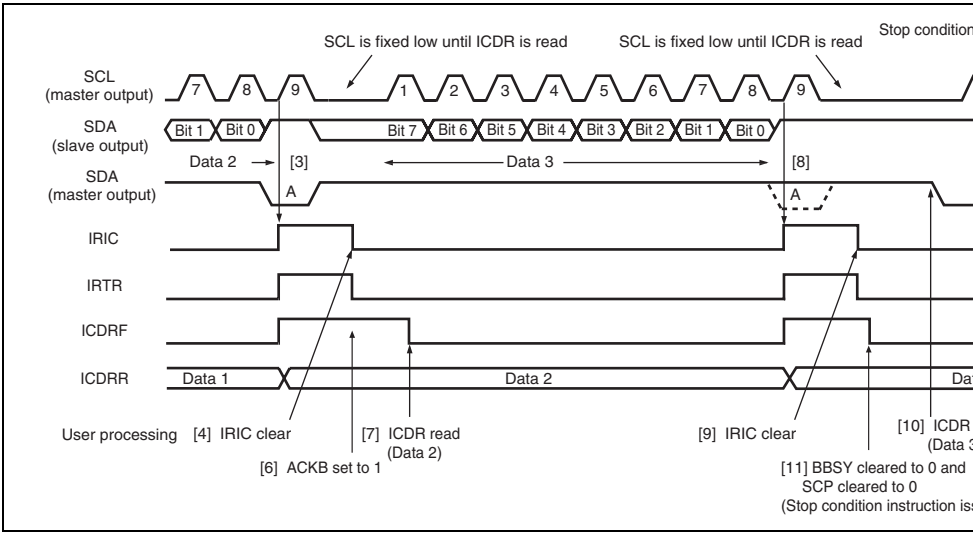


Figure 18.12 Stop Condition Issuance Timing Example in Master Receive Mo (MLS = WAIT = 0, HNDS = 1)

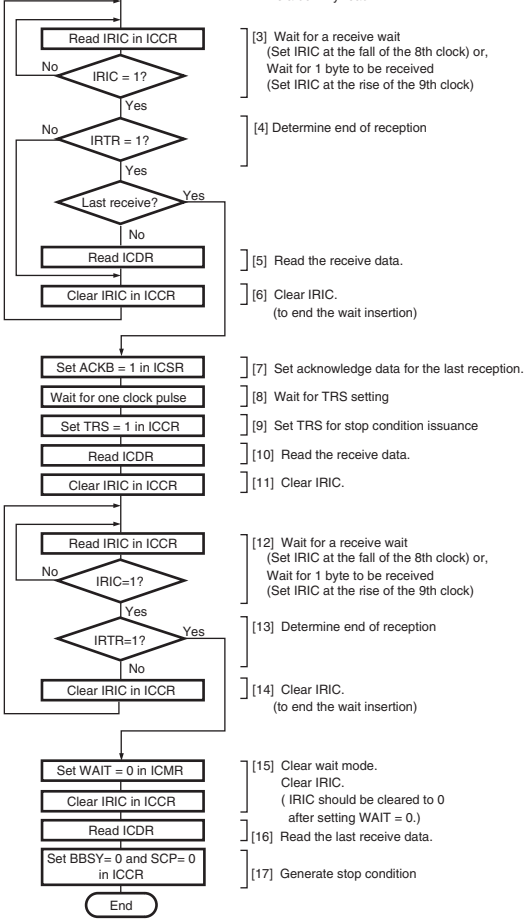


Figure 18.13 Sample Flowchart for Operations in Master Receive Mode (receiving multiple bytes) (WAIT = 1)

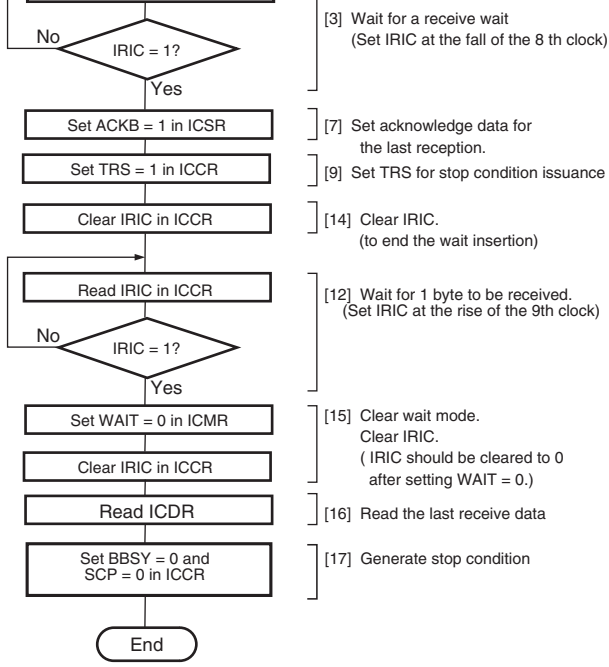


Figure 18.14 Sample Flowchart for Operations in Master Receive Mode (receiving a single byte) (WAIT = 1)

The reception procedure and operations using the wait function (WAIT bit), by which dat sequentially received in synchronization with ICDR (ICDRR) read operations, are descril below.

The following describes the multiple-byte reception procedure. In single-byte reception, s steps of the following procedure are omitted. At this time, follow the procedure shown in 18.14.

flag clearing.

(2) At the rise of the 9th receive clock pulse for one frame

The IRTR and ICDRF flags are set to 1, indicating that one frame of data has received. The master device outputs the receive clock continuously to receive data.

4. Read the IRTR flag in ICSR.

   If the IRTR flag is 0, execute step [6] to clear the IRIC flag to 0 to release the wait st

   If the IRTR flag is 1 and the next data is the last receive data, execute step [7] to halt
5. If IRTR flag is 1, read ICDR receive data.
6. Clear the IRIC flag. When the flag is set as (1) in step [3], the master device outputs clock and drives SDA low at the 9th receive clock pulse to return an acknowledge si

   Data can be received continuously by repeating steps [3] to [6].

7. Set the ACKB bit in ICSR to 1 so as to return the acknowledge data for the last recep
8. After the IRIC flag is set to 1, wait for at least one clock pulse until the rise of the fir pulse for the next receive data.
9. Set the TRS bit in ICCR to 1 to switch from receive mode to transmit mode. The TR becomes valid when the rising edge of the next 9th clock pulse is input.
10. Read the ICDR receive data.
11. Clear the IRIC flag to 0.

condition.

14. If IRTR flag is 0, clear the IRIC flag to 0 to release the wait state.

    Execute step [12] to read the IRIC flag to detect the end of reception.

15. Clear the WAIT bit in ICMR to cancel the wait mode.

    Clearing of the IRIC flag should be done while WAIT = 0. (If the WAIT bit is cleared after clearing the IRIC flag and then an instruction to issue a stop condition is executed stop condition may not be issued correctly.)

16. Read the last ICDR receive data.

17. Clear the BBSY bit and SCP bit to 0 in ICCR. This changes SDA from low to high while is high, and generates the stop condition.

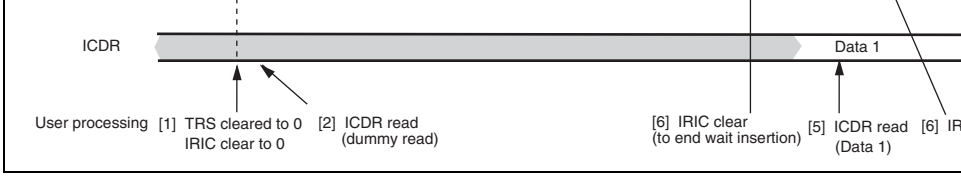


Figure 18.15 Master Receive Mode Operation Timing Example (MLS = ACKB = 0, WAIT = 1)

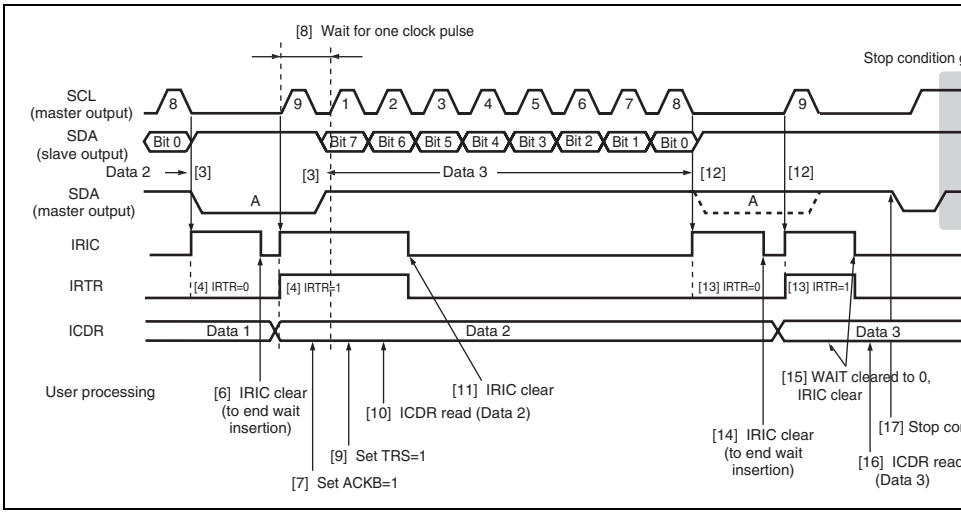


Figure 18.16 Stop Condition Issuance Timing Example in Master Receive M (MLS = ACKB = 0, WAIT = 1)

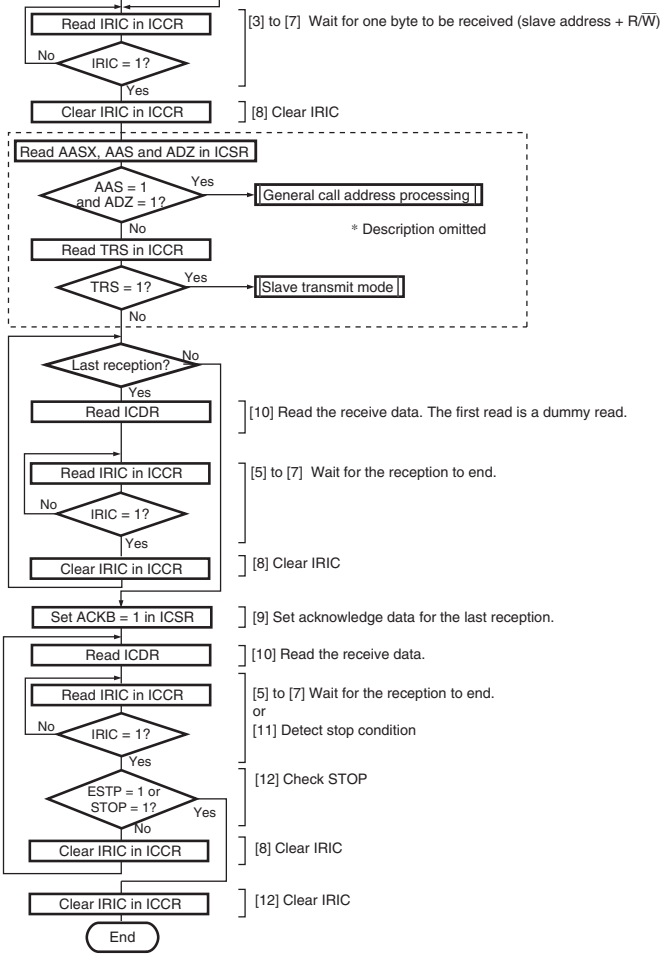


Figure 18.17 Sample Flowchart for Operations in Slave Receive Mode (HND

(R/$\overline{W}$), in synchronization with the transmit clock pulses.

4. When the slave address matches in the first frame following the start condition, the de operates as the slave device specified by the master device. If the 8th data bit (R/$\overline{W}$) is TRS bit remains cleared to 0, and slave receive operation is performed. If the 8th data (R/$\overline{W}$) is 1, the TRS bit is set to 1, and slave transmit operation is performed. When th address does not match, receive operation is halted until the next start condition is det
5. At the 9th clock pulse of the receive frame, the slave device returns the data in the AC as the acknowledge data.
6. At the rise of the 9th clock pulse, the IRIC flag is set to 1. If the IEIC bit has been set interrupt request is sent to the CPU.

   If the AASX bit has been set to 1, IRTR flag is also set to 1.
7. At the rise of the 9th clock pulse, the receive data is transferred from ICDRS to ICDR setting the ICDRF flag to 1. The slave device drives SCL low from the fall of the 9th clock pulse until data is read from ICDR.
8. Confirm that the STOP bit is cleared to 0, and clear the IRIC flag to 0.
9. If the next frame is the last receive frame, set the ACKB bit to 1.
10. If ICDR is read, the ICDRF flag is cleared to 0, releasing the SCL bus line. This enab master device to transfer the next data.

    Receive operations can be performed continuously by repeating steps [5] to [10].

11. When the stop condition is detected (SDA is changed from low to high when SCL is the BBSY flag is cleared to 0 and the STOP bit is set to 1. If the STOPIM bit has be cleared to 0, the IRIC flag is set to 1.
12. Confirm that the STOP bit is set to 1, and clear the IRIC flag to 0.

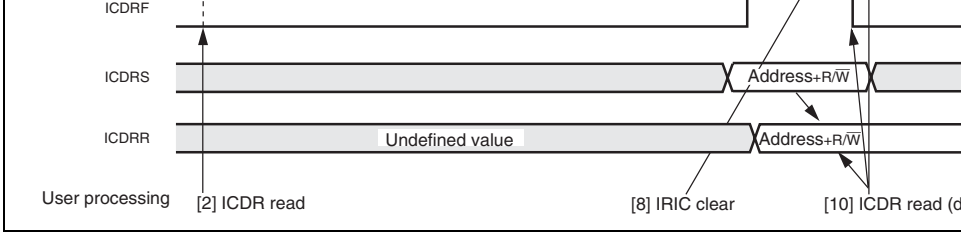


Figure 18.18 Slave Receive Mode Operation Timing Example (1) (MLS = 0, HN

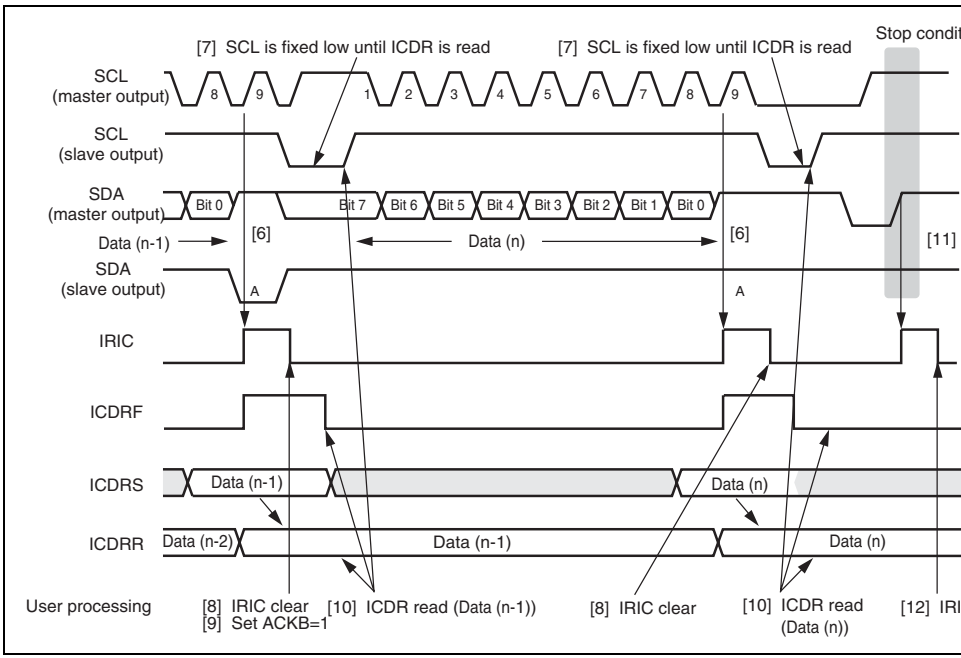


Figure 18.19 Slave Receive Mode Operation Timing Example (2) (MLS = 0, HN

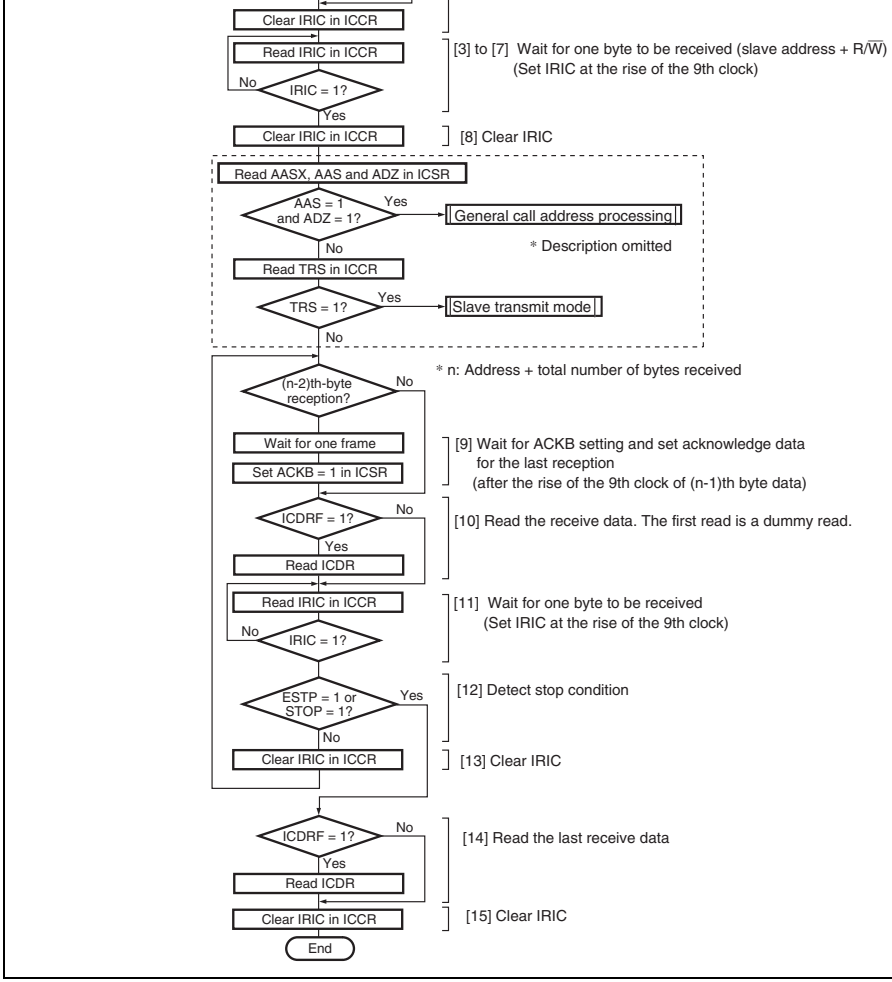


Figure 18.20 Sample Flowchart for Operations in Slave Receive Mode (HNDS

4. When the slave address matches in the first frame following the start condition, the d
operates as the slave device specified by the master device. If the 8th data bit (R/$\overline{W}$)
TRS bit remains cleared to 0, and slave receive operation is performed. If the 8th dat
(R/$\overline{W}$) is 1, the TRS bit is set to 1, and slave transmit operation is performed. When
address does not match, receive operation is halted until the next start condition is de
5. At the 9th clock pulse of the receive frame, the slave device returns the data in the A
as the acknowledge data.
6. At the rise of the 9th clock pulse, the IRIC flag is set to 1. If the IEIC bit has been se
interrupt request is sent to the CPU.

   If the AASX bit has been set to 1, the IRTR flag is also set to 1.
7. At the rise of the 9th clock pulse, the receive data is transferred from ICDRS to ICD
setting the ICDRF flag to 1.
8. Confirm that the STOP bit is cleared to 0 and clear the IRIC flag to 0.
9. If the next read data is the third last receive frame, wait for at least one frame time to
ACKB bit. Set the ACKB bit after the rise of the 9th clock pulse of the second last re
frame.
10. Confirm that the ICDRF flag is set to 1 and read ICDR. This clears the ICDRF flag t
11. At the rise of the 9th clock pulse or when the receive data is transferred from IRDRS
ICDRR due to ICDR read operation, The IRIC and ICDRF flags are set to 1.
12. When the stop condition is detected (SDA is changed from low to high when SCL is
BBSY flag is cleared to 0 and the STOP or ESTP flag is set to 1. If the STOPIM bit
cleared to 0, the IRIC flag is set to 1. In this case, execute step 14 to read the last rec
13. Clear the IRIC flag to 0.

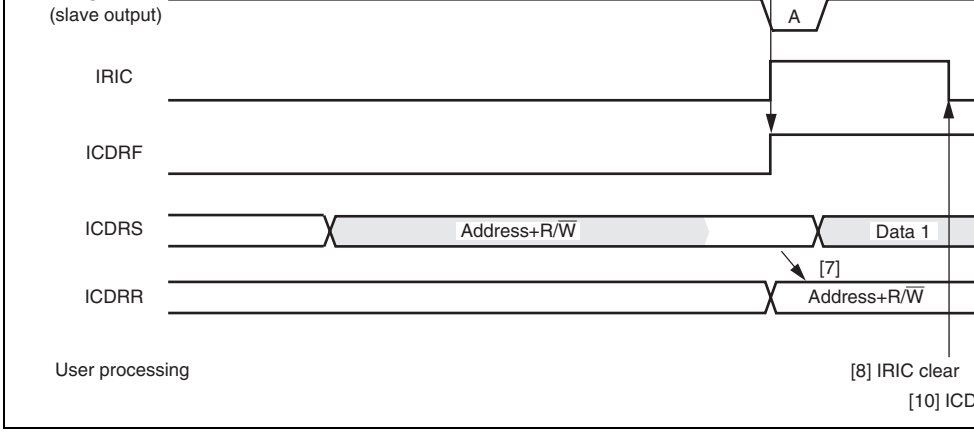


Figure 18.21 Slave Receive Mode Operation Timing Example (1)
(MLS = ACKB = 0, HNDS = 0)

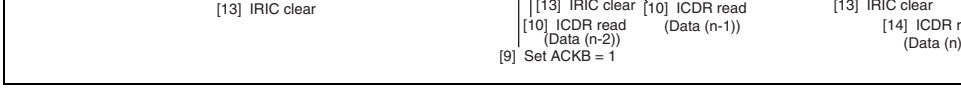


Figure 18.22 Slave Receive Mode Operation Timing Example (2)
(MLS = ACKB = 0, HNDS = 0)

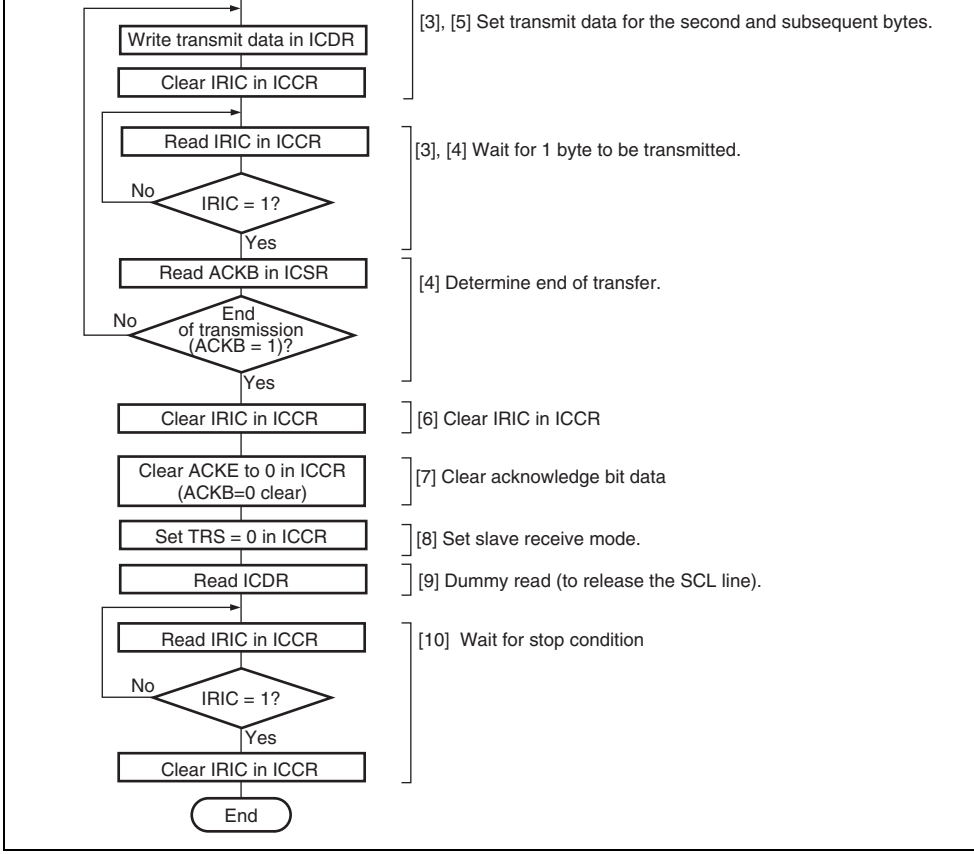


Figure 18.23 Sample Flowchart for Slave Transmit Mode

until ICDR data is written, to disable the master device to output the next transfer clo

3. After clearing the IRIC flag to 0, write data to ICDR. At this time, the ICDRE flag is 0. The written data is transferred to ICDRS, and the ICDRE and IRIC flags are set to The slave device sequentially sends the data written into ICDRS in accordance with output by the master device.

   The IRIC flag is cleared to 0 to detect the end of transmission. Processing from the I register writing to the IRIC flag clearing should be performed continuously. Prevent interrupt processing from being inserted.

4. The master device drives SDA low at the 9th clock pulse, and returns an acknowledg As this acknowledge signal is stored in the ACKB bit in ICSR, this bit can be used to determine whether the transfer operation was performed successfully. When one fram has been transmitted, the IRIC flag in ICCR is set to 1 at the rise of the 9th transmit pulse. When the ICDRE flag is 0, the data written into ICDR is transferred to ICDRS ICDRE and IRIC flags are set to 1 again. If the ICDRE flag has been set to 1, this sla drives SCL low from the fall of the 9th transmit clock until data is written to ICDR.

5. To continue transmission, write the next data to be transmitted into ICDR. The ICDR cleared to 0. The IRIC flag is cleared to 0 to detect the end of transmission. Processi the ICDR register writing to the IRIC flag clearing should be performed continuousl any other interrupt processing from being inserted.

   Transmit operations can be performed continuously by repeating steps 4 and 5.

6. Clear the IRIC flag to 0.
7. To end transmission, clear the ACKE bit in the ICCR register to 0, to clear the ackno bit stored in the ACKB bit to 0.
8. Clear the TRS bit to 0 for the next address reception, to set slave receive mode.
9. Dummy-read ICDR to release SCL on the slave side.

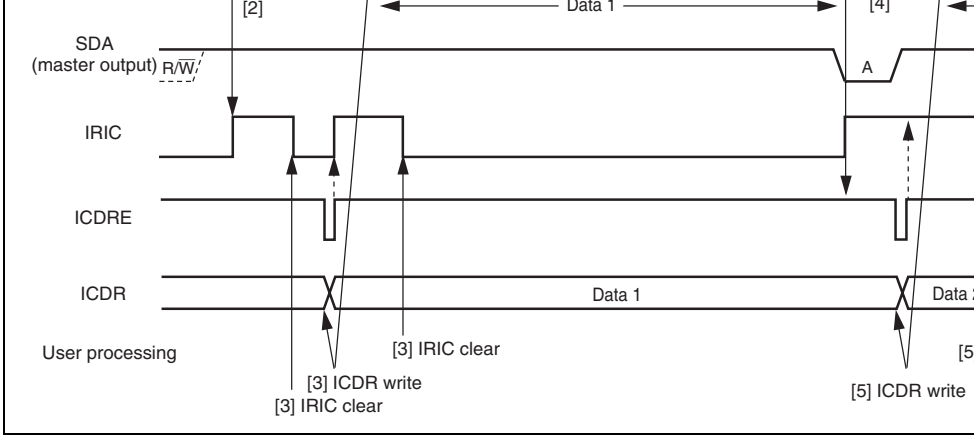
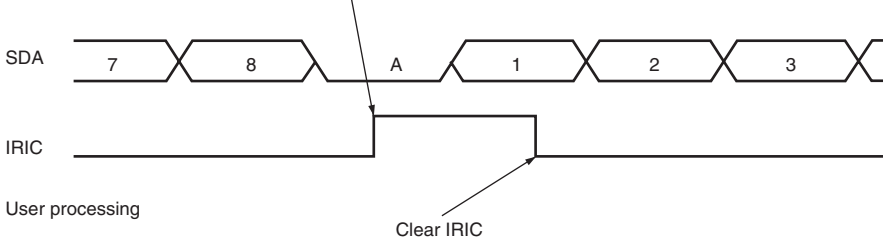
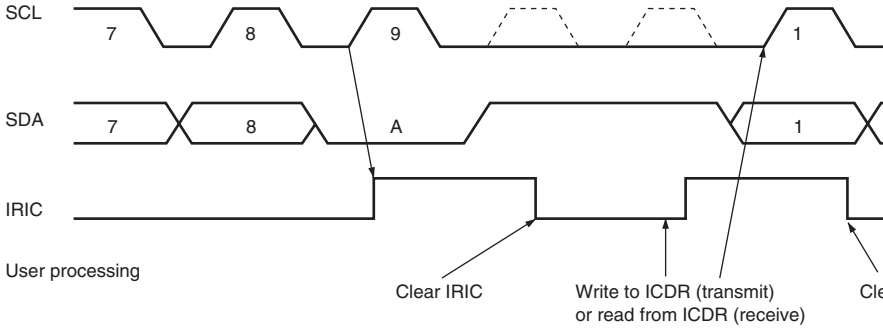


Figure 18.24 Slave Transmit Mode Operation Timing Example (MLS = 0)



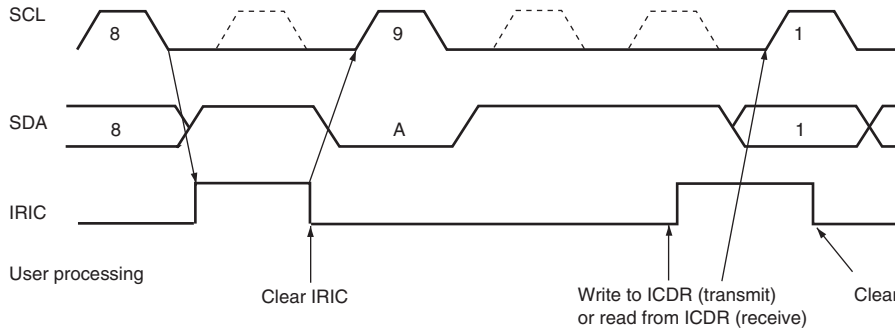
(a) Data transfer ends with ICDRE=0 at transmission, or ICDRF=0 at reception.



(b) Data transfer ends with ICDRE=1 at transmission, or ICDRF=1 at reception.

Figure 18.25 IRIC Setting Timing and SCL Control (1)

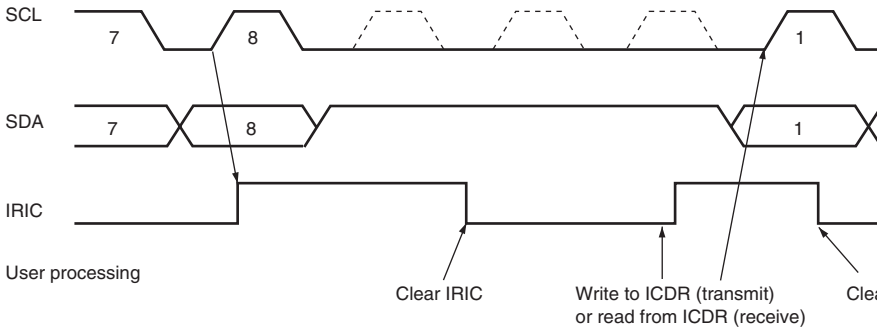
(a) Data transfer ends with ICDRE=0 at transmission, or ICDRF=0 at reception.



(b) Data transfer ends with ICDRE=1 at transmission, or ICDRF=1 at reception.

Figure 18.26 IRIC Setting Timing and SCL Control (2)

(a) Data transfer ends with ICDRE=0 at transmission, or ICDRF=0 at reception.



(b) Data transfer ends with ICDRE=1 at transmission, or ICDRF=1 at reception.

Figure 18.27 IRIC Setting Timing and SCL Control (3)

transmission is completed with the acknowledge bit value of 1 when the ACKE bit is 1, D
not initiated, thus allowing an interrupt to be generated if enabled.

The acknowledge bit may indicate specific events such as completion of receive data proc
for some receiving devices, and for other receiving devices, the acknowledge bit may be l
indicating no specific events.

The I^2C bus format provides for selection of the slave device and transfer direction by me
the slave address and the R/$\overline{W}$ bit, confirmation of reception with the acknowledge bit, in
of the last frame, and so on. Therefore, continuous data transfer using the DTC must be c
in conjunction with CPU processing by means of interrupts.

Table 18.9 shows some examples of processing using the DTC. These examples assume t
number of transfer data bytes is known in slave mode.

transmission/ reception	DTC (ICDRT write)	DTC (ICDRT read)	DTC (ICDRT write)	(ICDRT rea
Dummy data (H'FF) write	—	—	Processing by DTC (ICDR write)	—
Last frame processing	Not necessary	Reception by CPU (ICDR read)	Not necessary	Reception (ICDR rea
Transfer request processing after last frame processing	1st time: Clearing by CPU 2nd time: Stop condition issuance by CPU	Not necessary	Automatic clearing on detection of stop condition during transmission of dummy data (H'FF)	Not neces
Setting of number of DTC transfer data frames	Transmission: Actual data count + 1 (+1 equivalent to slave address + R/$\overline{W}$ bits)	Reception: Actual data count	Transmission: Actual data count + 1 (+1 equivalent to dummy data (H'FF))	Reception data coun

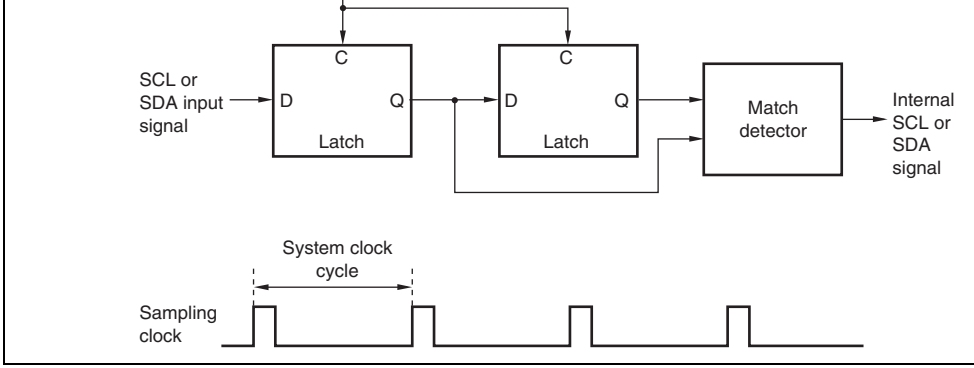


Figure 18.28 Block Diagram of Noise Canceler

18.4.10 Initialization of Internal State

The IIC has a function for forcible initialization of its internal state if a deadlock occurs d
communication.

Initialization is executed in accordance with clearing ICE bit.

Scope of Initialization: The initialization executed by this function covers the following

- ICDRE and ICDRF internal flags
- Transmit/receive sequencer and internal operating clock counter
- Internal latches for retaining the output state of the SCL and SDA pins (wait, clock, d
output, etc.)

- Interrupt flags and interrupt sources are not cleared, and so flag clearing measures m taken as necessary.
- Basically, other register flags are not cleared either, and so flag clearing measures m taken as necessary.
- If a flag clearing setting is made during transmission/reception, the IIC module will s transmitting/receiving at that point and the SCL and SDA pins will be released. Whe transmission/reception is started again, register initialization, etc., must be carried ou necessary to enable correct communication as a system.

The value of the BBSY bit cannot be modified directly by this module clear function, bu stop condition pin waveform is generated according to the state and release timing of the SDA pins, the BBSY bit may be cleared as a result. Similarly, state switching of other b flags may also have an effect.

To prevent problems caused by these factors, the following procedure should be used wh initializing the IIC state.

1. Execute initialization of the internal state according to the ICE bit clearing.
2. Execute a stop condition issuance instruction (write 0 to BBSY and SCP) to clear the bit to 0, and wait for two transfer rate clock cycles.
3. Re-execute initialization of the internal state according to the ICE bit clearing.
4. Initialize (re-set) the IIC registers.

			request		
3	IICI3	IEIC	I^2C bus interface interrupt request	IRIC	Possible
0	IICI0	IEIC	I^2C bus interface interrupt request	IRIC	Possible
1	IICI1	IEIC	I^2C bus interface interrupt request	IRIC	Possible
4	IICI4	IEIC	I^2C bus interface interrupt request	IRIC	Not possible
5	IICI5	IEIC	I^2C bus interface interrupt request	IRIC	Not possible

2. Either of the following two conditions will start the next transfer. Pay attention to the conditions when accessing to ICDR.
   — Write to ICDR when ICE = 1 and TRS = 1 (including automatic transfer from IC ICDRS)
   — Read from ICDR when ICE = 1 and TRS = 0 (including automatic transfer from ICDRR)
3. Table 18.11 shows the timing of SCL and SDA outputs in synchronization with the i clock. Timings on the bus are determined by the rise and fall times of signals affecte bus load capacitance, series resistance, and parallel resistance.

Table 18.11 I^2C Bus Timing (SCL and SDA Outputs)

Item	Symbol	Output Timing	Unit	No
SCL output cycle time	t_{SCLO}	28 t_{cyc} to 512 t_{cyc}	ns	Se
SCL output high pulse width	t_{SCLHO}	0.5 t_{SCLO}	ns	31 (re
SCL output low pulse width	t_{SCLLO}	0.5 t_{SCLO}	ns	
SDA output bus free time	t_{BUFO}	0.5 t_{SCLO} – 1 t_{cyc}	ns	
Start condition output hold time	t_{STAHO}	0.5 t_{SCLO} – 1 t_{cyc}	ns	
Retransmission start condition output setup time	t_{STASO}	1 t_{SCLO}	ns	
Stop condition output setup time	t_{STOSO}	0.5 t_{SCLO} + 2 t_{cyc}	ns	
Data output setup time (master)	t_{SDASO}	1 t_{SCLLO} – 3 t_{cyc}	ns	
Data output setup time (slave)		1 t_{SCLLO} – (6 t_{cyc} or 12 t_{cyc}*)		
Data output hold time	t_{SDAHO}	3 t_{cyc}	ns	

Note: * 6 t_{cyc} when IICXn is 0, 12 t_{cyc} when IICXn is 1 (n = 0 to 5).

Table 18.12 Permissible SCL Rise Time (t_{sr}) Values

				Time Indication [ns]			
TCSS	IICXn	t_{cyc} Indication		I^2C Bus Specification (Max.)	ϕ = 20 MHz	ϕ = 25 MHz	ϕ =
0	0	7.5 t_{cyc}	Standard mode	1000	375	300	22
			High-speed mode	300	300	300	22
	1	17.5 t_{cyc}	Standard mode	1000	875	700	51
1	0		High-speed mode	300	300	300	30
1	1	37.5 t_{cyc}	Standard mode	1000	1000	1000	10
			High-speed mode	300	300	300	30

Note: n = 0 to 5

t_{SCLLO} in high-speed mode and t_{STASO} in standard mode fail to satisfy the I²C bus interfa
specifications for worst-case calculations of t_{Sr}/t_{Sf}. Possible solutions that should be
investigated include (a) adjusting the rise and fall times by means of a pull-up resisto
capacitive load, (b) reducing the transfer rate to meet the specifications, or (c) selecti
whose input timing permits this output timing for use as slave devices connected to t
bus.

		High-speed mode	–250	1300	950*[1]	870*[1]	9
t_{BUFO}	0.5 t_{SCLO} –1 t_{cyc} ($-t_{Sr}$)	Standard mode	–1000	4700	3950*[1]	3440*[1]	3
		High-speed mode	–300	1300	850*[1]	780*[1]	8
t_{STAHO}	0.5 t_{SCLO} –1 t_{cyc} ($-t_{Sr}$)	Standard mode	–250	4000	4700	4190	4
		High-speed mode	–250	600	900	830	8
t_{STASO}	1 t_{SCLO} ($-t_{Sr}$)	Standard mode	–1000	4700	9000	7960	8
		High-speed mode	–300	600	2100	1940	2
t_{STOSO}	0.5 t_{SCLO} + 2 t_{cyc} ($-t_{Sr}$)	Standard mode	–1000	4000	4100	3560	3
		High-speed mode	–300	600	1000	900	9
t_{SDASO} (master)	1 t_{SCLLO}*[3] –3 t_{cyc} ($-t_{Sr}$)	Standard mode	–1000	250	3600	3110	3
		High-speed mode	–300	100	500	450	5
t_{SDASO} (slave)	1 t_{SCLL}*[3] –12 t_{cyc}*[2] ($-t_{Sr}$)	Standard mode	–1000	250	3100	3220	3
		High-speed mode	–300	100	400	520	6
t_{SDAHO}	3 t_{cyc}	Standard mode	0	0	150	120	8
		High-speed mode	0	0	150	120	8

Notes: 1. Does not meet the I^2C bus interface specification. Remedial action such as the is necessary: (a) secure a start/stop condition issuance interval; (b) adjust the fall times by means of a pull-up resistor and capacitive load; (c) reduce the tra (d) select slave devices whose input timing permits this output timing.

The values in the above table will vary depending on the settings of the bits TC IICX3 to IICX0 and CKS2 to CKS0. Depending on the frequency it may not be to achieve the maximum transfer rate; therefore, whether or not the I^2C bus int specifications are met must be determined in accordance with the actual settin conditions.

2. Value when the IICXn bit is set to 1. When the IICXn bit is cleared to 0, the val ($-6t_{cyc}$) (n = 0 to 5).
3. Calculated using the I^2C bus specification values (standard mode: 4700 ns mi speed mode: 1300 ns min.).

Note that if the receive data (ICDR data) is read in the interval between execution of instruction for issuance of the stop condition (writing of 0 to BBSY and SCP in ICCR actual generation of the stop condition, the clock may not be output correctly in subs master transmission.

Clearing of the MST bit after completion of master transmission/reception, or other modifications of IIC control bits to change the transmit/receive operating mode or se must be carried out during interval (a) in figure 18.29 (after confirming that the BBS been cleared to 0 in the ICCR register).

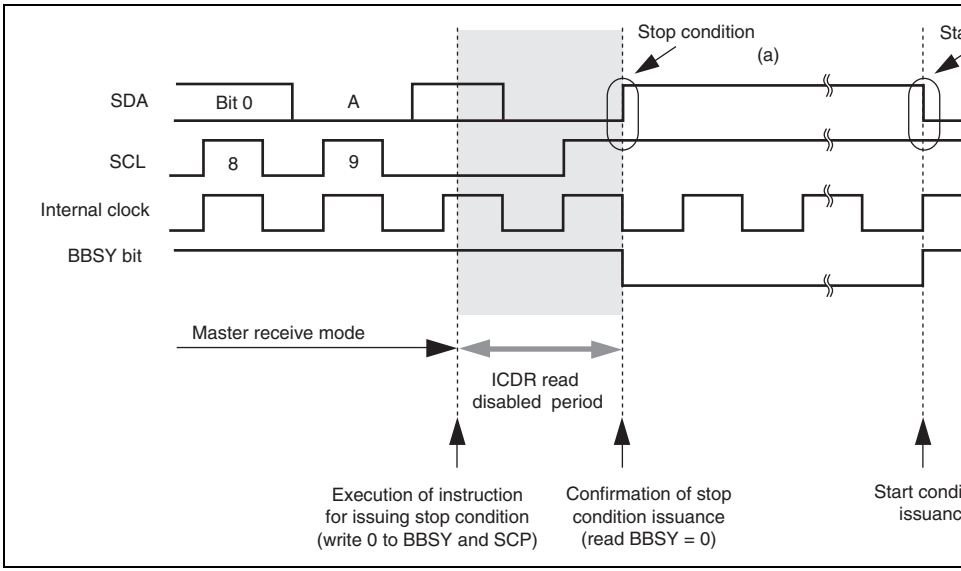


Figure 18.29 Notes on Reading Master Receive Data

Note: This restriction on usage can be canceled by setting the FNC1 and FNC0 bits to ICXR.

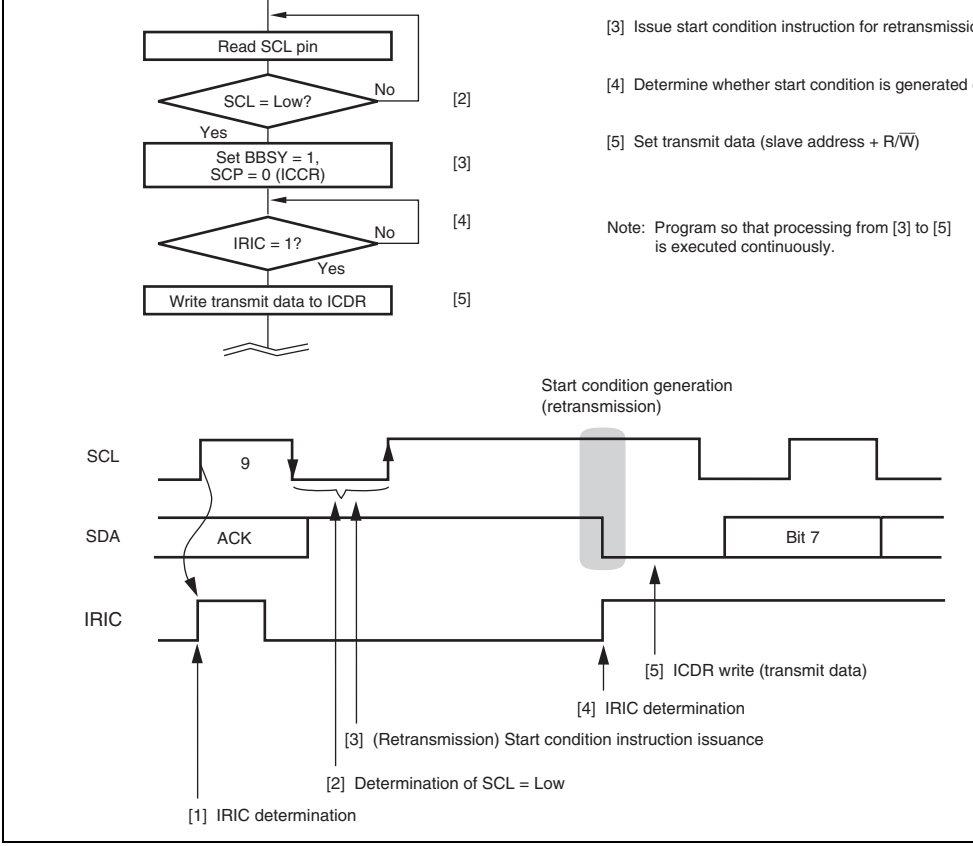


Figure 18.30 Flowchart for Start Condition Issuance Instruction for Retransmission and Timing

Note: This restriction on usage can be canceled by setting the FNC1 and FNC0 bits to B ICXR.

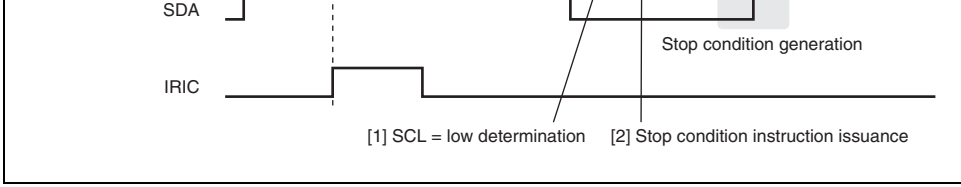


Figure 18.31 Stop Condition Issuance Timing

Note: This restriction on usage can be canceled by setting the FNC1 and FNC0 bits to
ICXR.

10. Note on IRIC flag clear when the wait function is used

When the wait function is used in I^2C bus interface master mode and in a situation w
rise time of SCL exceeds the stipulated value or where a slave device in which a wai
inserted by driving the SCL pin low is used, the IRIC flag should be cleared after de
that the SCL is low.

If the IRIC flag is cleared to 0 when WAIT = 1 while the SCL is extending the high
the SDA level may change before the SCL goes low, which may generate a start or s
condition erroneously.

Figure 18.32 IRIC Flag Clearing Timing When WAIT = 1

Note: This restriction on usage can be canceled by setting the FNC1 and FNC0 bits to B
ICXR.

11. Note on ICDR register read and ICCR register access in slave transmit mode

In I^2C bus interface slave transmit mode, do not read ICDR or do not read/write from/
during the time shaded in figure 18.33. However, such read and write operations sour
problem in interrupt handling processing that is generated in synchronization with the
edge of the 9th clock pulse because the shaded time has passed before making the tra
interrupt handling.

To handle interrupts securely, be sure to keep either of the following conditions.

— Read ICDR data that has been received so far or read/write from/to ICCR before s
the receive operation of the next slave address.

— Monitor the BC2 to BC0 counter in ICMR; when the count is B'000 (8th or 9th clo
pulse), wait for at least two transfer clock times in order to read ICDR or read/wri
ICCR during the time other than the shaded time.

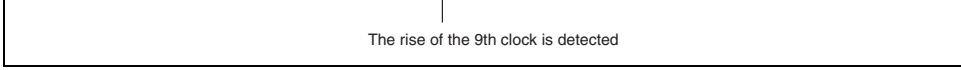


Figure 18.33 ICDR Register Read and ICCR Register Access Timing in Slave Transmit Mode

Note: This restriction on usage can be canceled by setting the FNC1 and FNC0 bits to ICXR.

12. Note on TRS bit setting in slave mode

In I^2C bus interface slave mode, if the TRS bit value in ICCR is set after detecting th edge of the 9th clock pulse or the stop condition before detecting the next rising edg SCL pin (the time indicated as (a) in figure 18.34), the bit value becomes valid imm when it is set. However, if the TRS bit is set during the other time (the time indicate figure 18.34), the bit value is suspended and remains invalid until the rising edge of clock pulse or the stop condition is detected. Therefore, when the address is received restart condition is input without the stop condition, the effective TRS bit value rema (transmit mode) internally and thus the acknowledge bit is not transmitted after the a been received at the 9th clock pulse.

To receive the address in slave mode, clear the TRS bit to 0 during the time indicate figure 18.34. To release the SCL low level that is held by means of the wait function mode, clear the TRS bit to and then dummy-read ICDR.

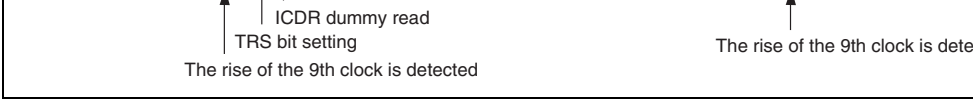


Figure 18.34 TRS Bit Set Timing in Slave Mode

Note: This restriction on usage can be canceled by setting the FNC1 and FNC0 bits to B ICXR.

13. Note on ICDR read in transmit mode and ICDR write in receive mode

When ICDR is read in transmit mode (TRS = 1) or ICDR is written to in receive mod 0), the SCL pin may not be held low in some cases after transmit/receive operation ha completed, thus inconveniently allowing clock pulses to be output on the SCL bus lin ICDR is accessed correctly. To access ICDR correctly, read the ICDR after setting rec mode or write to the ICDR after setting transmit mode.

of a series of transmit operation, clear the ACKE bit in ICCR once to initialize th
bit to 0.

— Set receive mode (TRS = 0) before the next start condition is input in slave mode
Complete transmit operation by the procedure shown in figure 18.23, in order to
from slave transmit mode to slave receive mode.

15. Notes on Arbitration Lost in Master Mode Operation

The I^2C bus interface recognizes the data in transmit/receive frame as an address wh
arbitration is lost in master mode and a transition to slave receive mode is automatic
carried out.

When arbitration is lost not in the first frame but in the second frame or subsequent f
transmit/receive data that is not an address is compared with the value set in the SAR
register as an address. If the receive data matches with the address in the SAR or SA
register, the I^2C bus interface erroneously recognizes that the address call has occurr
figure 18.35.)

In multi-master mode, a bus conflict could happen. When the I^2C bus interface is ope
master mode, check the state of the AL bit in the ICSR register every time after one
data has been transmitted or received.

When arbitration is lost during transmitting the second frame or subsequent frame, ta
avoidance measures.

Figure 18.35 Diagram of Erroneous Operation when Arbitration Lost

Though it is prohibited in the normal I^2C protocol, the same problem may occur when bit is erroneously set to 1 and a transition to master mode is occurred during data trans or reception in slave mode.

When the MST bit is set to 1 during data transmission or reception in slave mode, the arbitration decision circuit is enabled and arbitration is lost if conditions are satisfied. case, the transmit/receive data which is not an address may be erroneously recognized address.

In multi-master mode, pay attention to the setting of the MST bit when a bus conflict occur. In this case, the MST bit in the ICCR register should be set to 1 according to th below.

A. Make sure that the BBSY flag in the ICCR register is 0 and the bus is free before s the MST bit.
B. Set the MST bit to 1.
C. To confirm that the bus was not entered to the busy state while the MST bit is bein check that the BBSY flag in the ICCR register is 0 immediately after the MST bit set.

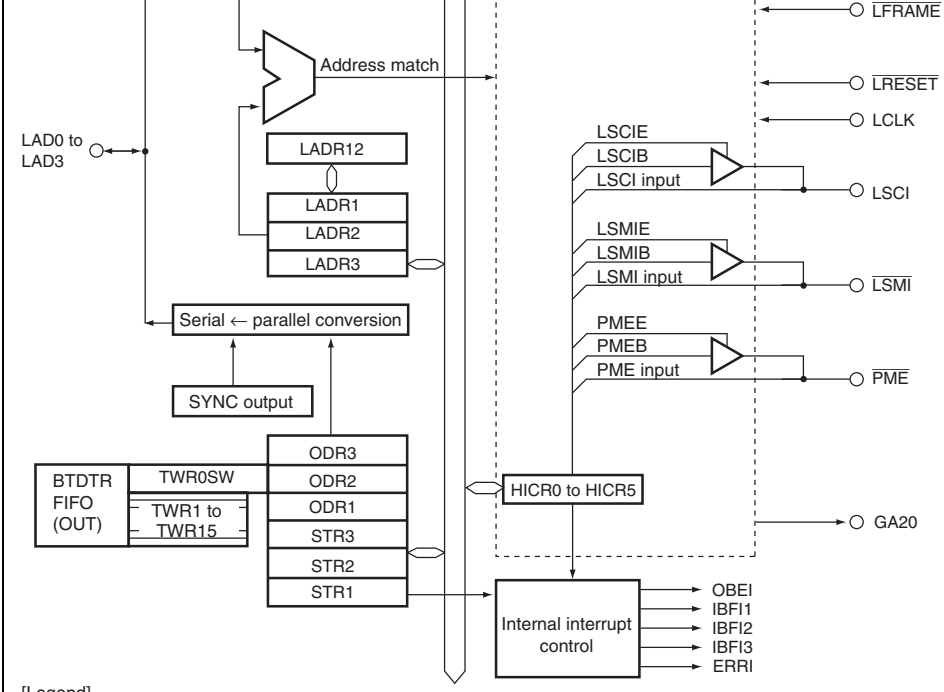
Note: Above restrictions can be released by setting the bits FNC1 and FNC2 in ICXR to

19.1 Features

- Supports LPC interface I/O read and I/O write cycles
  - Uses four signal lines (LAD3 to LAD0) to transfer the cycle type, address, and d
  - Uses three control signals: clock (LCLK), reset ($\overline{\text{LRESET}}$), and frame ($\overline{\text{LFRAME}}$
- Three register sets comprising data and status registers
  - The basic register set comprises three bytes: an input register (IDR), output regis and status register (STR).
  - I/O addresses from H'0000 to H'FFFF are selected for channels 1 to 3.
  - A fast Gate A20 function is provided for channel 1.
  - For channel 3, sixteen bidirectional data register bytes can be manipulated in add the basic register set.
- Supports SCIF
  - The LPC interface is connected to the SCIF, allowing direct control of the SCIF LPC host.

Three pins, PME, ESMI, and ESCI, are provided for general input/output.

- Supports version 1.5 of the Intelligent Platform Management Interface (IPMI) specifi
  - Channel 3 supports the SMIC interface, KCS interface, and BT interface.



[Legend]

HICR0 to HICR5:	Host interface control registers 0 to 5	TWR0MW:	Bidirectional data re
LADR12H, LADR12L:	LPC channel 1, 2 address registers 12H and 12L	TWR0SW:	Bidirectional data re
LADR3H, LADR3L:	LPC channel 3 address registers 3H and 3L	TWR1 to TWR15:	Bidirectional data re
IDR1 to IDR3:	Input data registers 1 to 3	SIRQCR0 to SIRQCR5:	SERIRQ control reg
ODR1 to ODR3:	Output data registers 1 to 3	HISEL:	Host interface selec
STR1 to STR3:	Status registers 1 to 3		

Figure 19.1 Block Diagram of LPC

				termination signal
LPC reset	$\overline{\text{LRESET}}$	PE5	Input*[1]	LPC interface reset signal
LPC clock	LCLK	PE6	Input	33-MHz PCI clock signal
Serialized interrupt request	SERIRQ	PE7	I/O*[1]	Serialized host interrupt re signal (SMI, HIRQ1 to HIR synchronization with LCLK
LSCI general output	LSCI	PD0	Output*[1, 2]	General output
LSMI general output	$\overline{\text{LSMI}}$	PD1	Output*[1, 2]	General output
PME general output	$\overline{\text{PME}}$	PD2	Output*[1, 2]	General output
GATE A20	GA20	PD3	Output*[1, 2]	Gate A20 control signal ou
LPC clock run	$\overline{\text{CLKRUN}}$	PD4	I/O*[1, 2]	LCLK restart request signa serial host interrupt is requ
LPC power-down	$\overline{\text{LPCPD}}$	PD5	Input*[1]	LPC module shutdown sign

Notes: 1. Pin state monitoring input is possible in addition to the LPC interface control input/output function.

2. Only 0 can be output. If 1 is output, the pin is in the high-impedance state, so a external resistor is necessary to pull the signal up to VCC.

- Pin function control register (PINFNCR)
- LPC channel 1, 2 address register H, L (LADR12H, LADR12L)
- LPC channel 3 address register H, L (LADR3H, LADR3L)
- Input data register 1 (IDR1)
- Input data register 2 (IDR2)
- Input data register 3 (IDR3)
- Output data register 1 (ODR1)
- Output data register 2 (ODR2)
- Output data register 3 (ODR3)
- Status register 1 (STR1)
- Status register 2 (STR2)
- Status register 3 (STR3)
- Bidirectional data registers 0 to 15 (TWR0 to TWR15)
- SERIRQ control register 0 (SIRQCR0)
- SERIRQ control register 1 (SIRQCR1)
- SERIRQ control register 2 (SIRQCR2)
- SERIRQ control register 3 (SIRQCR3)
- SERIRQ control register 4 (SIRQCR4)
- SERIRQ control register 5 (SIRQCR5)
- Host interface select register (HISEL)
- SCIF address register H, L (SCIFADRH, SCIFADRL)

- BT control/status register 0 (BTCSR0)
- BT control/status register 1 (BTCSR1)
- BT control register (BTCR)
- BT data buffer (BTDTR)
- BT interrupt mask register (BTIMSR)
- FIFO valid size register 0 (BTFVSR0)
- FIFO valid size register 1 (BTFVSR1)

6	LPC2E	0	R/W	—	Enable or disable the LPC interface function the LPC interface is enabled (one of the thre set to 1), processing for data transfer betwee slave (this LSI) and the host is performed us LAD3 to LAD0, LFRAME, LRESET, LCLK, S CLKRUN, and LPCPD. • LPC3E 0: LPC channel 3 operation is disabled No address (LADR3) matches for IDR3, STR3, TWR0 to TWR15, SMIC, KCS, or 1: LPC channel 3 operation is enabled • LPC2E 0: LPC channel 2 operation is disabled No address (LADR2) matches for IDR2, STR2 1: LPC channel 2 operation is enabled • LPC1E 0: LPC channel 1 operation is disabled No address (LADR1) matches for IDR1, STR1 1: LPC channel 1 operation is enabled
5	LPC1E	0	R/W	—	

					1: Fast Gate A20 function enabled GA20 pin output is open-drain (external pu resistor (Vcc) required)
3	SDWNE	0	R/W	—	LPC Software Shutdown Enable Controls LPC interface shutdown. For details o LPC shutdown function, and the scope of initia by an LPC reset and an LPC shutdown, see s 19.4.6, LPC Interface Shutdown Function (LP 0: Normal state, LPC software shutdown sett enabled [Clearing conditions] • Writing 0 • LPC hardware reset or LPC software reset • LPC hardware shutdown release (rising ed $\overline{\text{LPCPD}}$ signal) 1: LPC hardware shutdown state setting enab Hardware shutdown state when $\overline{\text{LPCPD}}$ si low level [Setting condition] Writing 1 after reading SDWNE = 0

					1 0 : PME output enabled, PME output goes to 0 level 1 1 : PME output enabled, PME output is high-impedance
1	LSMIE	0	R/W	—	LSMI output Enable Controls LSMI output in combination with the bit in HICR1. LSMI pin output is open-drain, a external pull-up resistor (Vcc) is needed. The bit should be cleared to 0 when the LPC is us LSMIE LSMIB 0 X : LSMI output disabled; gen function of pin PD1 is enal 1 0 : LSMI output enabled, LSM output goes to 0 level 1 1 : LSMI output enabled, LSM output is Hi-Z

			output goes to 0 level
1	1	:	LSCI output enabled, LSCI output is high-impedance

[Legend]
X: Don't care

- Cycle type or address indeterminate durin cycle

[Clearing conditions]

- LPC hardware reset or LPC software rese
- LPC hardware shutdown or LPC software shutdown
- Forced termination (abort) of transfer cycl to processing
- Normal termination of transfer cycle subje processing

1: LPC interface is performing transfer cycle processing

[Setting condition]

Match of cycle type and address

					• There are no further interrupts for transfer t host in quiet mode 1: LCLK restart request issued [Setting condition] In quiet mode, SERIRQ interrupt output becom necessary while LCLK is stopped
5	IRQBSY	0	R	—	SERIRQ Busy Indicates that the LPC interface's SERIRQ is e in transfer processing. 0: SERIRQ transfer frame wait state [Clearing conditions] • LPC hardware reset or LPC software reset • LPC hardware shutdown or LPC software s • End of SERIRQ transfer frame 1: SERIRQ transfer processing in progress [Setting condition] Start of SERIRQ transfer frame

					1: LPC software reset state [Setting condition] Writing 1 after reading LRSTB = 0
3	SDWNB	0	R/W	—	LPC Software Shutdown Bit Controls LPC interface shutdown. For details LPC shutdown function, and the scope of init by an LPC reset and an LPC shutdown, see 19.4.6, LPC Interface Shutdown Function (LP 0: Normal state [Clearing conditions] • Writing 0 • LPC hardware reset or LPC software rese • LPC hardware shutdown (falling edge of $\overline{\text{LPCPD}}$ signal when SDW • LPC hardware shutdown release (rising edge of $\overline{\text{LPCPD}}$ signal when SDW 1: LPC software shutdown state [Setting condition] Writing 1 after reading SDWNB = 0
2	PMEB	0	R/W	—	PME Output Bit Controls PME output in combination with the bit. For details, refer to description on the PM HICR0.

Bit	Bit Name	Value	Slave	Host	Description
7	GA20	Undefined	R	—	GA20 Pin Monitor
6	LRST	0	R/(W)*	—	LPC Reset Interrupt Flag This bit is a flag that generates an ERRI int when an LPC hardware reset occurs. 0: [Clearing condition] Writing 0 after reading LRST = 1 1: [Setting condition] $\overline{\text{LRESET}}$ pin falling edge detection
5	SDWN	0	R/(W)*	—	LPC Shutdown Interrupt Flag This bit is a flag that generates an ERRI int when an LPC hardware shutdown request generated. 0: [Clearing conditions] • Writing 0 after reading SDWN = 1 • LPC hardware reset ($\overline{\text{LRESET}}$ pin falling edge detection) • LPC software reset (LRSTB = 1) 1: [Setting condition] $\overline{\text{LPCPD}}$ pin falling edge detection

					• LPC software reset (LRSTB = 1) • LPC hardware shutdown (SDWNE = 1 and $\overline{\text{LPCPD}}$ pin falling edge detection) • LPC software shutdown (SDWNB = 1) 1: [Setting condition] $\overline{\text{LFRAME}}$ pin falling edge detection during LP transfer cycle
3	IBFIE3	0	R/W	—	IDR3 and TWR Receive Complete interrupt E Enables or disables IBFI3 interrupt to the slav LSI). 0: Input data register (IDR3) and TWR recei complete interrupt requests and SMIC/BT interrupt requests disabled 1: [When TWRIE = 0 in LADR3] Input data register (IDR3) receive com interrupt requests and SMIC/BT mode interrupt requests enabled [When TWRIE = 1 in LADR3] Input data register (IDR3) and TWR re complete interrupt requests and SMIC mode interrupt requests enabled

					Enables or disables IBFI1 interrupt to the sl LSI). 0: Input data register (IDR1) receive comple interrupt requests disabled 1: Input data register (IDR1) receive comple interrupt requests enabled
0	ERRIE	0	R/W	—	Error Interrupt Enable Enables or disables ERRI interrupt to the sl LSI). 0: Error interrupt requests disabled 1: Error interrupt requests enabled

Note: * Only 0 can be written to bits 6 to 4, to clear the flag.

- HICR3

			R/W		
Bit	**Bit Name**	**Initial Value**	**Slave**	**Host**	**Description**
7	LFRAME	Undefined	R	—	$\overline{\text{LFRAME}}$ Pin Monitor
6	CLKRUN	Undefined	R	—	$\overline{\text{CLKRUN}}$ Pin Monitor
5	SERIRQ	Undefined	R	—	SERIRQ Pin Monitor
4	LRESET	Undefined	R	—	$\overline{\text{LRESET}}$ Pin Monitor
3	LPCPD	Undefined	R	—	$\overline{\text{LPCPD}}$ Pin Monitor
2	PME	Undefined	R	—	$\overline{\text{PME}}$ Pin Monitor
1	LSMI	Undefined	R	—	$\overline{\text{LSMI}}$ Pin Monitor
0	LSCI	Undefined	R	—	LSCI Pin Monitor

6 to 4	—	All 0	R/W	—	Reserved The initial value should not be changed.
3	SWENBL	0	R/W	—	In BT mode, H'5 (short wait) or H'6 (long returned to the host in the synchronized r cycle from slave, thus can make the host 0: Short wait is issued 1: Long wait is issued
2	KCSENBL	0	R/W	—	Enables or disables the use of the KCS i included in channel 3. When the LPC3E I HICR0 is 0, this bit is valid. 0: KCS interface operation is disabled No address (LADR3) matches for IDR or STR3 in KCS mode 1: KCS interface operation is enabled

					included in channel 3. When the LPCSE HICR0 is 0, this bit is valid. 0: BT interface operation is disabled No address (LADR3) matches for BT BTCR, or BTDTR 1: BT interface operation is enabled

19.3.4 Host Interface Control Register 5 (HICR5)

HICR5 enables or disables the operation of the SCIF interface, and controls OBEI interr

Bit	Bit Name	Initial Value	R/W Slave	R/W Host	Description
7 to 2	—	All 0	R/W	—	Reserved The initial value bit should not be changed
1	SCIFE	0	R/W	—	SCIF Enable Enables or disables access from the LPC the SCIF. 0: Disables access to the SCIF from the L 1: Enables access to the SCIF from the L
0	—	0	R/W	—	Reserved The initial value should not be changed.

1	LPCPDOFF	0	R/W	—	0: LPCPD pin 1: General I/O port
0	CLKRUNOFF	0	R/W	—	0: CLKRUN pin 1: General I/O port

19.3.6 LPC Channel 1, 2 Address Register H, L (LADR12H, LADR12L)

LADR12H and LADR12L are temporary registers for accessing internal registers LADR
LADR1L, LADR2H, and LADR2L.

When the LADR12SEL bit in HICR4 is 0, LPC channel 1 host addresses (LADR1H, LAI
are set through LADR12. The contents of the address field in LADR1 must not be change
channel 1 is operating (while LPC1E is set to 1).

When the LADR12SEL bit is 1, LPC channel 2 host addresses (LADR2H, LADR2L) are
through LADR12. The contents of the address field in LADR2 must not be changed whil
2 is operating (while LPC2E is set to 1).

Table 19.2 shows the initial value of each register. Table 19.3 shows the host register sele
address match determination. Table 19.4 shows the slave selection internal registers in sla
LSI) access.

Table 19.2 LADR1, LADR2 Initial Values

Register Name	Initial Value	Description
LADR1	H'0060	I/O address of channel 1
LADR2	H'0062	I/O address of channel 2

LADR2 (bits 15 to 3)	0	LADR2 (bit 1)	LADR2 (bit 0)	I/O write	IDR2 write (data C/$\overline{D}$2 ← 0
LADR2 (bits 15 to 3)	1	LADR2 (bit 1)	LADR2 (bit 0)	I/O write	IDR2 write (com C/$\overline{D}$2 ← 1
LADR2 (bits 15 to 3)	0	LADR2 (bit 1)	LADR2 (bit 0)	I/O read	ODR2 read
LADR2 (bits 15 to 3)	1	LADR2 (bit 1)	LADR2 (bit 0)	I/O read	STR2 read

Table 19.4 Slave Selection Internal Registers

Slave (R/W)	Bus Width (B/W)	LADR12SEL	LADR12		Internal Re	
R/W	B	0	LADR12H		LADR1H	
R/W	B	1	LADR12H		LADR2H	
R/W	B	0		LADR12L		L
R/W	B	1		LADR12L		L
R/W	W	0	LADR12H	LADR12L	LADR1H	L
R/W	W	1	LADR12H	LADR12L	LADR2H	L

6	Bit 14				The host address of LPC channel 3 is set
5	Bit 13				
4	Bit 12				
3	Bit 11				
2	Bit 10				
1	Bit 9				
0	Bit 8				

- LADR3L

Bit	Bit Name	Initial Value	R/W Slave	R/W Host	Description
7	Bit 7	All 0	R/W	—	Channel 3 Address Bits 7 to 3
6	Bit 6				The host address of LPC channel 3 is set.
5	Bit 5				
4	Bit 4				
3	Bit 3				
2	—	0	R/W	—	Reserved The initial value should not be changed.
1	Bit 1	0	R/W	—	Channel 3 Address Bit 1 The host address of LPC channel 3 is set.
0	TWRE	0	R/W	—	Bidirectional data Register Enable Enables or disables bidirectional data register operation. Clear this bit to 0 in KCS mode. 0: TWR operation is disabled TWR-related address (LADR3) match does occur. 1: TWR operation is enabled

I/O Address						Transfer	Host Regis
Bits 15 to5	**Bit 4**	**Bit 3**	**Bit 2**	**Bit 1**	**Bit 0**	**Cycle**	**Selection**
Bits 15 to5	Bit 4	Bit 3	0	Bit 1	0	I/O write	IDR3 write,
Bits 15 to5	Bit 4	Bit 3	1	Bit 1	0	I/O write	IDR3 write,
Bits 15 to5	Bit 4	Bit 3	0	Bit 1	0	I/O read	ODR3 read
Bits 15 to5	Bit 4	Bit 3	1	Bit 1	0	I/O read	STR3 read
Bits 15 to5	$\overline{\text{Bit 4}}$	0	0	0	0	I/O write	TWR0MW w
Bits 15 to5	$\overline{\text{Bit 4}}$	0	0	0	1	I/O write	TWR1 to TW write
		•	•	•	•		
		•	•	•	•		
		•	•	•	•		
		1	1	1	1		
Bits 15 to5	$\overline{\text{Bit 4}}$	0	0	0	0	I/O read	TWR0SW re
Bits 15 to5	$\overline{\text{Bit 4}}$	0	0	0	1	I/O read	TWR1 to TW read
		•	•	•	•		
		•	•	•	•		
		•	•	•	•		
		1	1	1	1		

I/O Address						Transfer	
Bits 15 to5	**Bit 4**	**Bit 3**	**Bit 2**	**Bit 1**	**Bit 0**	**Cycle**	**Host Register Select**
Bits 15 to5	Bit 4	0	1	0	0	I/O write	BTCR write
Bits 15 to5	Bit 4	0	1	0	1	I/O write	BTDTR write
Bits 15 to5	Bit 4	0	1	1	0	I/O write	BTIMSR write
Bits 15 to5	Bit 4	0	1	0	0	I/O read	BTCR read
Bits 15 to5	Bit 4	0	1	0	1	I/O read	BTDTR read
Bits 15 to5	Bit 4	0	1	1	0	I/O read	BTIMSR read

- SMIC mode

I/O Address						Transfer	
Bits 15 to5	**Bit 4**	**Bit 3**	**Bit 2**	**Bit 1**	**Bit 0**	**Cycle**	**Host Register Select**
Bits 15 to5	Bit 4	1	0	0	1	I/O write	SMICDTR write
Bits 15 to5	Bit 4	1	0	1	0	I/O write	SMICCSR write
Bits 15 to5	Bit 4	1	0	1	1	I/O write	SMICFLG write
Bits 15 to5	Bit 4	1	0	0	1	I/O read	SMICDTR read
Bits 15 to5	Bit 4	1	0	1	0	I/O read	SMICCSR read
Bits 15 to5	Bit 4	1	0	1	1	I/O read	SMICFLG read

The initial values of the IDR registers are undefined.

19.3.9 Output Data Registers 0 to 3 (ODR1 to ODR3)

The ODR registers are 8-bit readable/writable registers to the slave processor (this LSI), read-only registers to the host processor. The registers selected from the host according address are described in the following sections: for information on ODR1 and ODR2 sel section 19.3.6, LPC Channel 1, 2 Address Register H, L (LADR12H, LADR12L), and f information on ODR3 selection, see section 19.3.7, LPC Channel 3 Address Register H, (LADR3H, LADR3L). In an LPC I/O read cycle, the data in the selected register is trans the host.

The initial values of the ODR registers are undefined.

the I/O address, see section 19.5.7, LPC Channel 5 Address Register H, L (LADR5H, LA

Data transferred in an LPC I/O write cycle is written to the selected register; in an LPC I/
cycle, the data in the selected register is transferred to the host.

The initial values of TWR0 to TWR15 are undefined.

Address Register H, L (LADR12H, LADR12L), and information on STR3 selection, se
19.3.7, LPC Channel 3 Address Register H, L (LADR3H, LADR3L). In an LPC I/O rea
the data in the selected register is transferred to the host processor.

The STR registers are initialized to H'00 by a reset or in hardware standby mode.

- STR1

			R/W		
Bit	**Bit Name**	**Initial Value**	**Slave**	**Host**	**Description**
7	DBU17	All 0	R/W	R	Defined by User
6	DBU16				The user can use these bits as necessary.
5	DBU15				
4	DBU14				
3	C/$\overline{D}$1	0	R	R	Command/Data When the host processor writes to an IDR1 bit 2 of the I/O address (when CH1OFFSE bit 0 of the I/O address (when CH1OFFSE written to this bit to indicate whether IDR1 data or a command. 0: Content of input data register (IDR1) is d 1: Content of input data register (IDR1) is a command
2	DBU12	0	R/W	R	Defined by User The user can use this bit as necessary.

					When the slave processor reads IDR 1: There is receive data in IDR1 [Setting condition] When the host processor writes to IDR usin write cycle
0	OBF1	0	R/(W)*	R	Output Data Register Full Indicates whether or not there is transmit da ODR1. 0: There is not transmit data in ODR1 [Clearing condition] When the host processor reads ODR1 usin read cycle, or the slave processor writes 0 t OBF1 bit 1: There is transmit data in ODR1 [Setting condition] When the slave processor writes to ODR1

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

					address (when CH2OFFSEL1 = 0) or bit 0 address (when CH2OFFSEL1 = 1) is writte bit to indicate whether IDR2 contains data command. 0: Content of input data register (IDR2) is a 1: Content of input data register (IDR2) is a command
2	DBU22	0	R/W	R	Defined by User The user can use this bit as necessary.
1	IBF2	0	R	R	Input Data Register Full Indicates whether or not there is receive da IDR2. This bit is an internal interrupt source to the (this LSI). 0: There is not receive data in IDR2 [Clearing condition] When the slave reads IDR2 1: There is receive data in IDR2 [Setting condition] When the host writes to IDR2 in an I/O writ

1: There is transmit data in ODR2

[Setting condition]

- When the slave writes to ODR2

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

					When the host writes to TWR15 in I/O write
6	OBF3B	0	R/(W)*	R	Bidirectional Data Register Output Buffer F 0: [Clearing conditions] • When the host reads TWR15 in I/O rea • When the slave writes 0 to the OBF3B 1: [Setting condition] When the slave writes to TWR15
5	MWMF	0	R	R	Master Write Mode Flag 0: [Clearing condition] When the slave reads TWR15 1: [Setting condition] When the host writes to TWR0 in I/O write while SWMF = 0
4	SWMF	0	R/(W)*	R	Slave Write Mode Flag In the event of simultaneous writes by the and the slave, the master write has priority 0: [Clearing conditions] • When the host reads TWR15 in I/O rea • When the slave writes 0 to the SWMF 1: [Setting condition] When the slave writes to TWR0 while MWI

					The user can use this bit as necessary.
1	IBF3A	0	R	R	Input Data Register Full Indicates whether or not there is receive da IDR3. This is an internal interrupt source to slave (this LSI). 0: There is not receive data in IDR3 [Clearing condition] When the slave reads IDR3 1: There is receive data in IDR3 [Setting condition] When the host writes to IDR3 in an I/O writ
0	OBF3A	0	R/(W)*	R	Output Data Register Full Indicates whether or not there is transmit d ODR3. 0: There is not transmit data in ODR3 [Clearing conditions] • When the host reads ODR3 in a read cycle • When the slave writes 0 to bit O 1: There is transmit data in ODR3 [Setting condition] • When the slave writes to ODR3

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

					address is written into this bit to indicate wh IDR3 contains data or a command. 0: Content of input data register (IDR3) is a 1: Content of input data register (IDR3) is a command
2	DBU32	0	R/W	R	Defined by User The user can use this bit as necessary.
1	IBF3A	0	R	R	Input Data Register Full Indicates whether or not there is receive da IDR3. This bit is an internal interrupt source slave (this LSI). 0: There is not receive data in IDR3 [Clearing condition] When the slave reads IDR3 1: There is receive data in IDR3 [Setting condition] When the host writes to IDR3 in an I/O writ

1: There is receive data in ODR3

[Setting condition]

- When the slave writes to ODR3

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

					0: Continuous mode [Clearing conditions] • LPC hardware reset, LPC software res • Specification by SERIRQ transfer cycle frame 1: Quiet mode [Setting condition] Specification by SERIRQ transfer cycle sto
6	SELREQ	0	R/W	—	Start Frame Initiation Request Select Selects the condition of a start frame initiat request when a host interrupt request is cl quiet mode. 0: Start frame initiation is requested when interrupt requests are cleared 1: Start frame initiation is requested when more interrupt requests are cleared
5	IEDIR2	0	R/W	—	Interrupt Enable Direct Mode Specifies whether LPC channel 2 and char SERIRQ interrupt source (SMI, IRQ6, IRQ IRQ11) generation is conditional upon OBF controlled only by the host interrupt enable 0: Host interrupt is requested when host in enable and corresponding OBF bits are 1 1: Host interrupt is requested when host in enable bit is set to 1

					• Clearing OBF3B to 0 (when IEDIR3 = 0) 1: [When IEDIR3 = 0] Host SMI interrupt request by setting OB is enabled [When IEDIR3 = 1] Host SMI interrupt is requested [Setting condition] Writing 1 after reading SMIE3B = 0
3	SMIE3A	0	R/W	—	Host SMI Interrupt Enable 3A Enables or disables an SMI interrupt reques OBF3A is set by an ODR3 write. 0: Host SMI interrupt request by OBF3A ar SMIE3A is disabled [Clearing conditions] • Writing 0 to SMIE3A • LPC hardware reset, LPC software reset • Clearing OBF3A to 0 (when IEDIR3 = 0) 1: [When IEDIR3 = 0] Host SMI interrupt request by setting is e [When IEDIR3 = 1] Host SMI interrupt is requested [Setting condition] Writing 1 after reading SMIE3A = 0

					• Clearing OBF2 to 0 (when IEDIR2 = 0) 1: [When IEDIR2 = 0] Host SMI interrupt request by setting O is enabled [When IEDIR2 = 1] Host SMI interrupt is requested [Setting condition] Writing 1 after reading SMIE2 = 0
1	IRQ12E1	0	R/W	—	Host IRQ12 Interrupt Enable 1 Enables or disables an HIRQ12 interrupt re when OBF1 is set by an ODR1 write. 0: HIRQ12 interrupt request by OBF1 and is disabled [Clearing conditions] • Writing 0 to IRQ12E1 • LPC hardware reset, LPC software res • Clearing OBF1 to 0 1: HIRQ12 interrupt request by setting OBF enabled [Setting condition] Writing 1 after reading IRQ12E1 = 0

- Clearing OBF1 to 0

1: HIRQ1 interrupt request by setting OBF1 enabled

[Setting condition]

Writing 1 after reading IRQ1E1 = 0

					IRQE11E3 is disabled [Clearing conditions] • Writing 0 to IRQ11E3 • LPC hardware reset, LPC software res • Clearing OBF3A to 0 (when IEDIR3 = 0 1: [When IEDIR3 = 0] HIRQ11 interrupt request by setting OB is enabled [When IEDIR3 = 1] HIRQ11 interrupt is requested [Setting condition] Writing 1 after reading IRQ11E3 = 0
6	IRQ10E3	0	R/W	—	Host IRQ10 Interrupt Enable 3 Enables or disables an HIRQ10 interrupt re when OBF3A is set by an ODR3 write. 0: HIRQ10 interrupt request by OBF3A ar IRQE10E3 is disabled [Clearing conditions] • Writing 0 to IRQ10E3 • LPC hardware reset, LPC software res • Clearing OBF3A to 0 (when IEDIR3 = 0 1: [When IEDIR3 = 0] HIRQ10 interrupt request by setting OB is enabled [When IEDIR3 = 1] HIRQ10 interrupt is requested [Setting condition] Writing 1 after reading IRQ10E3 = 0

					• Clearing OBF3A to 0 (when IEDIR3 = 0) 1: [When IEDIR3 = 0] HIRQ9 interrupt request by setting OBF3 enabled [When IEDIR3 = 1] HIRQ9 interrupt is requested [Setting condition] Writing 1 after reading IRQ9E3 = 0
4	IRQ6E3	0	R/W	—	Host IRQ6 Interrupt Enable 3 Enables or disables an HIRQ6 interrupt requ when OBF3A is set by an ODR3 write. 0: HIRQ6 interrupt request by OBF3A and is disabled [Clearing conditions] • Writing 0 to IRQ6E3 • LPC hardware reset, LPC software reset • Clearing OBF3A to 0 (when IEDIR3 = 0) 1: [When IEDIR3 = 0] HIRQ6 interrupt request by setting OBF3 enabled [When IEDIR3 = 1] HIRQ6 interrupt is requested [Setting condition] Writing 1 after reading IRQ6E3 = 0

					• Clearing OBF2 to 0 (when IEDIR2 = 0) 1: [When IEDIR2 = 0] HIRQ11 interrupt request by setting OB enabled [When IEDIR2 = 1] HIRQ11 interrupt is requested [Setting condition] Writing 1 after reading IRQ11E2 = 0
2	IRQ10E2	0	R/W	—	Host IRQ10 Interrupt Enable 2 Enables or disables an HIRQ10 interrupt re when OBF2 is set by an ODR2 write. 0: HIRQ10 interrupt request by OBF2 and IRQE10E2 is disabled [Clearing conditions] • Writing 0 to IRQ10E2 • LPC hardware reset, LPC software res • Clearing OBF2 to 0 (when IEDIR2 = 0) 1: [When IEDIR2 = 0] HIRQ10 interrupt request by setting OB enabled [When IEDIR2 = 1] HIRQ10 interrupt is requested [Setting condition] Writing 1 after reading IRQ10E2 = 0

					• Clearing OBF2 to 0 (when IEDIR2 = 0) 1: [When IEDIR2 = 0] HIRQ9 interrupt request by setting OBF2 enabled [When IEDIR2 = 1] HIRQ9 interrupt is requested [Setting condition] Writing 1 after reading IRQ9E2 = 0
0	IRQ6E2	0	R/W	—	Host IRQ6 Interrupt Enable 2 Enables or disables an HIRQ6 interrupt requ when OBF2 is set by an oDR2 write. 0: HIRQ6 interrupt request by OBF2 and IF is disabled [Clearing conditions] • Writing 0 to IRQ6E2 • LPC hardware reset, LPC software reset • Clearing OBF2 to 0 (when IEDIR2 = 0) 1: [When IEDIR2 = 0] HIRQ6 interrupt request by setting OBF2 enabled [When IEDIR2 = 1] HIRQ6 interrupt is requested [Setting condition] Writing 1 after reading IRQ6E2 = 0

					enable bit. 0: A host interrupt is generated when bot enable bit and the corresponding OBF set 1: A host interrupt is generated when the is set
6 to 0	—	All 0	R/W	—	Reserved The initial value should not be changed.

1	SCSIRQ1	0	R/W	—	host.
0	SCSIRQ0	0	R/W	—	0000: No interrupt request to the host 0001: HIRQ1 0010: SMI 0011: HIRQ3 0100: HIRQ4 0101: HIRQ5 0110: HIRQ6 0111: HIRQ7 1000: HIRQ8 1001: HIRQ9 1010: HIRQ10 1011: HIRQ11 1100: HIRQ12 1101: HIRQ13 1110: HIRQ14 1111: HIRQ15

					0: Disables HIRQ14 interrupt request by IF
					1: Enables HIRQ14 interrupt request
5	IRQ13E	0	R/W	—	Host IRQ13 Interrupt Enable
					0: Disables HIRQ13 interrupt request by IF
					1: Enables HIRQ13 interrupt request
4	IRQ8E	0	R/W	—	Host IRQ8 Interrupt Enable
					0: Disables HIRQ8 interrupt request by IRQ
					1: Enables HIRQ8 interrupt request
3	IRQ7E	0	R/W	—	Host IRQ7 Interrupt Enable
					0: Disables HIRQ7 interrupt request by IRQ
					1: Enables HIRQ7 interrupt request
2	IRQ5E	0	R/W	—	Host IRQ5 Interrupt Enable
					0: Disables HIRQ5 interrupt request by IRQ
					1: Enables HIRQ5 interrupt request
1	IRQ4E	1	R/W	—	Host IRQ4 Interrupt Enable
					0: Disables HIRQ4 interrupt request by IRQ
					1: Enables HIRQ4 interrupt request
0	IRQ3E	1	R/W	—	Host IRQ3 Interrupt Enable
					0: Disables HIRQ3 interrupt request by IRQ
					1: Enables HIRQ3 interrupt request

3	SELIRQ7	0	R/W	—	0: [When host interrupt request is cleared] SERIRQ pin output is in the Hi-Z state [When host interrupt request is set] SERIRQ pin output is low 1: [When host interrupt request is cleared] SERIRQ pin output is low [When host interrupt request is set] SERIRQ pin output is in the Hi-Z state.
2	SELIRQ5	0	R/W	—	
1	SELIRQ4	0	R/W	—	
0	SELIRQ3	0	R/W	—	

					details of STR3, see section 19.5.11, State Registers 1 to 3 (STR1 to STR3). 0: Bits 7 to 4 in STR3 indicate processing s the LPC interface. 1: [When TWRE = 1] Bits 7 to 4 in STR3 indicate processing s the LPC interface. [When TWRE = 0] Bits 7 to 4 in STR3 are readable/writabl which user can use as necessary
6	SELIRQ11	0	R/W	—	Host IRQ Interrupt Select These bits select the state of the output on SERIRQ pin. 0: [When host interrupt request is cleared] SERIRQ pin output is in the Hi-Z state [When host interrupt request is set] SERIRQ pin output is low 1: [When host interrupt request is cleared] SERIRQ pin output is low [When host interrupt request is set] SERIRQ pin output is in the Hi-Z state.
5	SELIRQ10	0	R/W	—	
4	SELIRQ9	0	R/W	—	
3	SELIRQ6	0	R/W	—	
2	SELSMI	0	R/W	—	
1	SELIRQ12	1	R/W	—	
0	SELIRQ1	1	R/W	—	

6	—	0	R/W	—	These bits set the host address for the SCIF
5	—	0	R/W	—	
4	—	0	R/W	—	
3	—	0	R/W	—	
2	—	0	R/W	—	
1	—	1	R/W	—	
0	—	1	R/W	—	

- SCIFADRL

Bit	Bit Name	Initial Value	R/W		Description
			Slave	Host	
7	—	1	R/W	—	SCIF Address 7 to 0
6	—	1	R/W	—	These bits set the host address for the SCIF
5	—	1	R/W	—	
4	—	1	R/W	—	
3	—	1	R/W	—	
2	—	0	R/W	—	
1	—	0	R/W	—	
0	—	0	R/W	—	

Note: When the SCIF is in use, SCIFADR must be set to an address that is different from for LPC channels 1, 2, and 3.

					the host read transfer. 0: Slave waits for ready status 1: Slave is ready for the host read tra
6	TX_DATA_RDY	0	R/W	R	Write Transfer Ready Indicates whether or not the slave is the host next write transfer. 0: The slave waits for ready status 1: The slave is ready for the host wri transfer.
5	—	0	R/W	R	Reserved The initial value should not be chang
4	SMI	0	R/W	R	SMI Flag This bit indicates that the SMI is asse 0: Indicates waiting for SMI assertion 1: Indicates SMI assertion
3	SEVT_ATN	0	R/W	R	Event Flag When the slave detects an event for this bit is set. 0: Indicates waiting for event detecti 1: Indicates event detection
2	SMS_ATN	0	R/W	R	SMS Flag When there is a message to be trans from the slave to the host, this bit is 0: There is not a message 1: There is a message

	0: Transfer cycle wait state [Clearing conditions] After the slave reads BUSY = 1, writes 0 to 1: Transfer cycle in progress [Setting condition] When the host writes 1 to this bit.

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

19.3.21 SMIC Control Status Register (SMICCSR)

SMICCSR is one of the registers used to implement SMIC mode. This is an 8-bit readable/writable register that stores a control code issued from the host and a status code returned from the slave.

The control code is written to this register accompanied by the transfer between the host a The status code is returned to this register to indicate that the slave has recognized the co code, and a specified transfer cycle has been completed.

19.3.22 SMIC Data Register (SMICDTR)

SMICDTR is one of the registers used to implement SMIC mode. This is an 8-bit register accessible (readable/writable) from both the slave processor (this LSI) and host processor used for data transfer between the host and slave.

					This is a status flag that indicates that the I finished transmitting the transfer data to SM When the IBFIE3 bit and HDTWIE bit are s the IBFI3 interrupt is requested to the slave 0: Transfer data transmission wait state [Clearing condition] After the slave reads HDTWI = 1, writes 0 t 1: Transfer data transmission end [Setting condition] The transfer cycle is write transfer and the writes the transfer data to SMICDTR.
3	HDTRI	0	R/(W)*	—	Transfer Data Receive End Interrupt This is a status flag that indicates that the I finished receiving the transfer data from SM When the IBFIE3 bit and HDTRIE bit are s the IBFI3 interrupt is requested to the slave 0: Transfer data receive wait state [Clearing condition] After the slave reads HDTRI = 1, writes 0 t 1: Transfer data receive end [Setting condition] The transfer cycle is read transfer and the reads the transfer data from SMICDTR.

				1: Status code receive end [Setting condition] When the host reads the status code of SMI
1	CTLWI	0	R/(W)* —	Control Code Transmission End Interrupt This is a status flag that indicates that the h finished transmitting the control code to SMI When the IBFIE3 bit and CTLWIE bit are se IBFI3 interrupt is requested to the slave. 0: Control code transmission wait state [Clearing condition] After the slave reads CTLWI = 1, writes 0 to 1: Control code transmission end [Setting condition] When the host writes the status code to SM
0	BUSYI	0	R/(W)* —	Transfer Start Interrupt This is a status flag that indicates that the h transferring. When the IBFIE3 bit and BUSY are set to 1, the IBFI3 interrupt is requested slave. 0: Transfer start wait state [Clearing condition] After the slave reads BUSYI = 1, writes 0 to 1: Transfer start [Setting condition] When the rising edge of the BUSY bit in SM detected.

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

4	HDTWIE	0	R/W	—	Transfer Data Transmission End Interrupt E Enables or disables HDTWI interrupt that is interrupt source to the slave. 0: Disables transfer data transmission end 1: Enables transfer data transmission end i
3	HDTRIE	0	R/W	—	Transfer Data Receive End Interrupt Enabl Enables or disables HDTRI interrupt that is interrupt source to the slave. 0: Disables transfer data receive end interr 1: Enables transfer data receive end interru
2	STARIE	0	R/W	—	Status Code Receive End Interrupt Enable Enables or disables STARI interrupt that is interrupt source to the slave. 0: Disables status code receive end interru 1: Enables status code receive end interrup
1	CTLWIE	0	R/W	—	Control Code Transmission End Interrupt E Enables or disables CTLWI interrupt that is interrupt source to the slave. 0: Disables control code transmission end 1: Enables control code transmission end i
0	BUSYIE	0	R/W	—	Transfer Start Interrupt Enable Enables or disables BUSYI interrupt that is interrupt source to the slave. 0: Disables transfer start interrupt 1: Enables transfer start interrupt

					This status flag indicates that host writes the BTDTR buffer with FIFO full state at the hos transfer. When the IBFIE3 bit and FRDIE bit to 1, IBFI3 interrupt is requested to the slave slave must clear the flag after creating an u area by reading the data in FIFO. 0: FIFO read is not requested [Clearing condition] After the slave reads FRDI = 1, writes 0 to th 1: FIFO read is requested [Setting condition] After the host processor transfers data, the writes the data with FIFO Full state.
3	HRDI	0	R/(W)*	—	BT Host Read Interrupt This status flag indicates that the host reads from BTDTR buffer. When the IBFIE3 bit an bit are set to 1, IBFI3 interrupt is requested t slave. 0: Host BTDTR read wait state [Clearing condition] After the slave reads HRDI = 1, writes 0 to t 1: The host reads from BTDTR [Setting condition] The host reads one byte from BTDTR.

					1: The host writes to BTDTR [Setting condition] The host writes one byte to BTDTR.
1	HBTWI	0	R/(W)*	—	BTDTR Host Write Start Interrupt This status flag indicates that the host write byte of valid data to BTDTR buffer. When t bit and HBTWIE bit are set to 1, IBFI3 inter requested to the slave. 0: BTDTR host write start wait state [Clearing condition] After the slave reads HBTWI = 1 and writes bit. 1: BTDTR host write start [Setting condition] The host starts writing valid data to BTDTR

		bit. 1: BTDTR host read end [Setting condition] When the host finished reading the valid dat BTDTR.

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

					This status flag indicates that the BMC_HW in BTIMSR is set to 1 by the host. When th bit and HRSTIE bit are set to 1, IBFI3 inter requested to the slave. 0: [Clearing condition] When the slave reads HRSTI = 1 and w this bit. 1: [Setting condition] When the slave detects the rising edge BMC_HWRST.
5	IRQCRI	0	R/(W)*	—	B2H_IRQ Clear Interrupt This status flag indicates that the B2H_IRQ BTIMSR is cleared by the host. When the I and IRQCRIE bit are set to 1, IBFI3 interru requested to the slave. 0: [Clearing condition] When the slave reads IRQCRI = 1 and v this bit. 1: [Setting condition] When the slave detects the falling edge B2H_IRQ.

					1: [Setting condition] When the slave detects the falling edge o BEVT_ATN.
3	B2HI	0	R/(W)*	—	Read End Interrupt This status flag indicates that the host has fi reading all data from the BTDTR buffer. Wh IBFIE3 bit and B2HIE bit are set to 1, the IB interrupt is requested to the slave. 0: [Clearing condition] When the slave reads B2HI = 1 and write this bit. 1: [Setting conditions] When the slave detects the falling edge o B2H_ATN.
2	H2BI	0	R/(W)*	—	Write End Interrupt This status flag indicates that the host has fi writing all data to the BTDTR buffer. When t IBFIE3 bit and H2BIE bit are set to 1, the IB interrupt is requested to the slave. 0: [Clearing condition] After the slave reads H2BI = 1, writes 0 t 1: [Setting condition] When the slave detects the falling edge o H2B_ATN.

					1: [Setting condition] When the slave detects the rising edge o CLR_RD_PTR.
0	CRWPI	0	R/(W)*	—	Write Pointer Clear Interrupt This status flag indicates that the CLR_WR in BTCR is set to 1 by the host. When the I and CRWPIE bit are set to 1, the IBFI3 inte requested to the slave. 0: [Clearing condition] After the slave reads CRWPI = 1, writes bit. 1: [Setting condition] When the slave detects the rising edge o CLR_WR_PTR.

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

6	FSEL1	0	R/W	—	These bits select either FIFO during BT trans
5	FSEL0	0	R/W	—	FSEL1 FSEL0 0 X :FIFO disabled 1 X :FIFO enabled The FIFO size: 64 bytes (for host write transfe additional 64 bytes (for host read transfer).
4	FRDIE	0	R/W	—	FIFO Read Request Interrupt Enable Enables or disables the FRDI interrupt which IBFI3 interrupt source to the slave. 0: FIFO read request interrupt is disabled. 1: FIFO read request interrupt is enabled.
3	HRDIE	0	R/W	—	BT Host Read Interrupt Enable Enables or disables the HRDI interrupt which IBFI3 interrupt source to the slave. When using FIFO, the HRDIE bit must not be 0: BT host read interrupt is disabled. 1: BT host read interrupt is enabled.
2	HWRIE	0	R/W	—	BT Host Write Interrupt Enable Enables or disables the HWRI interrupt which IBFI3 interrupt source to the slave. When using FIFO, the HWRIE bit must not be 1. 0: BT host write interrupt is disabled. 1: BT host write interrupt is enabled.

0: BTDTR host read end interrupt is disabled

1: BTDTR host read end interrupt is enabled

Note: X Don't care.

19.3.28 BT Control Status Register 1 (BTCSR1)

BTCSR1 is one of the registers used to implement the BT mode. The BTCSR1 register c the bits used to enable or disable interrupts to the slave (this LSI). The IBFI3 interrupt is by setting the IBFIE3 bit in HICR2 to 1.

Bit	Bit Name	Initial Value	R/W		Description
			Slave	Host	
7	RSTRENBL	0	R/W	—	Slave Reset Read Enable The host reads 0 from the BMC_HWRST bi BTIMSR. When this bit is set to 1, the host 1 from the BMC_HWRST bit. 0: Host always reads 0 from BMC_HWRST 1: Host can reads 0 from BMC_HWRST
6	HRSTIE	0	R/W	—	BT Reset Interrupt Enable Enables or disables the HRSTI interrupt wh IBFI3 interrupt source to the slave. 0: BT reset interrupt is disabled. 1: BT reset interrupt is enabled.

					0: BEVT_ATN clear interrupt is disabled. 1: BEVT_ATN clear interrupt is enabled.
3	B2HIE	0	R/W	—	Read End Interrupt Enable Enables or disables the B2HI interrupt which IBFI3 interrupt source to the slave. 0: Read end interrupt is disabled. 1: Read end interrupt is enabled.
2	H2BIE	0	R/W	—	Write End Interrupt Enable Enables or disables the H2BI interrupt which IBFI3 interrupt source to the slave. 0: Write end interrupt is disabled. 1: Write end interrupt is enabled.
1	CRRPIE	0	R/W	—	Read Pointer Clear Interrupt Enable Enables or disables the CRRPI interrupt whi IBFI3 interrupt source to the slave. 0: Read pointer clear interrupt is disabled. 1: Read pointer clear interrupt is enabled.
0	CRWPIE	0	R/W	—	Write Pointer Clear Interrupt Enable Enables or disables the CRWPI interrupt wh IBFI3 interrupt source to the slave. 0: Write pointer clear interrupt is disabled. 1: Write pointer clear interrupt is enabled.

					is in progress.) 0: Indicates waiting for BT write transfer 1: Indicates that the BTDTR buffer is being u
6	H_BUSY	0	R	(W)*3	BT Read Transfer Busy Flag This is a set/clear bit from the host. Indicates BTDTR buffer is being used for BT read trans transfer is in progress.) 0: Indicates waiting for BT read transfer [Clearing condition] When the host writes a 1 while H_BUSY is s 1: Indicates that the BTDTR buffer is being u [Setting condition] When the host writes a 1 while H_BUSY is s
5	OEM0	0	R/W	R/(W)*4	User defined bit This bit is defined by the user, and validated set to 1 by a 0 written from the host. 0: [Clearing condition] When the slave writes a 0 after a 1 has be from OEM0. 1: [Setting condition] When the slave writes a 1, after a 0 has be from OEM0, or when the host writes a 0.

					1: An event interrupt request is available [Setting condition] When the slave writes a 1 after a 0 has be from BEVT_ATN.
3	B2H_ATN	0	R/(W)*1	R/(W)*5	Slave Buffer Write End Indication Flag This status flag indicates that the slave ha finished writing all data to the BTDTR buf Setting the B2H_IRQ_EN bit in the BTIMS register enables the B2H_ATN bit to be u an interrupt source to the host. 0: Host has completed reading the BTDT [Clearing condition] When the host writes a 1 1: Slave has completed writing to the BTD buffer [Setting condition] When the slave writes a 1 after a 0 has be from B2N_ATN.
2	H2B_ATN	0	R/(W)*2	R/(W)*1	Host Buffer Write End Indication Flag This status flag indicates that the host has writing all data to the BTDTR buffer. 0: Slave has completed reading the BTDT [Clearing condition] When the slave writes a 0 after a 1 has be from H2B_ATN. 1: Host has completed writing to the BTD [Setting condition] When the host writes a 1

					1: Read pointer clear [Setting condition] When the host writes a 1.
0	CLR_WR_PTR	0	R/(W)*2	(W)*1	Write Pointer Clear This bit is used by the host to clear the writ during write transfer. A host read operation yields 0 on readout. 0: Write pointer clear wait [Clearing condition] When the slave writes a 0 after a 1 has bee from CLR_WR_PTR. 1: Write pointer clear [Setting condition] When the host writes a 1.

Notes: 1. Only 1 can be written to set this flag.
2. Only 0 can be written to clear this flag.
3. Only 1 can be written to toggle this flag.
4. Only 0 can be written to set this flag.
5. Only 1 can be written to clear this flag.

	slave is stored in FIFO (64 bytes) for host re transfer and read out by the host in order of writing.

19.3.31 BT Interrupt Mask Register (BTIMSR)

BTIMSR is one of the registers used to implement BT mode. The BTIMSR register conta
bits used to control the interrupts to the host.

Bit	Bit Name	Initial Value	R/W		Description
			Slave	Host	
7	BMC_ HWRST	0	R/(W)*2	R/(W)*1	Slave Reset Performs a reset from the host to the sla host can only write a 1. Writing a 0 to this invalid. The host will always return a 0 on out. Setting the RSTRENBL bit enables a read from the host. 0: The reset is cancelled [Clearing condition] When the slave writes a 0, after a 1 has read from BMC_HWRST. 1: The reset is in progress. [Setting condition] When the host writes a 1.
6	—	0	R/W	R/W	Reserved
5	—	0	R/W	R/W	

					Read from OEM, or when the host write
1	B2H_IRQ	0	R/(W)*[1]	R/(W)*[3]	BMC to HOST interrupt Informs the host that an interrupt has be requested when the BEVT_ATN or B2H_ has been set. The SERIRQ is not issued generate the SERIRQ, it should be issue program. 0: B2H_IRQ interrupt is not requested [Clearing condition] When the host writes a 1. 1: B2H_IRQ interrupt is requested [Setting condition] When the slave writes a 1, after a 0 has from B2H_IRQ
0	B2H_IRQ_EN	0	R	R/W	BMC to HOST interrupt enable Enables or disables the B2H_IRQ interru is an interrupt source from the slave to th 0: B2H_IRQ interrupt is disabled [Clearing condition] When a 0 is written by the host. 1: B2H_IRQ interrupt is enabled [Setting condition] When a 1 is written by the host.

Notes: 1. Only 1 can be written to set this flag.
2. Only 0 can be written to clear this flag.
3. Only 1 can be written to clear this flag.
4. Only 0 can be written to set this flag.

					by the number of bytes that have been writt Further, when data is read from the slave, t is decremented by only the number of bytes have been read.

19.3.33 BT FIFO Valid Size Register 1 (BTFVSR1)

BTFVSR1 is one of the registers used to implement BT mode. BTFVSR1 indicates a vali
size in the FIFO for host read transfer.

			R/W		
Bit	**Bit Name**	**Initial Value**	**Slave**	**Host**	**Description**
7 to 0	N7 to N0	All 0	R	—	These bits indicate the number of valid byte FIFO (the number of bytes which the host c for host read transfer. When data is written slave, the value in BTFVSR1 is incremente number of bytes that have been written to. I when data is read from the host, the value i decremented by only the number of bytes th been read.

Use the following procedure to activate the LPC interface after a reset release.

1. Read the signal line status and confirm that the LPC module can be connected. Also the LPC module is initialized internally.
2. When using channels 1 and 2, set LADR1 and LADR2 to determine the I/O address.
3. When using channel 3, set LADR3 to determine the I/O address and whether bidirec registers are to be used.
4. When using the SCIF module, set SCIFAR to determine the I/O address.
5. Set the enable bit (LPC3E to LPC1E) for the channel to be used. Also set SCIFE if t to be used.
6. Set the enable bits (FGA20E, PMEE, LSMIE, and LSCIE) for the additional functio used.
7. Set the selection bits for other functions (SDWNE, IEDIR).
8. As a precaution, clear the interrupt flags (LRST, SDWN, ABRT, OBF, and OBEI). I or TWR15 to clear IBF.
9. Set receive complete interrupt enable bits (IBFIE3 to IBFIE1, and ERRIE) as necess

19.4.2 LPC I/O Cycles

There are 12 types of LPC transfer cycle: LPC memory read, LPC memory write, I/O re write, DMA read, DMA write, bus master memory read, bus master memory write, bus read, bus master I/O write, FW memory read, and FW memory write. Of these, the LPC LSI supports I/O read and I/O write.

changes are made at this timing, so in the event of a transfer cycle forced termination (abo
registers and flags are not changed.

The timing of the $\overline{\text{LFRAME}}$, LCLK, and LAD signals is shown in figures 19.2 and 19.3.

Table 19.5 LPC I/O Cycle

	I/O Read Cycle			I/O Write Cycle		
State Count	**Contents**	**Drive Source**	**Value (3 to 0)**	**Contents**	**Drive Source**	**Va (3**
1	Start	Host	0000	Start	Host	00
2	Cycle type/direction	Host	0000	Cycle type/direction	Host	00
3	Address 1	Host	Bits 15 to 12	Address 1	Host	Bit
4	Address 2	Host	Bits 11 to 8	Address 2	Host	Bit
5	Address 3	Host	Bits 7 to 4	Address 3	Host	Bit
6	Address 4	Host	Bits 3 to 0	Address 4	Host	Bit
7	Turnaround (recovery)	Host	1111	Data 1	Host	Bit
8	Turnaround	None	ZZZZ	Data 2	Host	Bit
9	Synchronization	Slave	0000	Turnaround (recovery)	Host	11
10	Data 1	Slave	Bits 3 to 0	Turnaround	None	ZZ
11	Data 2	Slave	Bits 7 to 4	Synchronization	Slave	00
12	Turnaround (recovery)	Slave	1111	Turnaround (recovery)	Slave	11
13	Turnaround	None	ZZZZ	Turnaround	None	ZZ

Figure 19.2 Typical LFRAME Timing

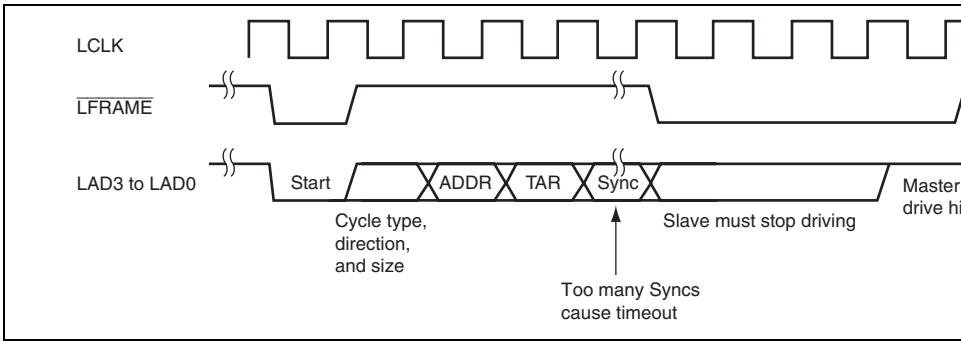


Figure 19.3 Abort Mechanism

19.4.3 SMIC Mode Transfer Flow

Figure 19.4 shows the write transfer flow and figure 19.5 shows the read transfer flow in
mode.

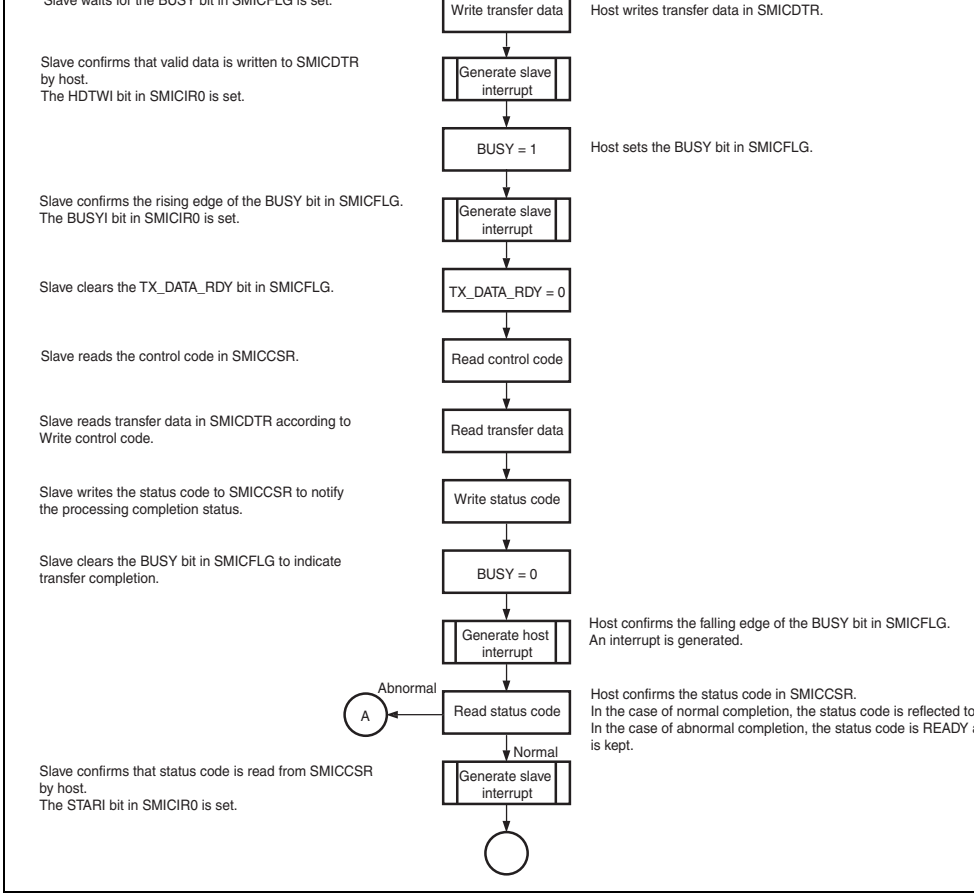


Figure 19.4 SMIC Write Transfer Flow

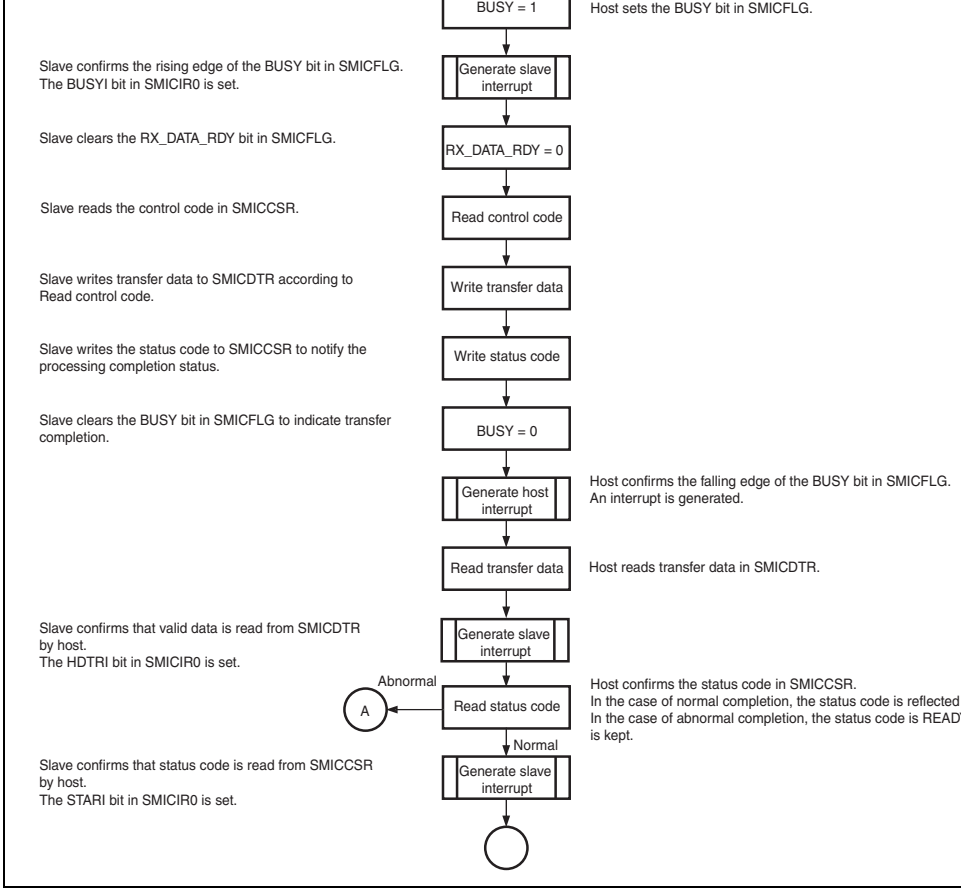


Figure 19.5 SMIC Read Transfer Flow

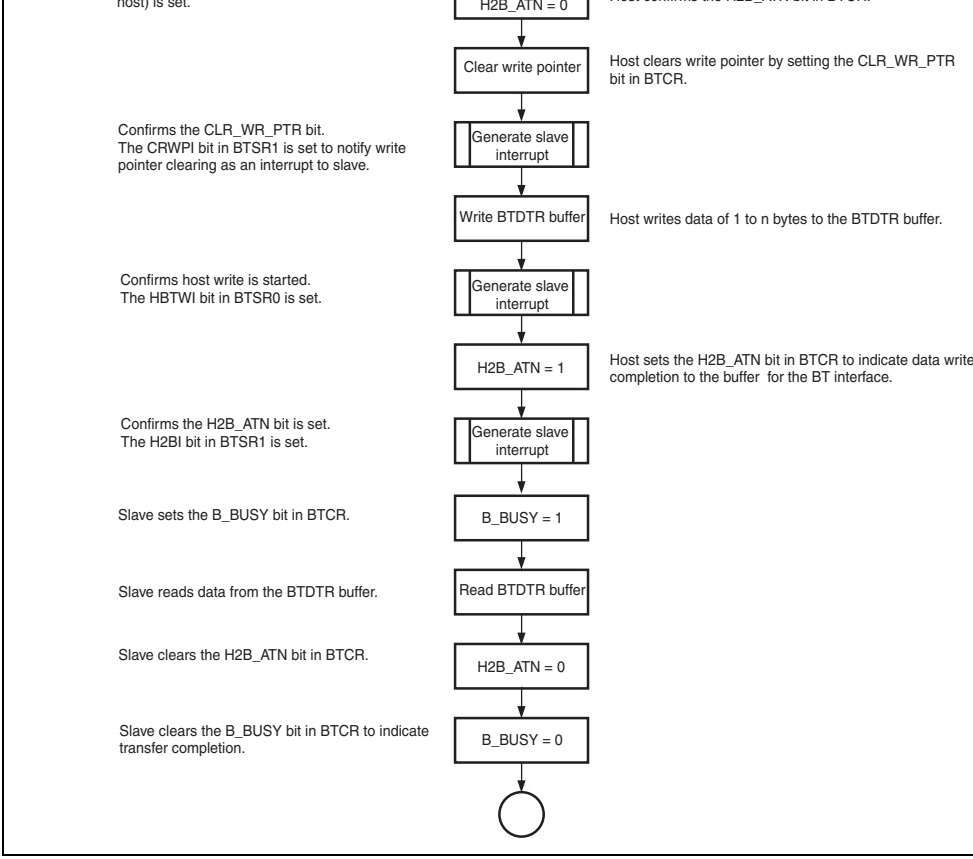


Figure 19.6 BT Write Transfer Flow

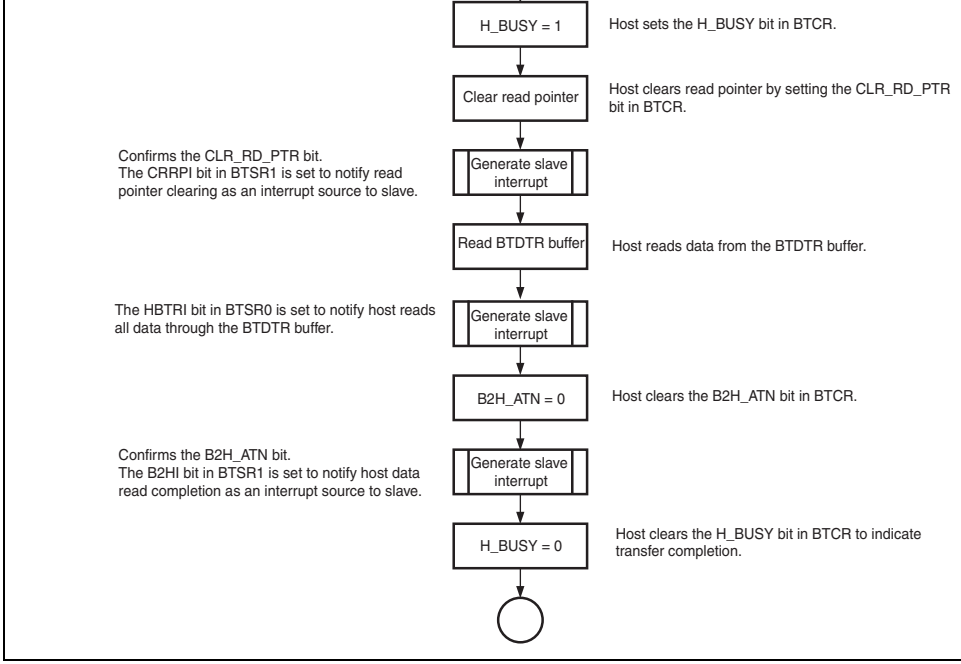


Figure 19.7 BT Read Transfer Flow

Output of the Gate A20 signal can be controlled by an H'D1 command and data. When th
(this LSI) receives data, it normally reads IDR1 in the interrupt handling routine activated
IBFI1 interrupt. At this time, firmware copies bit 1 of data following an H'D1 command a
outputs it on pin GA20.

(2) Fast Gate A20 Operation

The internal state of pin GA20 is initialized to 1 since the initial value of the FGA20E bit
When the FGA20E bit is set to 1, pin P81/GA20 functions as the output of the fast GA20
The state of pin GA20 can be monitored by reading bit GA20 in HICR2.

The initial output from this pin is 1, which is the initial value. Afterward, the host can ma
the output from this pin by sending commands and data. This function is only available v
IDR1. The LPC decodes commands input from the host. When an H'D1 host command is
detected, bit 1 of the data following the host command is output from pin GA20. This ope
does not depend on firmware or interrupts, and is faster than the regular processing using
interrupts. Table 19.6 shows the conditions that set and clear pin GA20. Figure 19.8 show
GA20 output flow. Table 19.7 indicates the GA20 output signal values.

Table 19.6 GA20 Setting/Clearing Timing

Pin Name	Setting Condition	Clearing Condition
GA20	When bit 1 of the data that follows an H'D1 host command is 1	When bit 1 of the data that follo H'D1 host command is 0

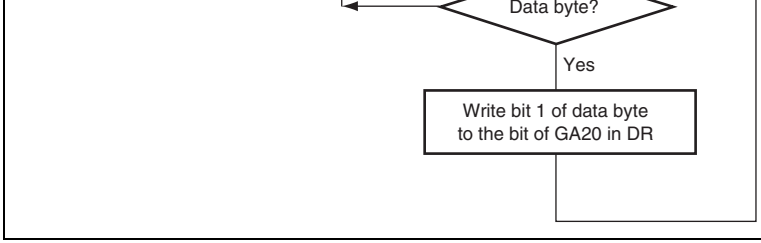


Figure 19.8 GA20 Output

1	H'FF command	0	Q (0)	
1	H'D1 command	0	Q	Turn-on sequen
0	1 data*[1]	0	1	(abbreviated for
1/0	Command other than H'FF and H'D1	1	Q (1)	
1	H'D1 command	0	Q	Turn-off sequen
0	0 data*[2]	0	0	(abbreviated for
1/0	Command other than H'FF and H'D1	1	Q (0)	
1	H'D1 command	0	Q	Cancelled sequ
1	Command other than H'D1	1	Q	
1	H'D1 command	0	Q	Retriggered seq
1	H'D1 command	0	Q	
1	H'D1 command	0	Q	Consecutively e
0	Any data	0	1/0	sequences
1	H'D1 command	0	Q (1/0)	

Notes: 1. Any data with bit 1 set to 1.
2. Any data with bit 1 cleared to 0.

for exiting software standby mode before clearing the shutdown state with the $\overline{\text{LPCPD}}$ s

If the SDWNE bit has been set to 1 beforehand, the LPC hardware shutdown state is ent same time as the $\overline{\text{LPCPD}}$ signal falls, and prior preparation is not possible. If the LPC so shutdown state is set by means of the SDWNB bit, on the other hand, the LPC software state cannot be cleared at the same time as the rising edge of the $\overline{\text{LPCPD}}$ signal. Taking points into consideration, the following operating procedure uses a combination of LPC shutdown and LPC hardware shutdown.

1. Clear the SDWNE bit to 0.
2. Set the ERRIE bit to 1 and wait for an interrupt by the SDWN flag.
3. When an ERRI interrupt is generated by the SDWN flag, check the LPC interface in status flags and perform any necessary processing.
4. Set the SDWNB bit to 1 to set LPC software standby mode.
5. Set the SDWNE bit to 1 and make a transition to LPC hardware standby mode. The bit is cleared automatically.
6. Check the state of the $\overline{\text{LPCPD}}$ signal to make sure that the $\overline{\text{LPCPD}}$ signal has not ris steps 3 to 5. If the signal has risen, clear SDWNE to 0 to return to the state in step 1.
7. If software standby mode has been set, exit software standby mode by some means independent of the LPC.
8. When a rising edge is detected in the $\overline{\text{LPCPD}}$ signal, the SDWNE bit is automaticall to 0. If the slave has been placed in sleep mode, the mode is exited by means of $\overline{\text{LRE}}$ signal input, on completion of the LPC transfer cycle, or by some other means.

LSCI	PD0	Δ	I/O	Hi-Z, only when LSCIE = 1
$\overline{\text{LSMI}}$	PD1	Δ	I/O	Hi-Z, only when LSMIE = 1
$\overline{\text{PME}}$	PD2	Δ	I/O	Hi-Z, only when PMEE = 1
GA20	PD3	Δ	I/O	Hi-Z, only when FGA20E = 1
$\overline{\text{CLKRUN}}$	PD4	O	Input	Hi-Z
$\overline{\text{LPCPD}}$	PD5	X	Input	Needed to clear shutdown state

[Legend]

O: Pin that is shutdown by the shutdown function

Δ: Pin that is shutdown only when the LPC function is selected by register setting

X: Pin that is not shutdown

In the LPC shutdown state, the LPC's internal state and some register bits are initialized. T of priority of LPC shutdown and reset states is as follows.

1. System reset (reset by $\overline{\text{RES}}$ pin input, or WDT0 overflow)
   All register bits, including bits LPC4E to LPC1E, are initialized.
2. LPC hardware reset (reset by $\overline{\text{LRESET}}$ pin input)
   LRSTB, SDWNE, and SDWNB bits are cleared to 0.
3. LPC software reset (reset by LRSTB)
   SDWNE and SDWNB bits are cleared to 0.
4. LPC hardware shutdown
   SDWNB bit is cleared to 0.
5. LPC software shutdown

The scope of the initialization in each mode is shown in table 19.9.

BTCSR1, BTMSR, BTTCSR, BTTCNT), SMIE (internal state)			
Host interrupt enable bits (IRQ1E1, IRQ12E1, SMIE2, IRQ6E2, IRQ9E2 to IRQ11E2, SMIE3B, SMIE3A, IRQ6E3, IRQ9E3 to IRQ11E3, SELREQ, IEDIR2 to IEDIR3), Q/$\overline{C}$ flag	Initialized	Initialized	R
LRST flag	Initialized (0)	Can be set/cleared	C s
SDWN flag	Initialized (0)	Initialized (0)	C s
LRSTB bit	Initialized (0)	HR: 0 SR: 1	C s
SDWNB bit	Initialized (0)	Initialized (0)	H S
SDWNE bit	Initialized (0)	Initialized (0)	H S
LPC interface operation control bits (LPC3E to LPC1E, FGA20E, LADR1 to LADR3, IBFIE1 to IBFIE3, PMEE, PMEB, LSMIE, LSMIB, LSCIE, LSCIB, TWRE, SELSTR3, SELIRQ1, SELSMI, SELIRQ3 to SELIRQ15, HICR4, HICR5, SCIFAR, HISEL, BTCSR0, BTCSR1)	Initialized	Retained	R
$\overline{\text{LRESET}}$ signal	Input (port function	Input	I
$\overline{\text{LPCPD}}$ signal		Input	I
LAD3 to LAD0, $\overline{\text{LFRAME}}$, LCLK, SERIRQ, $\overline{\text{CLKRUN}}$ signals		Input	H
$\overline{\text{PME}}$, $\overline{\text{LSMI}}$, LSCI, GA20 signals (when function is selected)		Output	H
$\overline{\text{PME}}$, $\overline{\text{LSMI}}$, LSCI, GA20 signals (when function is not selected)		Port function	P

Note: System reset: Reset by STBY input, RES input, or WDT overflow
LPC reset: Reset by LPC hardware reset (HR) or LPC software reset (SR)
LPC shutdown: Reset by LPC hardware shutdown (HS) or LPC software shutdow

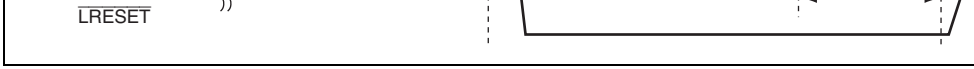


Figure 19.9 Power-Down State Termination Timing

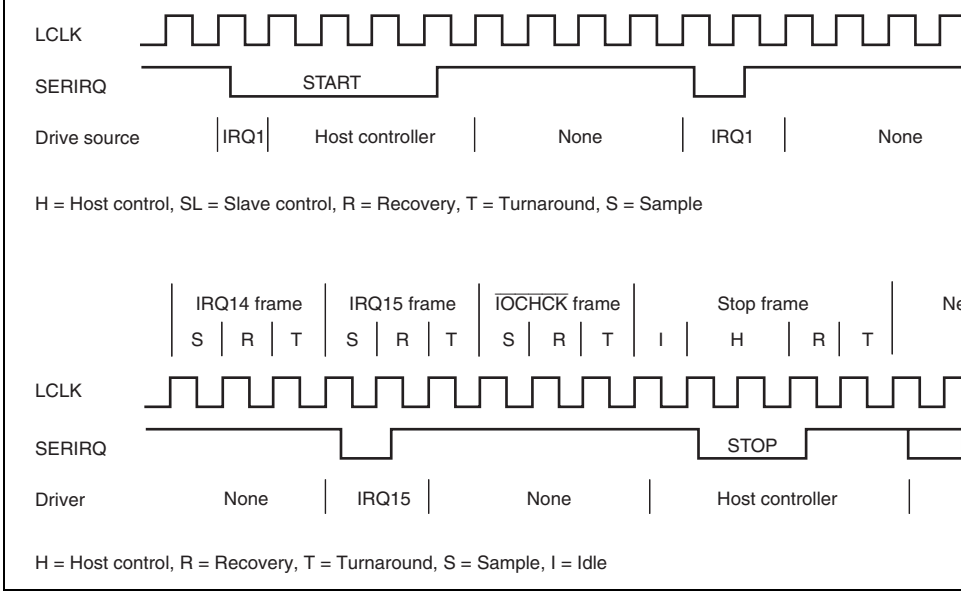


Figure 19.10 SERIRQ Timing

		Host		first state, then next 3 states 0-driven
1	IRQ0	Slave	3	Drive impossible
2	IRQ1	Slave	3	Drive possible in LPC channel 1 and S
3	SMI	Slave	3	Drive possible in LPC channels 2, 3, a
4	IRQ3	Slave	3	Drive possible in SCIF or by IRQ3E
5	IRQ4	Slave	3	Drive possible in SCIF or by IRQ4E
6	IRQ5	Slave	3	Drive possible in SCIF or by IRQ5E
7	IRQ6	Slave	3	Drive possible in LPC channels 2, 3, a
8	IRQ7	Slave	3	Drive possible in SCIF or by IRQ7E
9	IRQ8	Slave	3	Drive possible in SCIF or by IRQ8E
10	IRQ9	Slave	3	Drive possible in LPC channels 2, 3, a
11	IRQ10	Slave	3	Drive possible in LPC channels 2, 3, a
12	IRQ11	Slave	3	Drive possible in LPC channels 2, 3, a
13	IRQ12	Slave	3	Drive possible in LPC channel 1 and S
14	IRQ13	Slave	3	Drive possible in SCIF or by IRQ13E
15	IRQ14	Slave	3	Drive possible in SCIF or by IRQ14E
16	IRQ15	Slave	3	Drive possible in SCIF or by IRQ15E
17	IOCHCK	Slave	3	Drive impossible
18	Stop	Host	Undefined	First, 1 or more idle states, then 2 or 3 0-driven by host 2 states: Quiet mode next 3 states: Continuous mode next

19.4.8 LPC Interface Clock Start Request

A request to restart the clock (LCLK) can be sent to the host by means of the $\overline{\text{CLKRUN}}$
LPC data transfer and SERIRQ in continuous mode, a clock restart is never requested si
transfer cycles are initiated by the host. With SERIRQ in quiet mode, when a host interr
request is generated the $\overline{\text{CLKRUN}}$ signal is driven and a clock (LCLK) restart request is
the host. The timing for this operation is shown in figure 19.11.

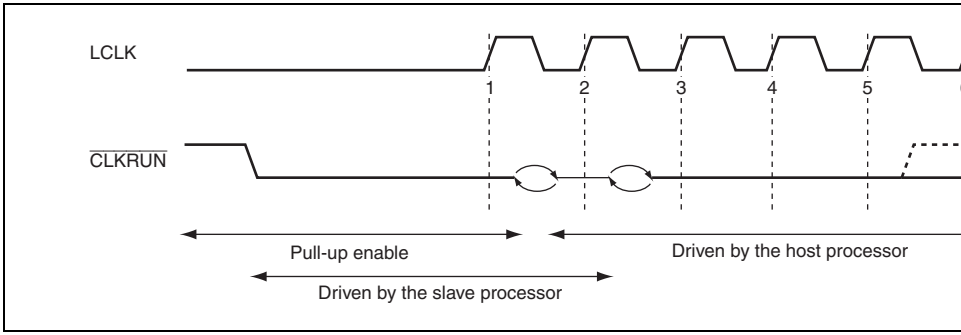


Figure 19.11 Clock Start Request Timing

Cases other than SERIRQ in quiet mode when clock restart is required must be handled
different protocol, using the $\overline{\text{PME}}$ signal, etc.

19.4.9 SCIF Control from LPC Interface

Setting the SCIFE bit in HICR5 to 1 allows the LPC host to communicate with the SCIF
the LPC interface can access the registers of the module SCIF other than SCIFCR. For d
transmission and reception, see section 15, Serial Communication Interface with FIFO (

Interrupt	Description
IBFI1	When IBFIE1 is set to 1 and IDR1 reception is completed
IBFI2	When IBFIE2 is set to 1 and IDR2 reception is completed
IBFI3	When IBFIE3 is set to 1 and IDR3 reception is completed, or when TWR IBFIE3 are set to 1 and reception is completed up to TWR15
ERRI	When ERRIE is set to 1 and one of LRST, SDWN and ABRT is set to 1

When the IEDIR bit in SIRQCR0is cleared to 0, host interrupt sources and LPC channel linked to the host interrupt request enable bits. When the OBF flag is cleared to 0 by a re ODR or TWR15 by the host in the corresponding LPC channel, the corresponding host i enable bit is automatically cleared to 0, and the host interrupt request is cleared.

When the IEDIR bit is set to 1 in SIRQCR, a host interrupt is only requested by the host enable bits. The host interrupt enable bit is not cleared when OBF is cleared. Therefore, SMIE3A, SMIE3B, SMIE4 and IRQ6En, IRQ9En, IRQ10En, IRQ11En lose their respe functional differences (n = 2, 3). In order to clear a host interrupt request, it is necessary the host interrupt enable bit. As for HIRQ3 to HIRQ5, HIRQ7, HIRQ8, and HIRQ13 to setting the enable bit in SIRQCR4 to 1 requests the corresponding host interrupt, and cle enable bit to 0 clears the corresponding host interrupt request.

When the SCIF channels are used, a host interrupt request is cleared when the relevant S interrupt is cleared.

Table 19.12 summarizes the methods of setting and clearing these bits when the LPC ch used, and table 19.13 summarizes the methods of setting and clearing these bits when th channels are used. Figure 19.12 shows the processing flowchart.

	SMIE3A and writes 1 • writes to TWR15, then reads 0 from bit SMIE3B and writes 1	reads ODR3 • writes 0 to bit SMIE3B, reads TWR15
SMI (IEDIR2 = 1 or IEDIR3 = 1)	Internal CPU • reads 0 from bit SMIE2, then writes 1 • reads 0 from bit SMIE3A, then writes 1 • reads 0 from bit SMIE3B, then writes 1	Internal CPU • writes 0 to bit SMIE2 • writes 0 to bit SMIE3A • writes 0 to bit SMIE3B
HIRQi (i = 6, 9, 10, 11) (IEDIR2 = 0 or IEDIR3 = 0)	Internal CPU • writes to ODR2, then reads 0 from bit IRQiE2 and writes 1 • writes to ODR3, then reads 0 from bit IRQiE3 and writes 1	Internal CPU • writes 0 to bit IRQiE2, o reads ODR2 • writes 0 to bit IRQiE3, o reads ODR3
HIRQi (i = 6, 9, 10, 11) (IEDIR2 = 1 or IEDIR3 = 1)	Internal CPU • reads 0 from bit IRQiE2, then writes 1 • reads 0 from bit IRQiE3, then writes 1	Internal CPU • writes 0 to bit IRQiE2 • writes 0 to bit IRQiE3

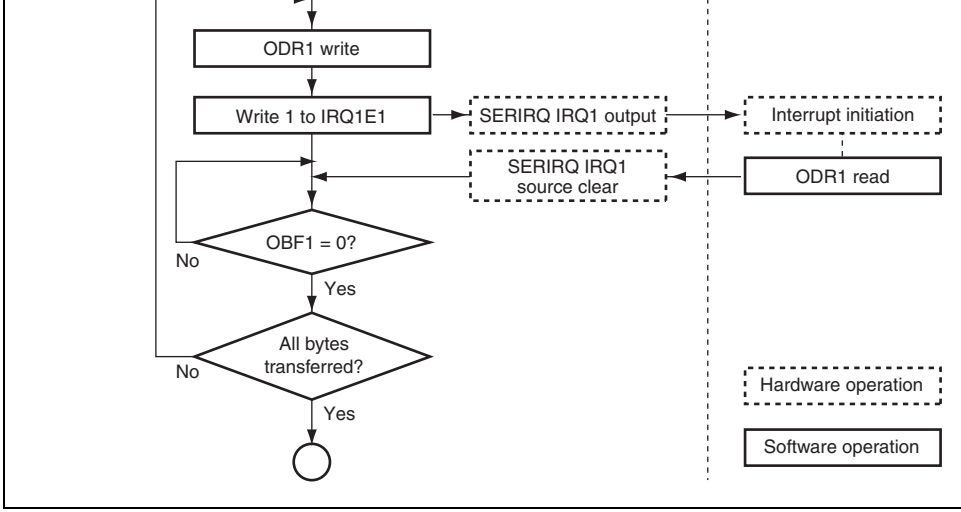


Figure 19.12 HIRQ Flowchart (Example of Channel 1)

Unlike the IDR and ODR registers, the transfer direction is not fixed for the bidirectional registers (TWR). MWMF and SWMF are provided in STR to handle this situation. After to TWR0, MWMF and SWMF must be used to confirm that the write authority for TWR TWR15 has been obtained.

Table 19.14 shows host address examples for LADR3 and registers, IDR3, ODR3, STR3, TWR0MW, TWR0SW, and TWR1 to TWR15.

TWR3	H'A253	H'3FC3
TWR4	H'A254	H'3FC4
TWR5	H'A255	H'3FC5
TWR6	H'A256	H'3FC6
TWR7	H'A257	H'3FC7
TWR8	H'A258	H'3FC8
TWR9	H'A259	H'3FC9
TWR10	H'A25A	H'3FCA
TWR11	H'A25B	H'3FCB
TWR12	H'A25C	H'3FCC
TWR13	H'A25D	H'3FCD
TWR14	H'A25E	H'3FCE
TWR15	H'A25F	H'3FCF

20.1 Features

- Transmission and reception of Ethernet/IEEE802.3 frames
- Supports 10/100 Mbps receive/transfer
- Supports full-duplex and half-duplex modes
- Conforms to IEEE802.3u standard RMII (Reduced Media Independent Interface)
- Magic Packet detection and Wake-On-LAN (WOL) signal output
- Conforms to IEEE802.3x flow control

Note: The EtherC operates only in high-speed mode.

Figure 20.1 shows the configuration of the EtherC.

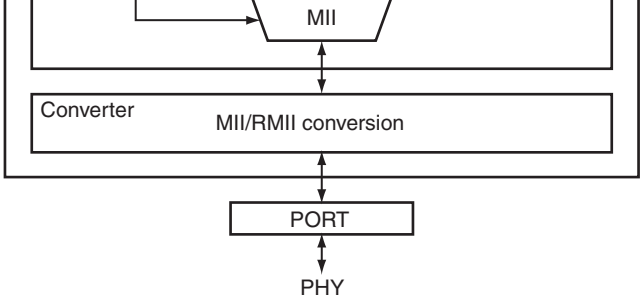


Figure 20.1 Configuration of EtherC

	RM_TX-EN	Output	Transmit Enable Indicates that transmit data is ready on pins RI and RM_TXD0.
	RM_TXD1 RM_TXD0	Output	Transmit Data 2-bit transmit data
	RM_CRS-DV	Input	Carrier Detection/Receive Data Valid Carrier detection signal/Signal that indicates th receive data is on pins RM_RXD1 and RM_RX
	RM_RXD1 RM_RXD0	Input	Receive Data 2-bit receive data
	RM_RX-ER	Input	Receive Error Indicates the error state during data reception.
PHY register interface signals	MDC	Output	Management Data Clock Reference clock signal for information transfer
	MDIO	Input/ Output	Management Data I/O Bidirectional signal for exchange of manageme information between the station management (STA) and physical layer (PHY)
Others	LNKSTA	Input	Link Status Inputs link status from PHY-LSI
	WOL	Output	Wake-On-LAN Signal indicating reception of Magic Packet
	EXOUT	Output	External Output

- MAC address high register (MAHR)
- MAC address low register (MALR)
- Receive frame length register (RFLR)
- PHY status register (PSR)
- Transmit retry over counter register (TROCR)
- Delayed collision detect counter register (CDCR)
- Lost carrier counter register (LCCR)
- Carrier not detect counter register (CNDCR)
- CRC error frame counter register (CEFCR)
- Frame receive error counter register (FRECR)
- Too-short frame receive counter register (TSFRCR)
- Too-long frame receive counter register (TLFRCR)
- Residual-bit frame counter register (RFCR)
- Multicast address frame counter register (MAFCR)
- IPG register (IPGR)
- Automatic PAUSE frame set register (APR)
- Manual PAUSE frame set register (MPR)
- Automatic PAUSE frame retransmission count set register (TPAUSER)

Bit	Bit Name	Value	R/W	Description
31 to 20	—	All 0	R	Reserved These bits are always read as 0. The initial v should not be changed.
19	ZPF	0	R/W	0-Time PAUSE Frame Use Enable 0: Disables PAUSE frame control in which th parameter is 0. The next frame is transmitted after the tim indicated by the Timer value has elapsed. EtherC receives a PAUSE frame with the indicated by the Timer value set to 0, the frame is discarded. 1: Enables PAUSE frame control in which th parameter is 0. A PAUSE frame with the Timer value set transmitted when the number of data in th FIFO is less than the FCFTR value before indicated by the Timer value has not elaps the EtherC receives a PAUSE frame with indicated by the Timer value set to 0, the wait state is canceled.
18	PFR	0	R/W	PAUSE Frame Receive Mode 0: PAUSE frame is not transferred to the E-D 1: PAUSE frame is transferred to the E-DMA
17	RXF	0	R/W	Receive Flow Control Operating Mode 0: PAUSE frame detection function is disabl 1: Receive flow control function is enabled

12	PRCEF	0	R/W	Permit Receive CRC Error Frame 0: A frame with a CRC error is received as a f with an error. 1: A frame with a CRC error is received as a f without an error. The CEFCR register is the not incremented. If this bit is clear and a frame with an error is a CRC error is reflected in ECSR of the E-DM the status of the receive descriptor. If this bit i 1, a frame with an error is received as a norm
11, 10	—	All 0	R	Reserved These bits are always read as 0. The initial va should not be changed.
9	MPDE	0	R/W	Magic Packet Detection Enable Enables or disables Magic Packet detection b hardware to allow activation from the Etherne 0: Magic Packet detection is not enabled 1: Magic Packet detection is enabled
8, 7	—	All 0	R	Reserved These bits are always read as 0. The initial va should not be changed.
6	RE	0	R/W	Reception Enable 0: Receive function is disabled 1: Receive function is enabled If this bit is changed from enabling to disablin frame is being received, the receive function r enabled until reception of the frame is comple

				This bit is always read as 0. The initial value not be changed.
3	ILB	0	R/W	Internal Loop Back Mode Specifies loopback mode in the EtherC. 0: Normal data transmission/reception is per 1: When DM = 1, data loopback is performed the MAC in the EtherC.
2	—	0	R	Reserved This bit is always read as 0. The initial value not be changed.
1	DM	0	R/W	Duplex Mode Specifies the EtherC transfer method. 0: Half-duplex transfer is specified 1: Full-duplex transfer is specified
0	PRM	0	R/W	Promiscuous Mode Setting this bit enables all Ethernet frames to received. All Ethernet frames means all rece frames, irrespective of differences in or presence/absence of the destination addres broadcast address, multicast bit, etc. 0: EtherC performs normal operation 1: EtherC performs promiscuous mode oper

Note: Bits other than TE and RE should be rewritten while both transmission and recep disabled (TE = 0 and also RE =0).

31 to 5	—	All 0	R	Reserved These bits are always read as 0. The initial v should not be changed.
4	PSRTO	0	R/W	PAUSE Frame Retransmission Retry Over Indicates that during the retransmission of P frames when the flow control is enabled, the of retries has exceeded the upper limit set in automatic PAUSE frame retransmission cou register (TPAUSER). 0: Number of PAUSE frame retransmissions exceeded the upper limit 1: Number of PAUSE frame retransmissions exceeded the upper limit
3	—	0	R	Reserved This bit is always read as 0. The initial value not be changed.
2	LCHNG	0	R/W	Link Signal Change Indicates that the LNKSTA signal input from has changed from high to low or low to high. To check the current Link state, refer to the L in the PHY status register (PSR). 0: Changes in the LNKSTA signal are not de 1: Changes in the LNKSTA signal are detect to low or low to high)

PHY occurs before the software recognition
the correct information may not be obtained
the timing specification for the PHY used.

0: LSI has not detected an illegal carrier on

1: LSI has detected an illegal carrier on the

4	PSRTOIP	0	R/W	PAUSE Frame Retransmission Retry Over Interrupt Enable 0: Interrupt notification by the PSRTO bit is d 1: Interrupt notification by the PSRTO bit is e
3	—	0	R	Reserved This bit is always read as 0. The initial value should not be changed.
2	LCHNGIP	0	R/W	LINK Signal Changed Interrupt Enable 0: Interrupt notification by the LCHNG bit is d 1: Interrupt notification by the LCHNG bit is e
1	MPDIP	0	R/W	Magic Packet Detection Interrupt Enable 0: Interrupt notification by the MPD bit is disa 1: Interrupt notification by the MPD bit is ena
0	ICDIP	0	R/W	Illegal Carrier Detection Interrupt Enable 0: Interrupt notification by the ICD bit is disab 1: Interrupt notification by the ICD bit is enab

				Indicates the level of the MDIO pin.
2	MDO	0	R/W	MII Management Data-Out Outputs the value set to this bit from the MDI when the MMD bit is 1.
1	MMD	0	R/W	MII Management Mode Specifies the data read/write direction with re the MII. 0: Read direction is indicated 1: Write direction is indicated
0	MDC	0	R/W	MII Management Data Clock Outputs the value set to this bit from the MDC supplies the MII with the management data cl the method of accessing the MII registers, se 20.4.4, Accessing MII Registers.

				These bits are used to set the upper 32 bits of address. If the MAC address is 01-23-45-67-89-AB (hexadecimal), the value set in this register is H'01234567.

20.3.6 MAC Address Low Register (MALR)

MALR is a 32-bit readable/writable register that specifies the lower 16 bits of the 48-bit M address. The settings in this register are normally made in the initialization process after a The MAC address setting must not be changed while the transmitting and receiving funct enabled. To switch the MAC address setting, return the EtherC and E-DMAC to their init by means of the SWR bit in EDMR before making settings again.

Bit	Bit Name	Initial Value	R/W	Description
31 to 16	—	All 0	R	Reserved These bits are always read as 0. The initial v should not be changed.
15 to 0	MA15 to MA0	All 0	R/W	MAC Address Bits 15 to 0 These bits are used to set the lower 16 bits o MAC address. If the MAC address is 01-23-45-67-89-AB (hexadecimal), the value set in this register i H'000089AB.

11 to 0	RFL11 to RFL0	All 0	R/W	Receive Frame Length 11 to 0 The frame length described here refers to a from the destination address up to and inclu CRC data. Frame contents from the destina address up to and including the data are ac transferred to memory. CRC data is not inc the transfer. When data that exceeds the specified value received, the part of the data that exceeds specified value is discarded. H'000 to H'5EE: 1,518 bytes H'5EF: 1,519 bytes H'5F0: 1,520 bytes : : H'7FF: 2,047 bytes H'800 to H'FFF: 2,048 bytes

				The Link status can be read by connecting th signal output from the PHY to the LNKSTA p the polarity, refer to the PHY specifications t connected.

20.3.9 Transmit Retry Over Counter Register (TROCR)

TROCR is a 32-bit counter that indicates the number of frames that were unable to be tra in 16 transmission attempts including the retransmission. When 16 transmission attempts failed, TROCR is incremented by 1. When the value in this register reaches H'FFFFFFFF count is halted. The counter value is cleared to 0 by a write to this register with any value

Bit	Bit Name	Initial Value	R/W	Description
31 to 0	TROC31 to TROC0	All 0	R/W	Transmit Retry Over Count These bits indicate the number of frames tha unable to be transmitted in 16 transmission a including retransmission.

20.3.11 Lost Carrier Counter Register (LCCR)

LCCR is a 32-bit counter that indicates the number of times the carrier was lost during d transmission. When the value in this register reaches H'FFFFFFFF, the count is halted. T counter value is cleared to 0 by writing to this register with any value.

Bit	Bit Name	Initial Value	R/W	Description
31 to 0	LCC31 to LCC0	All 0	R/W	Lost Carrier Count These bits indicate the number of times the was lost during data transmission.

20.3.12 Carrier Not Detect Counter Register (CNDCR)

CNDCR is a 32-bit counter that indicates the number of times the carrier could not be de while the preamble was being sent. When the value in this register reaches H'FFFFFFFF is halted. The counter value is cleared to 0 by a write to this register with any value.

Bit	Bit Name	Initial Value	R/W	Description
31 to 0	CNDC31 to CNDC0	All 0	R/W	Carrier Not Detect Count These bits indicate the number of times the was not detected.

20.3.14 Frame Receive Error Counter Register (FRECR)

FRECR is a 32-bit counter that indicates the number of frames input from the PHY-LSI f
a receive error was indicated by the RM_RX-ER pin. FRECR is incremented each time th
RM_RX-ER pin becomes active. When the value in this register reaches H'FFFFFFFF, th
is halted. The counter value is cleared to 0 by a write to this register with any value.

Bit	Bit Name	Initial Value	R/W	Description
31 to 0	FREC31 to FREC0	All 0	R/W	Frame Receive Error Count These bits indicate the count of errors during reception.

20.3.15 Too-Short Frame Receive Counter Register (TSFRCR)

TSFRCR is a 32-bit counter that indicates the number of frames of fewer than 64 bytes th
been received. When the value in this register reaches H'FFFFFFFF, the count is halted. T
counter value is cleared to 0 by a write to this register with any value.

Bit	Bit Name	Initial Value	R/W	Description
31 to 0	TSFC31 to TSFC0	All 0	R/W	Too-Short Frame Receive Count These bits indicate the count of frames recei a length of less than 64 bytes.

31 to 0	TLFC31 to TLFC0	All 0	R/W	Too-Long Frame Receive Count These bits indicate the count of frames rec a length exceeding the value in RFLR.

20.3.17 Residual-Bit Frame Counter Register (RFCR)

RFCR is a 32-bit counter that indicates the number of frames received containing residu (less than an 8-bit unit). When the value in this register reaches H'FFFFFFFF, the count The counter value is cleared to 0 by a write to this register with any value.

Bit	Bit Name	Initial Value	R/W	Description
31 to 0	RFC31 to RFC0	All 0	R/W	Residual-Bit Frame Count These bits indicate the count of frames rec containing residual bits.

20.3.18 Multicast Address Frame Counter Register (MAFCR)

MAFCR is a 32-bit counter that indicates the number of frames received with a specifie address. When the value in this register reaches H'FFFFFFFF, the count is halted. The c value is cleared to 0 by a write to this register with any value.

Bit	Bit Name	Initial Value	R/W	Description
31 to 0	MAFC31 to MAFC0	All 0	R/W	Multicast Address Frame Count These bits indicate the count of multicast fr received.

4 to 0	IPG4 to IPG0	H'13	R/W	Inter Packet Gap Sets the IPG value every 4-bit time. H'00: 20-bit time H'01: 24-bit time : : H'13: 96-bit time (Initial value) : : H'1F: 144-bit time

20.3.20 Automatic PAUSE Frame Set Register (APR)

APR sets the TIME parameter value of the automatic PAUSE frame. When transmitting t
automatic PAUSE frame, the value set in this register is used as the TIME parameter of th
PAUSE frame.

Bit	Bit Name	Initial Value	R/W	Description
31 to 16	—	All 0	R	Reserved These bits are always read as 0. The write v should always be 0.
15 to 0	AP15 to AP0	All 0	R/W	Automatic PAUSE Sets the TIME parameter value of the autom PAUSE frame. At this time, 1 bit means 512-

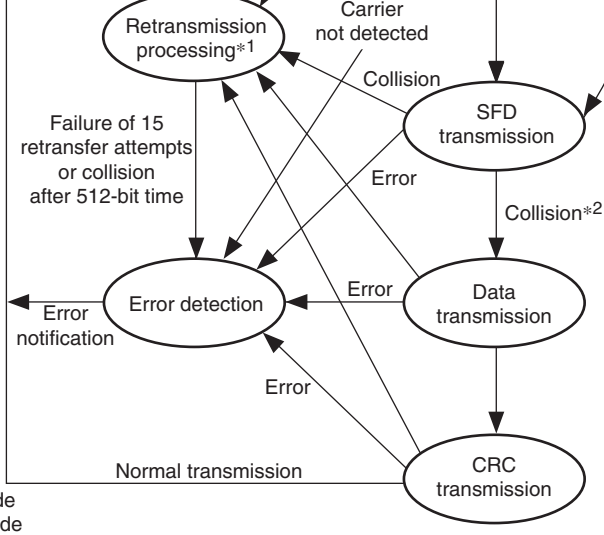
				Sets the TIME parameter value of the manu PAUSE frame. At this time, 1 bit means 512 Read values are undefined.

20.3.22 Automatic PAUSE Frame Retransmission Count Set Register (TPAUSER

TPAUSER sets the upper limit of the number of times of the PAUSE frame retransmissi
TPAUSER must not be changed while the transmitting function is enabled.

Bit	Bit Name	Initial Value	R/W	Description
31 to 16	—	All 0	R	Reserved These bits are always read as 0. The write should always be 0.
15 to 0	TPAUSE15 to TPAUSE0	All 0	R/W	Upper Limit of the Number of Times of PAU Frame Retransmission H'0000: Unlimited number of times of retran H'0001: Retransmit once : : H'FFFF: Number of times of retransmission

transmitter.



[Legend]
FDPX: Path for full-duplex mode
HDPX: Path for half-duplex mode
SFD: Start Frame Delimiter

Notes: 1. Retransmission processing includes both jam transmission that depends on collision detection and the adjustment of transmission intervals based on the back-off algorithm
2. Retransmission processing is only performed when data of 512 bits or less (including preamble and SFD) is transmitted. When a collision is detected during transmission of greater than 512 bits, only jam is transmitted and retransmission based on the back-off algorithm is not performed.

Figure 20.2 EtherC Transmitter State Transitions

more transmit data, continues transmitting.

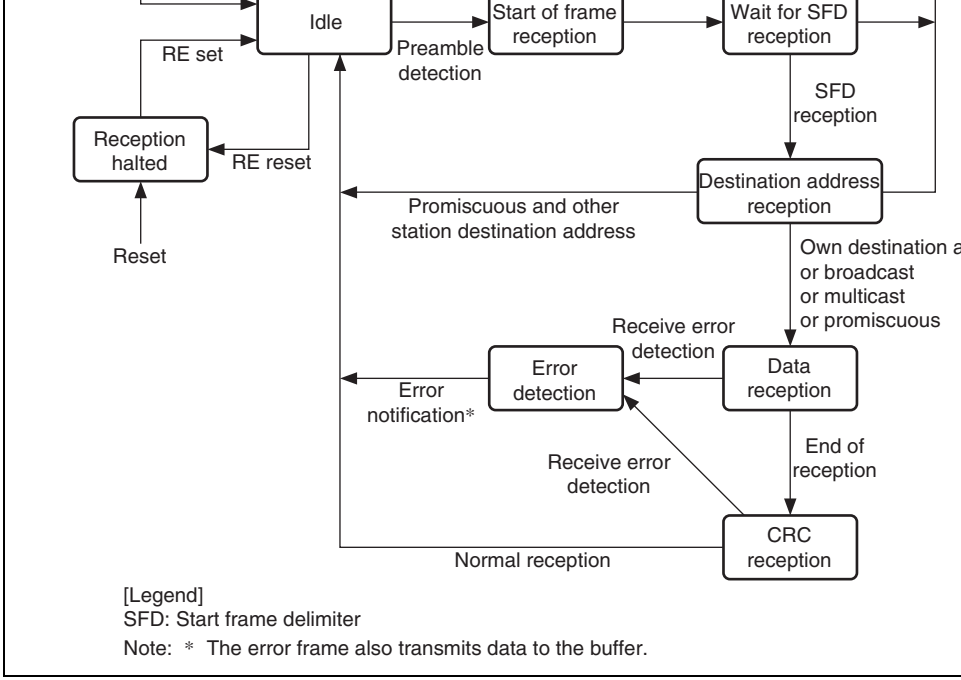


Figure 20.3 EtherC Receiver State Transmissions

1. When the receive enable (RE) bit is set, the receiver enters the receive idle state.
2. When an SFD (start frame delimiter) is detected after a receive packet preamble, the starts receive processing. Discards a frame with an invalid pattern.
3. In normal mode, if the destination address matches the receiver's own address, or if or multicast transmission or promiscuous mode is specified, the receiver starts data r

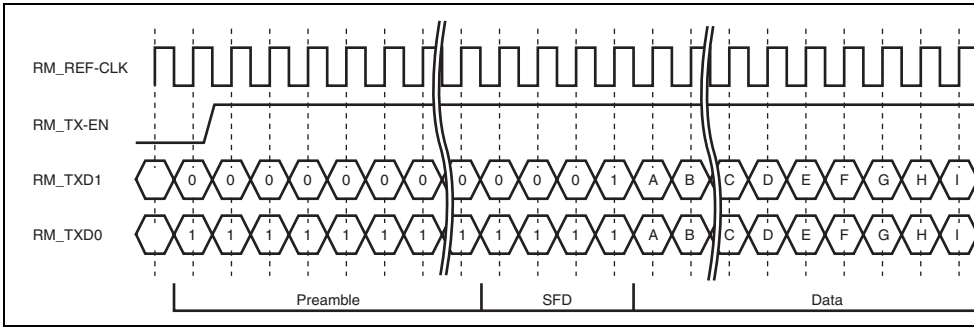


Figure 20.4 RMII Frame Transmit Timing (Normal Transmission)

(2) RMII Frame Reception Timing

Timing of RMII frame reception is shown in figures 20.5 and 20.6.

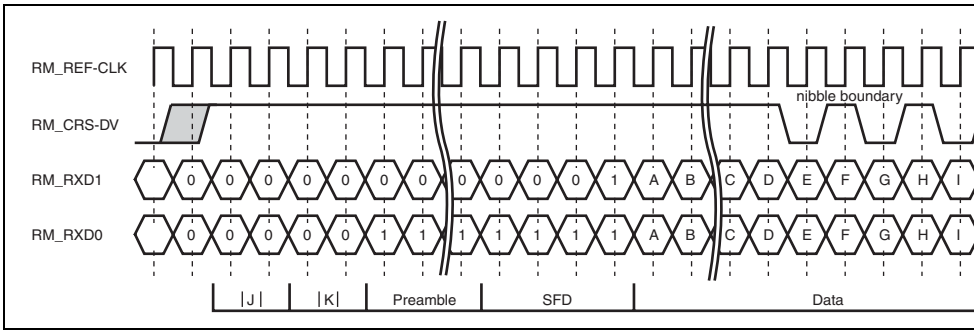


Figure 20.5 RMII Frame Receive Timing (Normal Reception)

Access Type	MII Management Frame							
Item	PRE	ST	OP	PHYAD	REGAD	TA	DATA	ID
Number of bits	32	2	2	5	5	2	16	
Read	1..1	01	10	00001	RRRRR	Z0	D..D	
Write	1..1	01	01	00001	RRRRR	10	D..D	

[Legend]

PRE: 32 consecutive 1s

ST: Write of 01 indicating start of frame

OP: Write of code indicating access type

PHYAD: Write of 0001 if the PHY address is 1 (sequential write starting with the MSB).
This bit changes depending on the PHY address.

REGAD: Write of 0001 if the register address is 1 (sequential write starting with the MSB).
This bit changes depending on the PHY register address.

TA: Time for switching data transmission source on MII interface
(a) Read: Bus is released (indicated as Z0).
(b) Write: B'10 is written.

DATA: 16-bit data. Sequential write or read from MSB
(a) Read: 16-bit data read
(b) Write: 16-bit data write

IDLE: Wait time until next MII management format input
(a) Read: Since the bus has been relased at TA already, control is not required.
(b) Write: Independent bus release (indicated as X) is performed.

Figure 20.7 MII Management Frame Format

Figure 20.8 1-Bit Data Write Flowchart

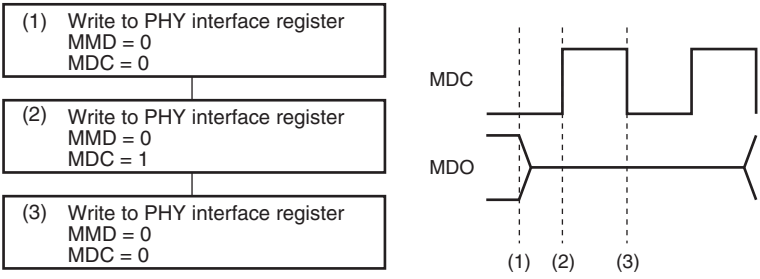


Figure 20.9 Bus Release Flowchart (TA in Read in Figure 20.7)

Figure 20.10 1-Bit Data Read Flowchart

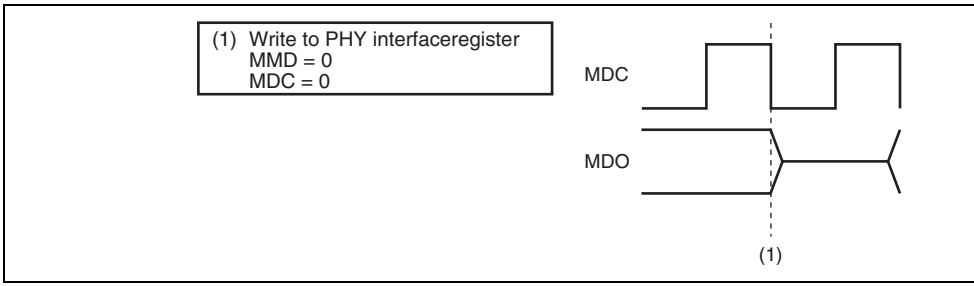


Figure 20.11 Independent Bus Release Flowchart (IDLE in Write in Figure 2

With a Magic Packet, reception is performed regardless of the destination address. As a function is valid, and the WOL pin enabled, only in the case of a match with the destina address specified by the format in the Magic Packet. Further information on Magic Pack found in the technical documentation published by AMD Corporation.

The procedure for using the WOL function with this LSI is as follows.

1. Disable interrupt source output by means of the various interrupt enable/mask regist
2. Set the Magic Packet detection enable bit (MPDE) in the EtherC mode register (ECM
3. Set the Magic Packet detection interrupt enable bit (MPDIP) in the EtherC interrupt register (ECSIPR) to the enable setting.
4. If necessary, set the CPU operating mode to sleep mode or set supporting functions t standby mode.
5. When a Magic Packet is detected, an interrupt is sent to the CPU. The WOL pin noti peripheral LSIs that the Magic Packet has been detected.

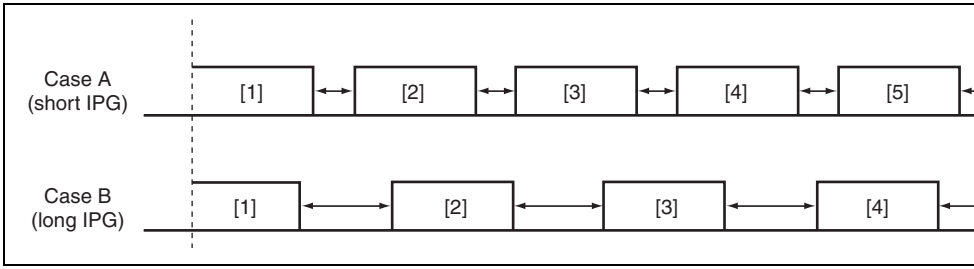


Figure 20.12 Changing IPG and Transmission Efficiency

20.4.7 Flow Control

The EtherC supports flow control functions conforming to IEEE802.3x in full-duplex ope
Flow control can be applied to both receive and transmit operations. The methods for tran
PAUSE frames when controlling flow are as follows:

(1) Automatic PAUSE Frame Transmission

For receive frames, PAUSE frames are automatically transmitted when the number of dat
receive FIFO (included in E-DMAC) reaches the value set in the flow control FIFO thres
register (FCFTR) of the E-DMAC. The TIME parameter included in the PAUSE frame a
time is set by the automatic PAUSE frame setting register (APR). The automatic PAUSE
transmission is repeated until the number of data in the receive FIFO becomes less than t
FCFTR setting as the receive data is read from the FIFO.

The upper limit of the number of retransmissions of the PAUSE frame can also be set by
automatic PAUSE frame retransmission count set register (TPAUSER). In this case, PAU
frame transmission is repeated until the number of data becomes FCFTR value set or belo

receiving a PAUSE frame. However, the transmission of the current frame is continued.
received PAUSE frame is valid only when the RXF bit in the EtherC mode register (EC
to 1.

during the software reset state regardless of the external pin level.

Clear the LCHNG bit before setting the LCHNGIP bit in ECSIPR not to request a LINK changed interrupt accidentally.

20.5.2 Flow Control Defect 1

Once a PAUSE frame is received while the receiving flow control is enabled in full-duple (the RXF bit in ECMR = 1), each time when the local station receives a normal unicast fr (non-PAUSE frame without a CRC error), the TIME parameter specified by the PAUSE f that has been previously received is incorrectly applied. As a result, unnecessary waiting generated to slow down the transmission throughput. The TIME parameter value is maint until another PAUSE frame is received.

This defect can be prevented if the destination station supports the function to transmit th PAUSE frame as the same as this LSI does. Enable the use of 0-time PAUSE frame in thi (the ZPF bit in ECMR = 1) before the 0-time PAUSE frame is received from the destinati station. This clears the TIME parameter incorrectly maintained in the EtherC and prevent unnecessary waiting time for transmission to be generated.

20.5.3 Flow Control Defect 2

When a PAUSE period is generated while the transmitting/receiving flow control is enabl full-duplex mode (the TXF/RXF bit in ECMR = 1), non-PAUSE frames are waited for transmission (this is a normal operation) whereas PAUSE frames are incorrectly waited fo transmission. The transmission of non-PAUSE frames in a PAUSE period is prohibited, t the transmission of PAUSE frames is enabled in IEEE802.3.

When a PAUSE period is generated by the request from the destination station (that is, a P frame is received from the destination station), the load of the destination station is high a

The E-DMAC has the following features:

- The load on the CPU is reduced by means of a descriptor management system
- Transmit/receive frame status information is indicated in descriptors
- Achieves efficient system bus utilization through the use of block transfer (16-byte u
- Supports single-frame/multi-buffer operation

Figure 21.1 shows the configuration of the E-DMAC, and the descriptors and transmit/r
buffers in memory.

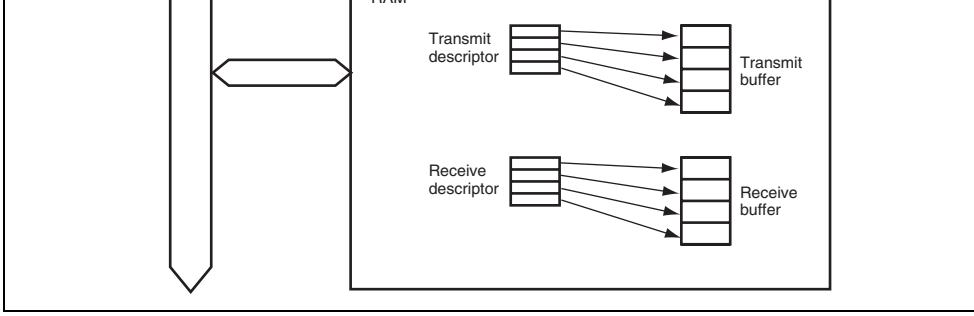


Figure 21.1 Configuration of E-DMAC, and Descriptors and Buffers

21.2 Register Descriptions

The E-DMAC has the following registers.

- E-DMAC mode register (EDMR)
- E-DMAC transmit request register (EDTRR)
- E-DMAC receive request register (EDRRR)
- Transmit descriptor list address register (TDLAR)
- Receive descriptor list address register (RDLAR)
- EtherC/E-DMAC status register (EESR)
- EtherC/E-DMAC status interrupt permission register (EESIPR)
- Transmit/receive status copy enable register (TRSCER)
- Receive missed-frame counter register (RMFCR)
- Transmit FIFO threshold register (TFTR)
- FIFO depth register (FDR)
- Receiving method control register (RMCR)

Bit	Bit Name	Initial value	R/W	Description
31 to 7	—	All 0	R	Reserved These bits are always read as 0. The initial v should not be changed.
6	DE	0	R/W	E-DMAC Data Endian Convert Selects whether or not the endian format is c on data transfer by the E-DMAC. However, tl endian format of the descriptors and E-DMAC values are not converted regardless of this b 0: Endian format not converted (big endian) 1: Endian format converted (little endian)
5	DL1	0	R/W	Transmit/Receive Descriptor Length
4	DL0	0	R/W	These bits specify the transmit/receive descr length. 00: 16 bytes 01: 32 bytes 10: 64 bytes 11: Setting prohibited
3 to 1	—	All 0	R	Reserved These bits are always read as 0. The initial v should not be changed.

				0: Writing 0 is ignored (E-DMAC operation is affected)
				1: Writing 1 resets the EtherC and E-DMAC a automatically cleared

21.2.2 E-DMAC Transmit Request Register (EDTRR)

The EDTRR is a 32-bit readable/writable register that issues transmit directives to the E When transmission of one frame is completed, the next descriptor is read. If the transmit descriptor active bit in this descriptor has the "active" setting, transmission is continued. transmit descriptor active bit has the "inactive" setting, the TR bit is cleared and operatio transmit DMAC is halted.

Bit	Bit Name	Initial value	R/W	Description
31 to 1	—	All 0	R	Reserved These bits are always read as 0. The initial va should not be changed.
0	TR	0	R/W	Transmit Request Check TR = 0 before transmission is started. 0: Transmission-halted state. Writing 0 does r transmission. Termination of transmission i controlled by the active bit in the transmit d 1: Start of transmission. The relevant descript and a frame is sent with the transmit active 1

Bit	Bit Name	value	R/W	Description
31 to 1	—	All 0	R	Reserved These bits are always read as 0. The initial v should not be changed.
0	RR	0	R/W	Receive Request Check RR = 0 before reception is started. 0: The receive function is disabled* 1: A receive descriptor is read and the E-DM ready to receive

Note: * If the receive function is disabled during frame reception, write-back is not perf successfully to the receive descriptor. Following pointers to read a receive des become abnormal and the E-DMAC cannot operate successfully. In this case, the E-DMAC reception enabled again, execute a software reset by the SWR bi EDMR. To make the E-DMAC reception disabled without executing a software set the RE bit in ECMR. Next, after the E_DMAC has completed the reception write-back to the receive descriptor has been confirmed, disable the receive fu this register.

	TDLA0			The lower bits are set as follows according t specified descriptor length. 16-byte boundary: TDLA3 to TDLA0 = 0000 32-byte boundary: TDLA4 to TDLA0 = 0000 64-byte boundary: TDLA5 to TDLA0 = 0000

21.2.5 Receive Descriptor List Address Register (RDLAR)

RDLAR is a 32-bit readable/writable register that specifies the start address of the receiv descriptor list. Descriptors have a boundary configuration in accordance with the descrip indicated by the DL bit in EDMR. This register must not be written to during reception. Modifications to this register should only be made while reception is disabled by the RR in the E-DMAC Receive Request Register (EDRRR).

Bit	Bit Name	Initial value	R/W	Description
31 to 0	RDLA31 to RDLA0	All 0	R/W	Receive Descriptor Start Address The lower bits are set as follows according t specified descriptor length. 16-byte boundary: RDLA3 to RDLA0 = 0000 32-byte boundary: RDLA4 to RDLA0 = 0000 64-byte boundary: RDLA5 to RDLA0 = 0000

Bit	Bit Name	Initial value	R/W	Description
31	—	0	R	Reserved This bit is always read as 0. The initial value not be changed.
30	TWB	0	R/W	Write-Back Complete Indicates that write-back from the E-DMAC to corresponding descriptor has completed. This operation is enabled when the TIS bit in TRIM to 1. 0: Write-back has not completed, or no trans directive 1: Write-back has completed
29 to 27	—	All 0	R	Reserved These bits are always read as 0. The initial v should not be changed.
26	TABT	0	R/W	Transmit Abort Detection Indicates that the EtherC aborts transmitting because of failures during transmitting the fra 0: Frame transmission has not been aborted transmit directive 1: Frame transmit has been aborted

				overflowed. 0: Receive frame counter has not overflowe 1: Receive frame counter overflows
23	ADE	0	R/W	Address Error Indicates that the memory address that the tried to transfer is found illegal. 0: Illegal memory address not detected (nor operation) 1: Illegal memory address detected Note: When an address error is detected, th E-DMAC halts transmitting/receiving. resume the operation, set the E-DMA after software reset by means of the S EDMR.
22	ECI	0	R	EtherC Status Register Interrupt Source This bit is a read-only bit. When the source ECSR interrupt in the EtherC is cleared, this also cleared. 0: EtherC status interrupt source has not be detected 1: EtherC status interrupt source has been

				transmission, the E-DMAC writes the transm status back to the descriptor. 0: Transfer not complete, or no transfer direc 1: Transfer complete
20	TDE	0	R/W	Transmit Descriptor Empty Indicates that the transmission descriptor val (TACT) in the descriptor is not set when the reads the transmission descriptor when the p descriptor is not the last one of the frame for buffer frame processing. As a result, an incor frame may be transmitted. 0: Transmit descriptor active bit TACT = 1 de 1: Transmit descriptor active bit TACT = 0 de When transmission descriptor empty (TDE = occurs, execute a software reset and initiate transmission. In this case, the address that is in the transmit descriptor list address register (TDLAR) is transmitted first.
19	TFUF	0	R/W	Transmit FIFO Underflow Indicates that underflow has occurred in the FIFO during frame transmission. Incomplete sent onto the line. 0: Underflow has not occurred 1: Underflow has occurred

				receive descriptor and initiating receiving. 0: Receive descriptor active bit RACT = 1 n detected 1: Receive descriptor active bit RACT = 0 d
16	RFOF	0	R/W	Receive FIFO Overflow Indicates that the receive FIFO has overflow frame reception. 0: Overflow has not occurred 1: Overflow has occurred
15 to 12	—	All 0	R	Reserved These bits are always read as 0. The initial should not be changed.
11	CND	0	R/W	Carrier Not Detect Indicates the carrier detection status. 0: A carrier is detected when transmission s 1: A carrier is not detected when transmissi
10	DLC	0	R/W	Detect Loss of Carrier Indicates that loss of the carrier has been d during frame transmission. 0: Loss of carrier not detected 1: Loss of carrier detected
9	CD	0	R/W	Delayed Collision Detect Indicates that a delayed collision has been during frame transmission. 0: Delayed collision not detected 1: Delayed collision detected

				0: Multicast address frame has not been rece 1: Multicast address frame has been receive
6, 5	—	All 0	R	Reserved These bits are always read as 0. The initial v should not be changed.
4	RRF	0	R/W	Receive Residual-Bit Frame 0: Residual-bit frame has not been received 1: Residual-bit frame has been received
3	RTLF	0	R/W	Receive Too-Long Frame Indicates that the frame more than the numb receive frame length upper limit set by RFLR EtherC has been received. 0: Too-long frame has not been received 1: Too-long frame has been received
2	RTSF	0	R/W	Receive Too-Short Frame Indicates that a frame of fewer than 64 bytes been received. 0: Too-short frame has not been received 1: Too-short frame has been received
1	PRE	0	R/W	PHY Receive Error 0: PHY receive error not detected 1: PHY receive error detected
0	CERF	0	R/W	CRC Error on Received Frame 0: CRC error not detected 1: CRC error detected

30	TWBIP	0	R/W	Write-Back Complete Interrupt Permission 0: Write-back complete interrupt is disabled 1: Write-back complete interrupt is enabled
29 to 27	—	All 0	R	Reserved These bits are always read as 0. The initial va should not be changed.
26	TABTIP	0	R/W	Transmit Abort Detection Interrupt Permissio 0: Transmit abort detection interrupt is disabl 1: Transmit abort detection interrupt is enable
25	RABTIP	0	R/W	Receive Abort Detection Interrupt Permission 0: Receive abort detection interrupt is disable 1: Receive abort detection interrupt is enable
24	RFCOFIP	0	R/W	Receive Frame Counter Overflow Interrupt P 0: Receive frame counter overflow interrupt is 1: Receive frame counter overflow interrupt is
23	ADEIP	0	R/W	Address Error Interrupt Permission 0: Address error interrupt is disabled 1: Address error interrupt is enabled
22	ECIIP	0	R/W	EtherC Status Register Interrupt Permission 0: EtherC status interrupt is disabled 1: EtherC status interrupt is enabled
21	TCIP	0	R/W	Frame Transmit Complete Interrupt Permissi 0: Frame transmit complete interrupt is disabl 1: Frame transmit complete interrupt is enabl

				1: Frame received interrupt is enabled
17	RDEIP	0	R/W	Receive Descriptor Empty Interrupt Permissio 0: Receive descriptor empty interrupt is disabl 1: Receive descriptor empty interrupt is enable
16	RFOFIP	0	R/W	Receive FIFO Overflow Interrupt Permission 0: Receive FIFO overflow interrupt is disabled 1: Receive FIFO overflow interrupt is enabled
15 to 12	—	All 0	R	Reserved These bits are always read as 0. The initial val should not be changed.
11	CNDIP	0	R/W	Carrier Not Detect Interrupt Permission 0: Carrier not detect interrupt is disabled 1: Carrier not detect interrupt is enabled
10	DLCIP	0	R/W	Detect Loss of Carrier Interrupt Permission 0: Detect loss of carrier interrupt is disabled 1: Detect loss of carrier interrupt is enabled
9	CDIP	0	R/W	Delayed Collision Detect Interrupt Permission 0: Delayed collision detect interrupt is disabled 1: Delayed collision detect interrupt is enabled
8	TROIP	0	R/W	Transmit Retry Over Interrupt Permission 0: Transmit retry over interrupt is disabled 1: Transmit retry over interrupt is enabled

				1: Receive residual-bit frame interrupt is enab
3	RTLFIP	0	R/W	Receive Too-Long Frame Interrupt Permissio
				0: Receive too-long frame interrupt is disable
				1: Receive too-long frame interrupt is enable
2	RTSFIP	0	R/W	Receive Too-Short Frame Interrupt Permissio
				0: Receive too-short frame interrupt is disabl
				1: Receive too-short frame interrupt is enable
1	PREIP	0	R/W	PHY-LSI Receive Error Interrupt Permission
				0: PHY-LSI receive error interrupt is disabled
				1: PHY-LSI receive error interrupt is enabled
0	CERFIP	0	R/W	CRC Error on Received Frame
				0: CRC error on received frame interrupt is di
				1: CRC error on received frame interrupt is er

31 to 8	—	All 0	R	Reserved These bits are always read as 0. The initial val should not be changed.
7	RMAFCE	0	R/W	RMAF Bit Copy Directive 0: Indicates the RMAF bit state in bit RFE of th receive descriptor 1: Occurrence of the corresponding interrupt is indicated in bit RFE of the receive descriptc
6, 5	—	All 0	R	Reserved These bits are always read as 0. The initial val should not be changed.
4	RRFCE	0	R/W	RRF Bit Copy Directive 0: Indicates the RRF bit state in bit RFE of the descriptor 1: Occurrence of the corresponding interrupt is indicated in bit RFE of the receive descriptc
3 to 0	—	All 0	R	Reserved These bits are always read as 0. The initial val should not be changed.

21.2.9 Receive Missed-Frame Counter Register (RMFCR)

RMFCR is a 16-bit counter that indicates the number of frames missed (discarded, and no transferred to the receive buffer) during reception. When the receive FIFO overflows, the frames in the FIFO are discarded. The number of frames discarded at this time is counted the value in this register reaches H'FFFF, counting-up is halted. When this register is read counter value is cleared to 0. Write operations to this register have no effect.

TFTR is a 32-bit readable/writable register that specifies the transmit FIFO threshold at first transmission is started. The actual threshold is 4 times the set value. The EtherC sta transmission when the amount of data in the transmit FIFO exceeds the number of bytes by this register, when the transmit FIFO is full, or when 1-frame write is executed. Whe this register, do so in the transmission-halt state.

Bit	Bit Name	Initial value	R/W	Description
31 to 11	—	All 0	R	Reserved These bits are always read as 0. The initial should not be changed.

H'02: 32 bytes

: :

H'20: 128 bytes

: :

H'40: 256 bytes

: :

H'80: 512 bytes

H'200: 2048 bytes

Note: When starting transmission before one frame of data write has completed, take ca
generation of the underflow.

10 to 8	TFD2 to TFD0	B'000	R	Transmit FIFO Capacity Specify the capacity of transmit FIFO, from to 2048 bytes, in 256-byte units. The set va not be changed after the transmit/receive op started.
7 to 3	—	All 0	R	Reserved These bits are always read as 0. The initial should not be changed.
2 to 0	RFD2 to RFD0	B'000	R	Receive FIFO Capacity Specify the capacity of receive FIFO, from 2 to 2048 bytes, in 256-byte units. The set va not be changed after the transmit/receive op started.

0	RNC	0	R/W	Receive Enable Control 0: When reception of one frame is completed DMAC writes the receive status into the d and clears the RR bit in EDRRR 1: When reception of one frame is completed DMAC writes the receive status into the d reads the next descriptor, and prepares to the next frame

21.2.13 Receiving-Buffer Write Address Register (RBWAR)

RBWAR stores the address of data to be written in the receiving buffer when the E-DMA data to the receiving buffer. Which addresses in the receiving buffer are processed by the DMAC can be recognized by monitoring addresses displayed in this register. The address E-DMAC is actually processing may be different from the value read from this register.

Bit	Bit Name	Initial value	R/W	Description
31 to 0	RBWA31 to RBWA0	All 0	R	Receiving-Buffer Write Address These bits can only be read. Writing is prohib

	RDFA0			These bits can only be read. Writing is proh

21.2.15 Transmission-Buffer Read Address Register (TBRAR)

TBRAR stores the address of the transmission buffer when the E-DMAC reads data fro transmission buffer. Which addresses in the transmission buffer are processed by the E- can be recognized by monitoring addresses displayed in this register. The address from E-DMAC is actually reading in the buffer may be different from the value read from thi

Bit	Bit Name	Initial value	R/W	Description
31 to 0	TBRA31 to TBRA0	All 0	R	Transmission-Buffer Read Address These bits can only be read. Writing is proh

21.2.16 Transmission-Descriptor Fetch Address Register (TDFAR)

TDFAR stores the descriptor start address that is required when the E-DMAC fetches de information from the transmission descriptor. Which transmission descriptor informatio for processing by the E-DMAC can be recognized by monitoring addresses displayed in register. The address from which the E-DMAC is actually fetching a descriptor may be from the value read from this register.

Bit	Bit Name	Initial value	R/W	Description
31 to 0	TDFA31 to TDFA0	All 0	R	Transmission-Descriptor Fetch Address These bits can only be read. Writing is proh

RFD in FCFTR = 0, flow control is started when (256 − 64) bytes of data is stored in the
FIFO. The value set in the RFD bits in this register should be equal to or less than those i

Bit	Bit Name	Initial value	R/W	Description
31 to 19	—	All 0	R	Reserved These bits are always read as 0. The initial v should not be changed.
18	RFF2	1	R/W	Receive Frame Number Flow Control Thresh
17	RFF1	1	R/W	000: When 2 receive frame has been stored i
16	RFF0	1	R/W	receive FIFO 001: When 4 receive frames have been store receive FIFO : : 110: When 14 receive frames have been stor receive FIFO 111: When 16 receive frames have been stor receive FIFO

110: When (1792 − 32) bytes of data is stor
receive FIFO

001: When (2048 − 64) bytes of data is stor
receive FIFO

21.2.18 Bit Rate Setting Register (ECBRR)

ECBRR sets the bit rate for retransmission and reception.

Bit	Bit Name	Initial Value	R/W	Description
7 to 1	—	All 0	R	Reserved These bits are always read as 0. The write should always be 0.
0	RTM	0	R/W	Transmit/Receive Rate 0: 10 Mbps 1: 100 Mbps

0	TIS	0	R/W	Transmit Interrupt Setting 0: Write-back completion for each frame is n notified 1: Write-back completion for each frame usir TWB bit in EESR is notified

Before starting transmission/reception, the communication program creates transmit and
descriptor lists in memory. The start addresses of these lists are then set in the transmit a
descriptor list start address registers.

The descriptor start address must be aligned so that it matches the address boundary acc
the descriptor length set by the E-DMAC mode register (EDMR). The transmit buffer st
may be set on a byte, a word, and a longword boundary.

(1) Transmit Descriptor

Figure 21.2 shows the relationship between a transmit descriptor and the transmit buffer
According to the specification in this descriptor, the relationship between the transmit fr
transmit buffer can be defined as one frame/one buffer or one frame/multi-buffer.

Figure 21.2 Relationship between Transmit Descriptor and Transmit Buffe

				is suspended. 0: The transmit descriptor is invalid. Indicates that valid data has not been wr this bit by the CPU, or this bit has been r write-back operation on termination of E- frame transfer processing (completion or suspension of transmission) If this state is recognized in an E-DMAC read, the E-DMAC terminates transmit p and transmit operations cannot be contir restart is necessary) 1: The transmit descriptor is valid. Indicates that valid data has been written transmit buffer by the CPU and frame tra processing has not yet been executed, o frame transfer is in progress When this state is recognized in an E-DM descriptor read, the E-DMAC continues v transmit operation
30	TDLE	0	R/W	Transmit Descriptor List End After completion of the corresponding buffe the E-DMAC references the first descriptor. specification is used to set a ring configurati transmit descriptors. 0: This is not the last transmit descriptor list 1: This is the last transmit descriptor list

				01: Transmit buffer indicated by this descripto contains end of frame (frame is conclude 10: Transmit buffer indicated by this descripto of frame (frame is not concluded) 11: Contents of transmit buffer indicated by th descriptor are equivalent to one frame (o frame/one buffer)
27	TFE	0	R/W	Transmit Frame Error Indicates that one or other bit of the transmit status indicated by bits 26 to 0 is set. 0: No error during transmission 1: An error occurred during transmission
26 to 0	TFS26 to TFS0	All 0	R/W	Transmit Frame Status TFS26 to TFS9: Reserved (The write value s always be 0.) TFS8: Transmit Abort Detection (indicates ar TFS3 to TFS0 has been set.) TFS7 to TFS4: Reserved (The write value sh always be 0.) TFS3: Carrier Not Detected (corresponds to in EESR) TFS2: Detect Loss of Carrier (corresponds to in EESR) TFS1: Delayed Collision Detect (corresponds bit in EESR) TFS0: Transmit Retry Over (corresponds to T in EESR)

				byte length specified in the descriptors at th and midway can be set in byte units.
15 to 0	—	All 0	R	Reserved These bits are always read as 0. The initial should not be changed.

(c) Transmit Descriptor 2 (TD2)

TD2 specifies the 32-bit transmit buffer start address. The transmit buffer start address s
be on a byte, a word, or a longword boundary.

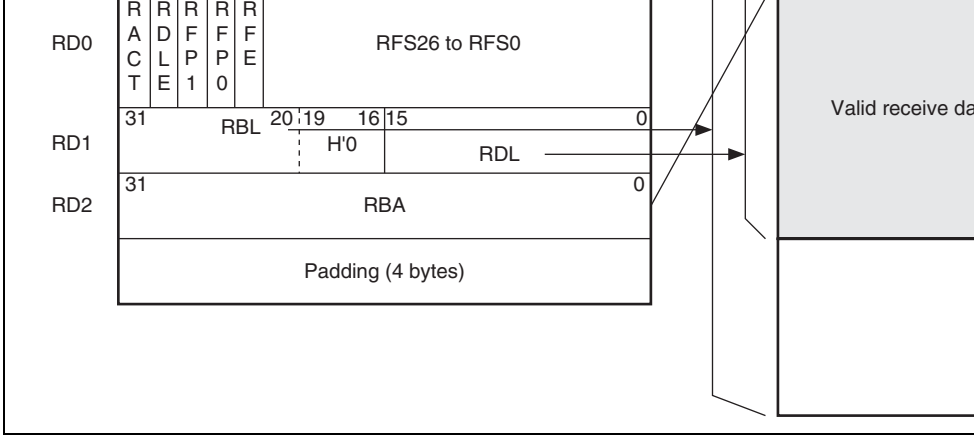


Figure 21.3 Relationship between Receive Descriptor and Receive Buffer

				reception. 0: The receive descriptor is invalid. Indicates that the receive buffer is not re (access disabled by E-DMAC), or this bit reset by a write-back operation on termin E-DMAC frame transfer processing (com suspension of reception). If this state is recognized in an E-DMAC read, the E-DMAC terminates receive pr and receive operations cannot be continu Reception can be restarted by setting RA and executing receive initiation. 1: The receive descriptor is valid Indicates that the receive buffer is ready enabled) and processing for frame transf the FIFO has not been executed, or that transfer is in progress. When this state is recognized in an E-DM descriptor read, the E-DMAC continues v receive operation.
30	RDLE	0	R/W	Receive Descriptor List Last After completion of the corresponding buffe the E-DMAC references the first receive des This specification is used to set a ring config for the receive descriptors. 0: This is not the last receive descriptor list 1: This is the last receive descriptor list

				11: Contents of receive buffer indicated by th descriptor are equivalent to one frame (o frame/one buffer)
27	RFE	0	R/W	Receive Frame Error Indicates that one or other bit of the receive f status indicated by bits 26 to 0 is set. Whethe the receive frame status information is copie bit is specified by the transmit/receive status enable register. 0: No error during reception 1: A certain kind of error occurred during rec

RFS7: Multicast address frame received (corresponds to RMAF bit in EESR)

RFS6: CAM entry unregistered frame recei (corresponds to the RUAF bit in EES

RSF5: Reserved (The write value should al be 0.)

RFS4: Receive residual-bit frame error (cor to RRF bit in EESR)

RFS3: Receive too-long frame error (corres RTLF bit in EESR)

RFS2: Receive too-short frame error (corre RTSF bit in EESR)

RFS1: PHY-LSI receive error (corresponds bit in EESR)

RFS0: CRC error on received frame (corres CERF bit in EESR)

				1,514 bytes, excluding the CRC data. Theref the receive buffer length specification, a valu 1,520 bytes (H'05F0) that takes account of a boundary is set as the maximum receive fran length.
15 to 0	RDL	All 0	R/W	Receive Data Length These bits specify the data length of a receiv stored in the receive buffer. The receive data transferred to the receive b does not include the 4-byte CRC data at the the frame. The receive frame length is report number of words (valid data bytes) not includ CRC data.

(c) Receive Descriptor 2 (RD2)

RD2 specifies the 32-bit receive buffer start address. The receive buffer start address mus
longword boundary.

1. TFP = 00 or 01 (frame continuation):
   Descriptor write-back is performed after DMA transfer.
2. TFP = 01 or 11 (frame end):
   Descriptor write-back is performed after completion of frame transmission.

The E-DMAC continues reading descriptors and transmitting frames as long as the setti
TACT bit in the read descriptors is "active." When a descriptor with an "inactive" TACT
read, the E-DMAC resets the transmit request bit (TR) in the transmit register and ends
processing (EDTRR).

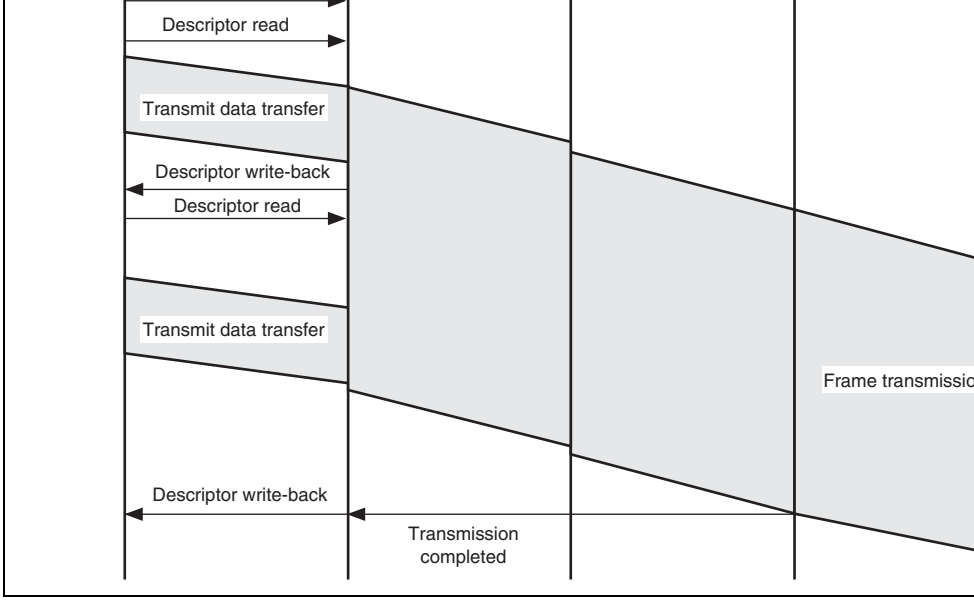


Figure 21.4 Sample Transmission Flowchart

DMAC then continues to transfer data to the receive buffer specified by the new RD2. W
frame reception is completed, or if frame reception is suspended because of a certain kin
the E-DMAC performs write-back to the relevant descriptor (RFP = 11 or 01), and then
receive processing. The E-DMAC then reads the next descriptor and enters the receive-s
state again.

To receive frames continuously, the receive enable control bit (RNC) must be set to 1 in
receive control register (RCR). After initialization, this bit is cleared to 0.

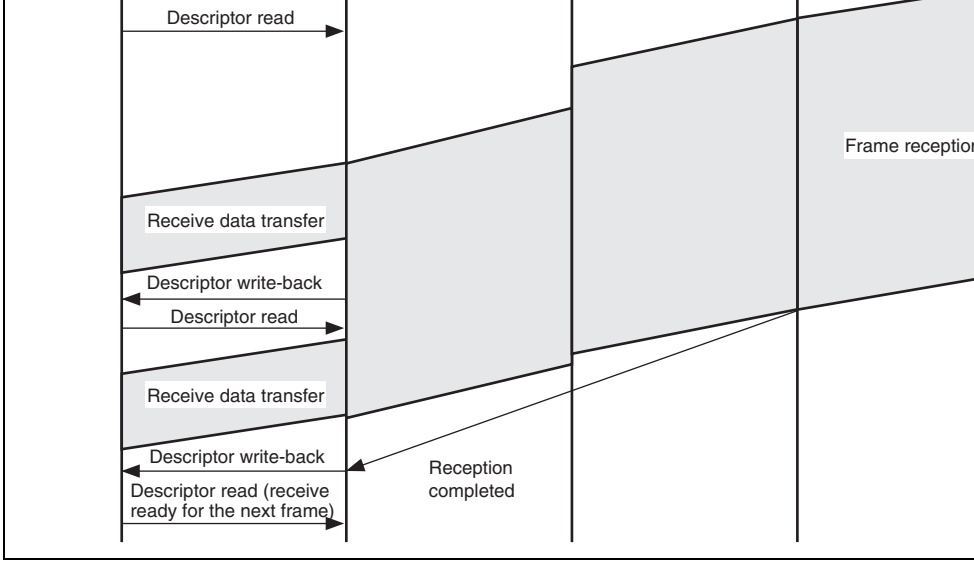


Figure 21.5 Sample Reception Flowchart

bit cleared to 0, immediately. The next descriptor is then read, and the position within th
frame is determined on the basis of bits TFP1 and TFP0 (continuing [B'00] or end [B'01
case of a continuing descriptor, the TACT bit is cleared to 0, only, and the next descripto
immediately. If the descriptor is the final descriptor, not only is the TACT bit cleared to
write-back is also performed to the TFE and TFS bits at the same time. Data in the buff
transmitted between the occurrence of an error and write-back to the final descriptor. If
interrupts are enabled in the EtherC/E-DMAC status interrupt permission register (EESI
interrupt is generated immediately after the final descriptor write-back.

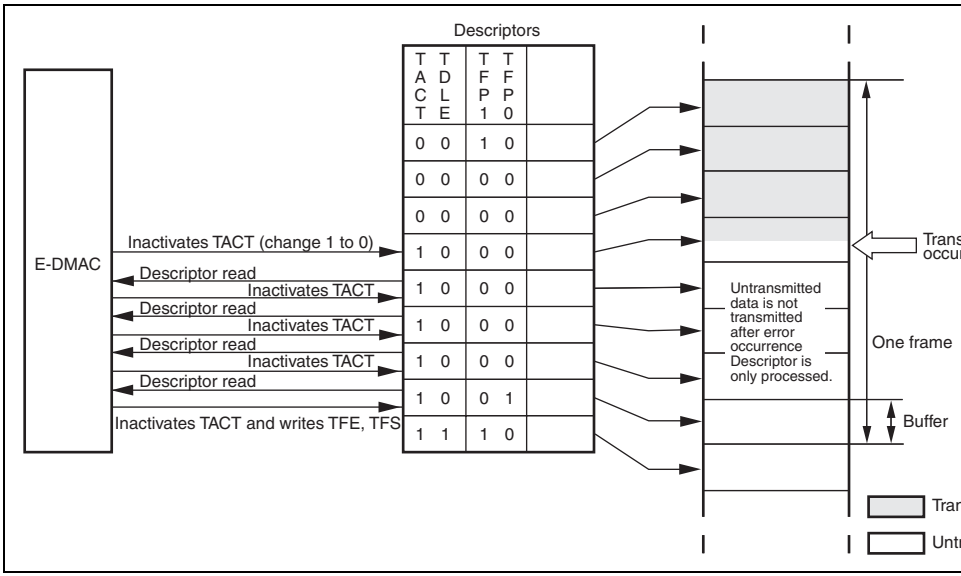


Figure 21.6 E-DMAC Operation after Transmit Error

If error interrupts are enabled in the EtherC/E-DMAC status interrupt permission register (EESIPR), an interrupt is generated immediately after the write-back. If there is a new fr receive request, reception is continued from the buffer after that in which the error occurr

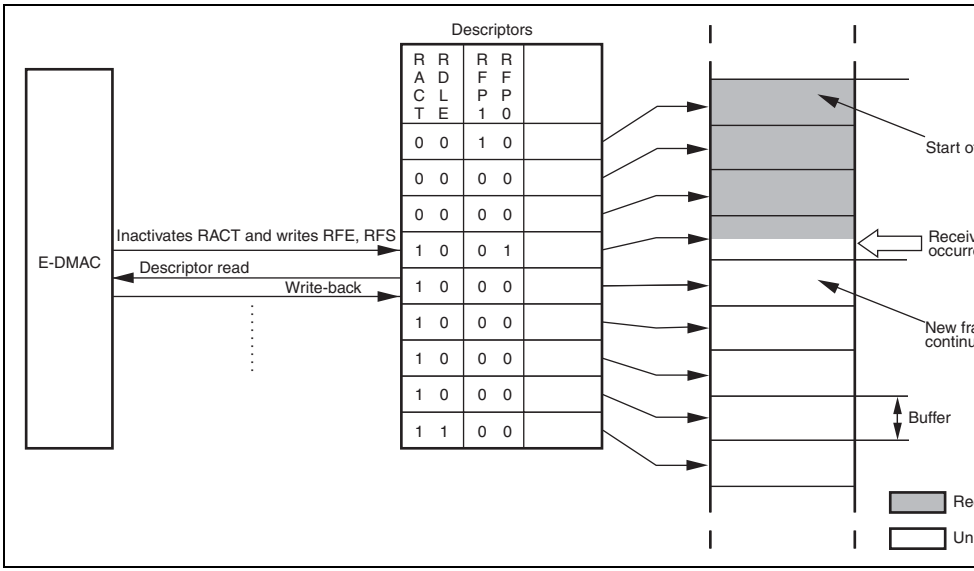
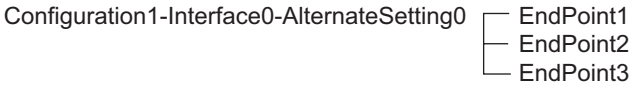


Figure 21.7 E-DMAC Operation after Receive Error

- Transfer speed: Supports full-speed (12 Mbps)
- Endpoint configuration:

Endpoint Name	Abbreviation	Transfer Type	Maximum Packet Size	FIFO Buffer Capacity (Byte)	DTC Tr
Endpoint 0	EP0s	Setup	8	8	—
	EP0i	Control-in	8	8	—
	EP0o	Control-out	8	8	—
Endpoint 1	EP1	Bulk-out	64	128	Possible
Endpoint 2	EP2	Bulk-in	64	128	Possible
Endpoint 3	EP3	Interrupt-in	8	8	—



- Interrupt requests: Generates various interrupt signals necessary for USB transmission/reception
- Power mode: Self-powered mode

Note: The EtherC operates only in high-speed mode

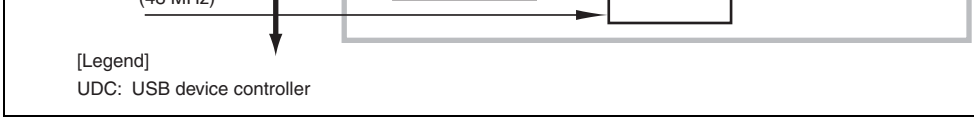


Figure 22.1 Block Diagram of USB

22.2 Input/Output Pins

Table 22.1 shows the USB pin configuration.

Table 22.1 Pin Configuration

Pin Name	I/O	Function
VBUS	Input	USB cable connection monitor pin
USD+	I/O	USB data I/O pin
USD-	I/O	USB data I/O pin
DrVcc	Input	Power supply pin for USB built-in transceiver
DrVss	Input	Ground pin for USB built-in transceiver
PUPDPLS	Output	Pull-up control pin
UXTAL	Input	USB clock pin
UEXTAL	Input	USB clock pin
UXSEL	Input	USB clock select pin

- Interrupt enable register 0 (IER0)
- Interrupt enable register 1 (IER1)
- Interrupt enable register 2 (IER2)
- EP0i data register (EPDR0i)
- EP0o data register (EPDR0o)
- EP0s data register (EPDR0s)
- EP1 data register (EPDR1)
- EP2 data register (EPDR2)
- EP3 data register (EPDR3)
- EP0o receive data size register (EPSZ0o)
- EP1 receive data size register (EPSZ1)
- Trigger register (TRG)
- Data status register (DASTS)
- FIFO clear register (FCLR)
- DTC transfer setting register (DMA)
- Endpoint stall register (EPSTL)
- Configuration value register (CVR)
- Control register (CTLR)
- Endpoint information register (EPIR)
- Transceiver test register 0 (TRNTREG0)
- Transceiver test register 1 (TRNTREG1)

				This bit is set to 1 when a bus reset signal is det the USB bus.
6	EP1FULL	0	R/W	EP1 FIFO Full [Reading] This bit is set when endpoint 1 receives one pac data successfully from the host, and holds a valu as long as there is valid data in the FIFO buffer. cannot be cleared. [Writing] When the data in endpoint 1 is transferred by the writing 0 to this bit clears the request for a DTC end interrupt. If the DTC transfer is not used, alw write 1 to this bit.
5	EP2TR	0	R/W	EP2 Transfer Request This bit is set if there is no valid transmit data in buffer when an IN token for endpoint 2 is receive the host. A NAK handshake is returned to the ho data is written to the FIFO buffer and packet transmission is enabled.

				write 1 to this bit.
3	SETUPTS	0	R/W	Setup Command Receive Complete This bit is set to 1 when endpoint 0 receives su a setup command requiring decoding on the ap side, and returns an ACK handshake to the hos
2	EP0oTS	0	R/W	EP0o Receive Complete This bit is set to 1 when endpoint 0 receives da the host successfully, stores the data in the FIF and returns an ACK handshake to the host.
1	EP0iTR	0	R/W	EP0i Transfer Request This bit is set if there is no valid transmit data in buffer when an IN token for endpoint 0 is receiv the host. A NAK handshake is returned to the h data is written to the FIFO buffer and packet transmission is enabled.
0	EP0iTS	0	R/W	EP0i Transmit Complete This bit is set when data is transmitted to the h endpoint 0 and an ACK handshake is returned.

				These bits are always read as 0. The initial val should not be changed.
3	VBUS MN	0	R	This is a status bit which monitors the state of t VBUS pin. This bit reflects the state of the VBUS pin and generates no interrupt request. This bit is alwa when the PULLUP_E bit in DMA is 0.
2	EP3 TR	0	R/W	EP3 Transfer Request This bit is set if there is no valid transmit data i FIFO buffer when an IN token for endpoint 3 is received from the host. A NAK handshake is re to the host until data is written to the FIFO buff packet transmission is enabled.
1	EP3 TS	0	R/W	EP3 Transmit Complete This bit is set when data is transmitted to the h endpoint 3 and an ACK handshake is returned
0	VBUSF	0	R/W	USB Disconnection Detection When the function is connected to the USB bus disconnected from it, this bit is set to 1. The VB of this module is used for detecting connection disconnection.

				should not be changed.
5	SURSS	0	R	Suspend/Resume Status This is a status bit that describes bus state. 0: Normal state 1: Suspended state This bit is a status bit and generates no interr request.
4	SURSF	0	R/W	Suspend/Resume Detection This bit is set to 1 when the state changed fro to suspended state or vice versa. The corresp interrupt output is RESUME, USBINTN2, and USBINTN3.
3	CFDN	0	R/W	End Point Information Load End This bit is set to 1 when writing data in the en information register to the EPIR register ends end). This module starts the USB operation a endpoint information is completely set.
2	SOF	0	R	SOF Interrupt Detection This bit is set to 1 when an SOF interrupt is d
1	SETC	0	R/W	Set_Configuration Command Detection When the Set_Configuration command is det bit is set to 1.
0	SETI	0	R/W	Set_Interface Command Detection When the Set_Interface command is detected is set to 1.

5	EP2 TR	0	R/W	EP2 Transfer Request
4	EP2 EMPTY	0	R/W	EP2 FIFO Empty
3	SETUP TS	0	R/W	Setup Command Receive Complete
2	EP0o TS	0	R/W	EP0o Receive Complete
1	EP0i TR	0	R/W	EP0i Transfer Request
0	EP0i TS	0	R/W	EP0i Transmission Complete

				These bits are always read as 0. The initial va should not be changed.
2	EP3 TR	1	R/W	EP3 Transfer Request
1	EP3 TS	1	R/W	EP3 Transmission Complete
0	VBUSF	1	R/W	USB Bus Connect

22.3.6 Interrupt Select Register 2 (ISR2)

ISR2 selects the vector numbers of the interrupt requests indicated in interrupt flag regis (IFR2). If the USB issues an interrupt request to the INTC when a bit in ISR0 is cleared interrupt corresponding to the bit will be USBINTN2. If the USB issues an interrupt req INTC when a bit in ISR0 is set to 1, the corresponding interrupt will be USBINTN3.

Bit	Bit Name	Initial Value	R/W	Description
7 to 5	—	All 0	R	Reserved These bits are always read as 0. The initial va should not be changed.
4	SURSE	1	R/W	Suspend/Resume Detection
3	CFDN	1	R/W	End Point Information Load End
2	SOF	1	R/W	SOF Interrupt Detection
1	SETCE	1	R/W	Set_Configuration Command Detection
0	SETIE	1	R/W	Set_Interface Command Detection

6	EP1 FULL	0	R/W	EP1 FIFO Full
5	EP2 TR	0	R/W	EP2 Transfer Request
4	EP2 EMPTY	0	R/W	EP2 FIFO Empty
3	SETUP TS	0	R/W	Setup Command Receive Complete
2	EP0o TS	0	R/W	EP0o Receive Complete
1	EP0i TR	0	R/W	EP0i Transfer Request
0	EP0i TS	0	R/W	EP0i Transmission Complete

22.3.8 Interrupt Enable Register 1 (IER1)

IER1 enables the interrupt requests of interrupt flag register 1 (IFR1). When an interrupt
to 1 while the corresponding bit of each interrupt is set to 1, an interrupt request is sent to
CPU. The interrupt vector number is determined by the contents of interrupt select registe
(ISR1).

Bit	Bit Name	Initial Value	R/W	Description
7 to 3	—	All 0	R	Reserved These bits are always read as 0. The initial val should not be changed.
2	EP3 TR	0	R/W	EP3 Transfer Request
1	EP3 TS	0	R/W	EP3 Transmission Complete
0	VBUSF	0	R/W	USB Bus Connect

				Suspend and Resume Operations.
6, 5	—	All 0	R	Reserved These bits are always read as 0. The initial va should not be changed.
4	SURSE	0	R/W	Suspend/Resume Detection For the details of the operation, see section 2 Suspend and Resume Operations.
3	CFDN	0	R/W	End Point Information Load End
2	SOF	0	R/W	SOF Interrupt Detection
1	SETCE	0	R/W	Set_Configuration Command Detection
0	SETIE	0	R/W	Set_Interface Command Detection

22.3.10 EP0i Data Register (EPDR0i)

EPDR0i is an 8-byte transmit FIFO buffer for endpoint 0. EPDR0i holds one packet of t data for control-in. Transmit data is fixed by writing one packet of data and setting EP0i the trigger register. When an ACK handshake is returned from the host after the data has transmitted, EP0iTS in interrupt flag register 0 is set. This FIFO buffer can be initialized of EP0iCLR in the FCLR register.

Bit	Bit Name	Initial Value	R/W	Description
7 to 0	D7 to D0	Undefined	W	Data register for control-in transfer

22.3.12 EP0s Data Register (EPDR0s)

EPDR0s is an 8-byte FIFO buffer specifically for receiving endpoint 0 setup commands. setup command to be processed by the application is received. When command data is re successfully, the SETUPTS bit in interrupt flag register 0 is set.

As a latest setup command must be received in high priority, if data is left in this buffer, i overwritten with new data. If reception of the next command is started while the current c is being read, command reception has priority, the read by the application is forcibly stop the read data is invalid.

Bit	Bit Name	Initial Value	R/W	Description
7 to 0	D7 to D0	All 0	R	Data register for storing the setup command at control-out transfer

Bit	Bit Name	Value	R/W	Description
7 to 0	D7 to D0	All 0	R	Data register for endpoint 1 transfer

22.3.14 EP2 Data Register (EPDR2)

EPDR2 is a 128-byte transmit FIFO buffer for endpoint 2. EPDR2 has a dual-buffer con
and has a capacity of twice the maximum packet size. When transmit data is written to t
buffer and EP2PKTE in the trigger register is set, one packet of transmit data is fixed, ar
dual-FIFO buffer is switched over. The transmit data for this FIFO buffer can be transfe
DTC. This FIFO buffer can be initialized by means of EP2CLR in the FCLR register.

Bit	Bit Name	Initial Value	R/W	Description
7 to 0	D7 to D0	Undefined	W	Data register for endpoint 2 transfer

22.3.15 EP3 Data Register (EPDR3)

EPDR3 is an 8-byte transmit FIFO buffer for endpoint 3. EPDR3 holds one packet of tra
for the interrupt transfer of endpoint 3. Transmit data is fixed by writing one packet of d
setting EP3PKTE in the trigger register. When an ACK handshake is returned from the h
one packet of data has been transmitted successfully, EP3TS in interrupt flag register 0 i
FIFO buffer can be initialized by means of EP3CLR in the FCLR register.

Bit	Bit Name	Initial Value	R/W	Description
7 to 0	D7 to D0	Undefined	W	Data register for endpoint 3 transfer

EP3Z1 is a receive data size resister for endpoint 1. EP3Z1 indicates the number of bytes from the host. The FIFO for endpoint 1 has a dual-buffer configuration. The size of the re data indicated by this register is the size of the currently selected side (can be read by CP

Bit	Bit Name	Initial Value	R/W	Description
7 to 0	—	All 0	R	Number of received bytes for endpoint 1

22.3.18 Trigger Register (TRG)

TRG generates one-shot triggers to control the transfer sequence for each endpoint.

Bit	Bit Name	Initial Value	R/W	Description
7	—	Undefined	—	Reserved The initial value should not be changed.
6	EP3 PKTE	Undefined	W	EP3 Packet Enable After one packet of data has been written to t endpoint 3 transmit FIFO buffer, the transmit fixed by writing 1 to this bit.
5	EP1 RDFN	Undefined	W	EP1 Read Complete Write 1 to this bit after one packet of data has read from the endpoint 1 FIFO buffer. The en receive FIFO buffer has a dual-buffer configu Writing 1 to this bit initializes the FIFO that wa enabling the next packet to be received.

				transfer of data in the following data stage. A handshake is returned in response to transfe requests from the host in the data stage unti written to this bit.
1	EP0o RDFN	Undefined	W	EP0o Read Complete Writing 1 to this bit after one packet of data h read from the endpoint 0 transmit FIFO buffe initializes the FIFO buffer, enabling the next be received.
0	EP0i PKTE	Undefined	W	EP0i Packet Enable After one packet of data has been written to endpoint 0 transmit FIFO buffer, the transmit fixed by writing 1 to this bit.

				should not be changed.
5	EP3 DE	0	R	EP3 Data Present This bit is set when the endpoint 3 FIFO buffe contains valid data.
4	EP2 DE	0	R	EP2 Data Present This bit is set when the endpoint 2 FIFO buffe contains valid data.
3 to 1	—	All 0	R	Reserved These bits are always read as 0.
0	EP0i DE	0	R	EP0i Data Present This bit is set when the endpoint 0 FIFO buffe contains valid data.

6	EP3 CLR	Undefined	W	EP3 Clear Writing 1 to this bit initializes the endpoint 3 FIFO buffer.
5	EP1 CLR	Undefined	W	EP1 Clear Writing 1 to this bit initializes both sides of th endpoint 1 receive FIFO buffer.
4	EP2 CLR	Undefined	W	EP2 Clear Writing 1 to this bit initializes both sides of th endpoint 2 transmit FIFO buffer.
3 to 1	—	All 0	R	Reserved The initial value should not be changed.
0	EP0i CLR	Undefined	W	EP0i Clear Writing 1 to this bit initializes the endpoint 0 FIFO buffer.

				These bits are always read as 0. The initial va should not be changed.
2	PULLUP_E	0	R/W	PULLUP Enable This pin performs pull-up control for the D+ pi use of PUPDPLS as the pull-up control pin. 0: D+ pull-up is disabled. (The PULLUP pin is driven low.) 1: D+ pull-up is enabled. (The PULLUP pin is driven high.)

(USBINTN1) is asserted again. However, if t
the data packet to be transmitted is less thar
bytes, the EP2 packet enable bit is not set
automatically, and so should be set by the C
DTC transfer end interrupt.

As EP2-related interrupt requests to the CPU
automatically masked, interrupt requests sho
masked as necessary in the interrupt enable

- Operating procedure

1. Set the number of transfers in the DTC.
2. Set the DTC to be activated by USBINTN
3. Write 1 to this bit.
4. Activate the DTC.
5. DTC transfer is performed.
6. DTC transfer end interrupt is generated.
7. Write 0 to the EP1DMAE bit in DMA.
8. Write 0 to the EP1FULL bit in IFR0.

See section 22.8.3, DTC Transfer for Endpo

2. Related interrupt requests to the CPU are automatically masked.

- Operating procedure:

1. Set the number of transfers in the DTC.
2. Set the DTC to be activated by USBINTN
3. Write 1 to this bit.
4. Activate the DTC.
5. DTC transfer is performed.
6. DTC transfer end interrupt is generated.
7. Write 0 to the EP2DMAE bit in DMA.
8. Write 0 to the EP2EMPTY bit in IFR0.

See section 22.8.2, DTC Transfer for Endpoi

7 to 4	—	All 0	R	Reserved These bits are always read as 0. The initial v should not be changed.
3	EP3STL	0	R/W	EP3 Stall When this bit is set to 1, endpoint 3 is placed stall state.
2	EP2STL	0	R/W	EP2 Stall When this bit is set to 1, endpoint 2 is placed stall state.
1	EP1STL	0	R/W	EP1 Stall When this bit is set to 1, endpoint 1 is placed stall state.
0	EP0STL	0	R/W	EP0 Stall When this bit is set to 1, endpoint 0 is placed stall state.

4	INTV0			when the SETI bit in IFR2 is set to 1.
3	—	0	R	Reserved This bit is always read as 0. The initial value s not be changed.
2	ALTV2	0	R	These bits store Alternate Setting value when receive Set Interface command. ALTV2 to AL updated when the SETI bit in IFR2 is set to 1.
1	ALTV1	0	R	
0	ALTV0	0	R	

22.3.24 Control Register (CTLR)

This register sets functions by setting bits ASCE, RSME, and, RWUPS.

Bit	Bit Name	Initial Value	R/W	Description
7 to 5	—	All 0	R	Reserved These bits are always read as 0. The initial va should not be changed.
4	RWUPS	0	R	Remote Wakeup Status This status bit indicates remote wakeup comm from USB host is enabled or disabled. This bit is set to 0 when remote wakeup comm from UBM host is disabled by Device_Remote_Wakeup due to Set Feature Feature request. This bit is set to 1 when rem wakeup command is enabled.

1	ASCE	0	R/W	Automatic Stall Clear Enable Setting the ASCE bit to 1 automatically clear setting bit (the EPxSTL (x = 1, 2, or 3) bit in or EPSTR1) of the end point that has returne stall handshake to the host. The automatic s enable is common to the all end points. Thus individual control of the end point is not poss When the ASCE bit is set to 0, the stall settin not automatically cleared. This bit must be re by the users. To enable this bit, make sure th ASCE bit should be set to 1 before the EPxS 2, or 3) bit in EPSTL is set to 1.
0	—	0	R	Reserved This bit is always read as 0. The initial value not be changed.

- EPIR00

Bit	Bit Name	Initial Value	R/W	Description
7 to 4	D7 to D4	Undefined	W	Endpoint Number [Enable setting range] 0 to 3
3, 2	D3, D2	Undefined	W	Endpoint Configuration Number [Enable setting range] 0 or 1
1, 0	D1, D0	Undefined	W	Endpoint Interface Number [Enable setting range] 0 to 3

				2: Bulk 3: Interrupt
3	D3	Undefined	W	Endpoint Transmission Direction [Possible setting range] 0: Out 1: In
2 to 0	D2 to D0	Undefined	W	Reserved [Possible setting range] Fixed to 0.

- EPIR02

Bit	Bit Name	Initial Value	R/W	Description
7 to 1	D7 to D1	Undefined	W	Endpoint Maximum Packet Size [Possible setting range] 0 to 64
0	D0	Undefined	W	Reserved [Possible setting range] Fixed to 0.

- EPIR03

Bit	Bit Name	Initial Value	R/W	Description
7 to 0	D7 to D0	Undefined	W	Reserved [Possible setting range] Fixed to 0.

described below.

Since each endpoint FIFO number is optimized by the exclusive software that correspond
transfer system, direction, and the maximum packet size, make sure to set the endpoint FI
number to the data described in table 22.2.

1. The endpoint FIFO number 1 cannot designate other than the maximum packed size c
   control transfer method, and out transfer direction.
2. The endpoint number 0 and the endpoint FIFO number must have one-on one relation
3. The maximum packet size for the endpoint FIFO number 0 is 8 bytes only.
4. The endpoint FIFO number 0 can specify only the maximum packet size and the data
   rest should be all 0.
5. The maximum packet size for the endpoint FIFO numbers 1 and 2 is limited to 64 byt
6. The maximum packet size for the endpoint FIFO numbers 3 is limited to 8 bytes.
7. The maximum number of endpoint information setting is ten.
8. Up to ten endpoint information setting should be made.
9. Write 0 to the endpoints not in use.

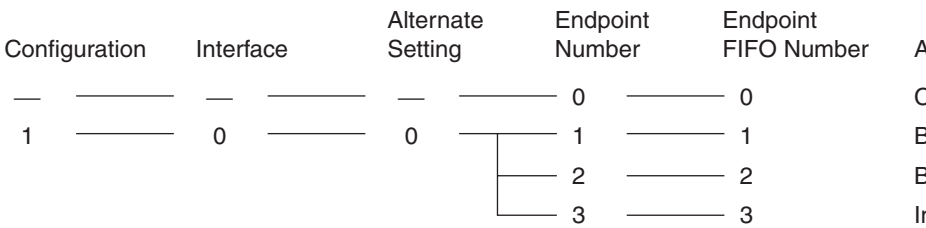
Table 22.2 shows the example of limitations for the maximum packet size, the transfer m
and the transfer direction.

Table 22.2 Example of Limitations for Setting Values

Endpoint FIFO Number	Maximum Packet Size	Transfer Method	Transfer Dire
0	8 bytes	Control	—
1	64 bytes	Bulk	Out
2	64 bytes	Bulk	In
3	8 bytes	Interrupt	In

—	1	1	1	—	—	—	—

N	EPIR[N]0	EPIR[N]1	EPIR[N]2	EPIR[N]3	EPIR[
0	00	00	10	00	00
1	14	20	80	00	01
2	24	28	80	00	02
3	34	38	10	00	03
4	00	00	00	00	00
5	00	00	00	00	00
6	00	00	00	00	00
7	00	00	00	00	00
8	00	00	00	00	00
9	00	00	00	00	00



6 to 4	—	All 0	R	Reserved These bits are always read as 0. The initial va should not be changed.
3	SUSPEND	0	R/W	Built-In Transceiver Output Signal Setting
2	txenl	0	R/W	SUSPEND: Sets the (SUSPEND) signal of th transceiver.
1	txse0	0	R/W	
0	txdata	0	R/W	txenl: Sets the output enable (txenl) sig built-in transceiver.
				txse0: Sets the Signal-ended 0 (txse0) s the built-in transceiver.
				txdata: Sets the (txdata) signal of the bui transceiver.

1	1	1	X	X	Hi-Z	Hi-Z

[Legend]

X: Don't care.

—: Cannot be controlled. Indicates state in normal operation according to the USB o and port settings.

22.3.27 Transceiver Test Register 1 (TRNTREG1)

TRNTREG1 is a test register that can monitor the built-in transceiver input signal.

Setting bits PTSTE and txenl in TRNTREG0 to 1 enables monitoring the built-in transc signal. Table 22.5 shows the relationship between pin input and TRNTREG1 monitorin

Bit	Bit Name	Initial Value	R/W	Description
7 to 3	—	All 0	R	Reserved These bits are always read as 0. The initial va should not be changed.
2	xver_data	—*	R	Built-In Transceiver Input Signal Monitor
1	dpls	—*	R	xver_data: Monitors the differential input level (xver_data) signal of the built-in tra
0	dmns	—*	R	dpls: Monitors the USD+ (dpls) signal of in transceiver.
				dmns: Monitors the USD- (dmns) signal o in transceiver.

Note: * Determined by the state of pins, VBUS, USD+, and USD-

1	0	1	1	0	1	1	0	
1	0	1	1	1	X	1	1	
1	1	1	0	0	0	0	0	
1	1	1	0	1	0	0	1	
1	1	1	1	0	0	1	0	
1	1	1	1	1	0	1	1	
1	X	0	X	X	0	1	1	Can be mo when VBU

[Legend]

X: Don't care.

		(EP0)				
	1		EP0i_TR*	EP0i transfer request	USBINTN2 or USBINTN3	×
	2		EP0o_TS*	EP0o receive complete	USBINTN2 or USBINTN3	×
	3		SETUP_TS*	Setup command receive complete	USBINTN2 or USBINTN3	×
	4	Bulk_in transfer (EP2)	EP2_EMPTY	EP2 FIFO empty	USBINTN2 or USBINTN3	USBINTN
	5		EP2_TR	EP2 transfer request	USBINTN2 or USBINTN3	×
	6	Bulk_out transfer (EP1)	EP1_FULL	EP1 FIFO Full	USBINTN2 or USBINTN3	USBINTN
	7	Status	BRST	Bus reset	USBINTN2 or USBINTN3	×
IFR1	0	Status	VBUSF	USB disconnection detection	USBINTN2 or USBINTN3	×
	1	Interrupt_in transfer (EP3)	EP3_TS	EP3 transfer complete	USBINTN2 or USBINTN3	×
	2		EP3_TR	EP3 transfer request	USBINTN2 or USBINTN3	×
	3	Status	VBUSMN	VBUS connection status	—	×
	4	—	Reserved	—	—	—
	5					
	6					
	7					

				detection	USBINTN3, or RESUME	
	5		SURSS	Suspend/resume status	—	×
	6	—	Reserved	—	—	—
	7					

Note: * EP0 interrupts must be assigned to the same interrupt request signal.

- USBINTN0 signal

  DTC start interrupt signal only EP1. See section 22.8, DTC Transfer.
- USBINTN1 signal

  DTC start interrupt signal only EP1. See section 22.8, DTC Transfer.
- USBINTN2 signal

  The USBINTN2 signal requests interrupt sources for which the corresponding bits in select registers 0 to 2 (ISR0 to ISR2) are cleared to 0. The USBINTN2 is driven low i corresponding bit in the interrupt flag register is set to 1.
- USBINTN3 signal

  The USBINTN3 signal requests interrupt sources for which the corresponding bits in select registers 0 to 2 (ISR0 to ISR2) are cleared to 0. The USBINTN3 is driven low i corresponding bit in the interrupt flag register is set to 1.
- RESUME signal

  The RESUME signal is a resume interrupt signal for canceling software standby mod RESUME signal is driven low at the transition to the resume state for canceling softw standby mode.

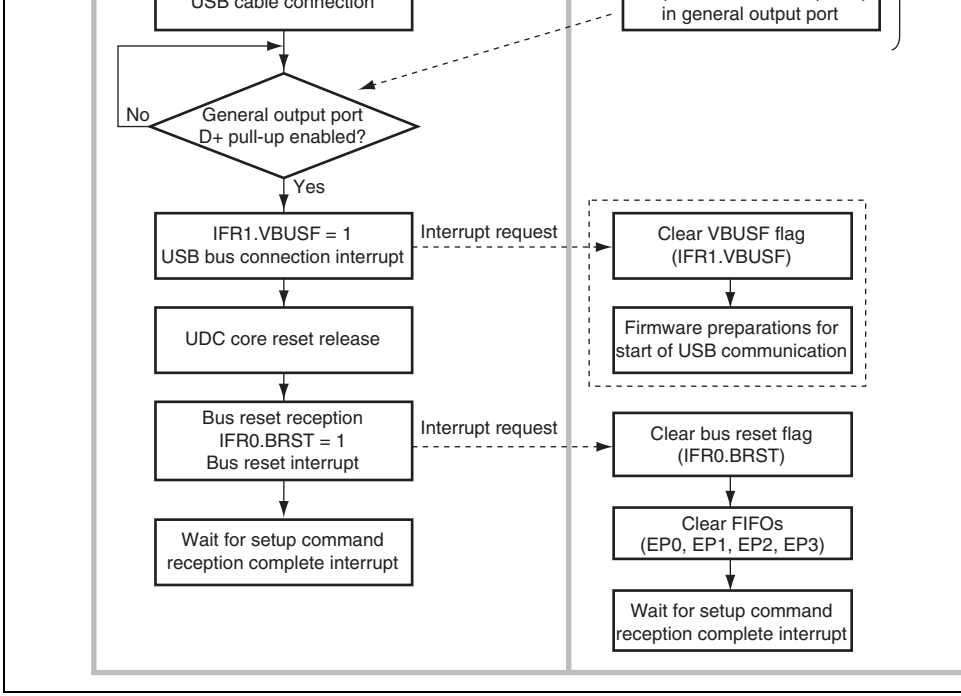


Figure 22.2 Operation at Cable Connection

The above flowchart shows the operation in the case of in section 22.9, Example of USB Circuitry.

In applications that do not require USB cable connection to be detected, processing by the bus connection interrupt is not necessary. Preparations should be made with the bus-reset interrupt.

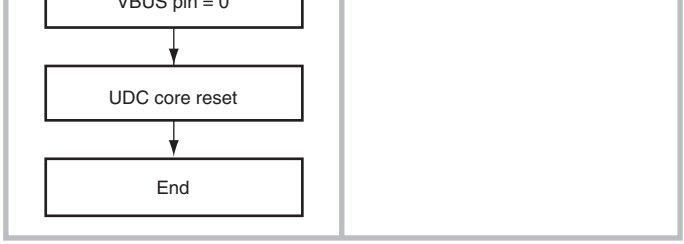


Figure 22.3 Operation at Cable Disconnection

The above flowchart shows the operation in section 22.9, Example of USB External Circu

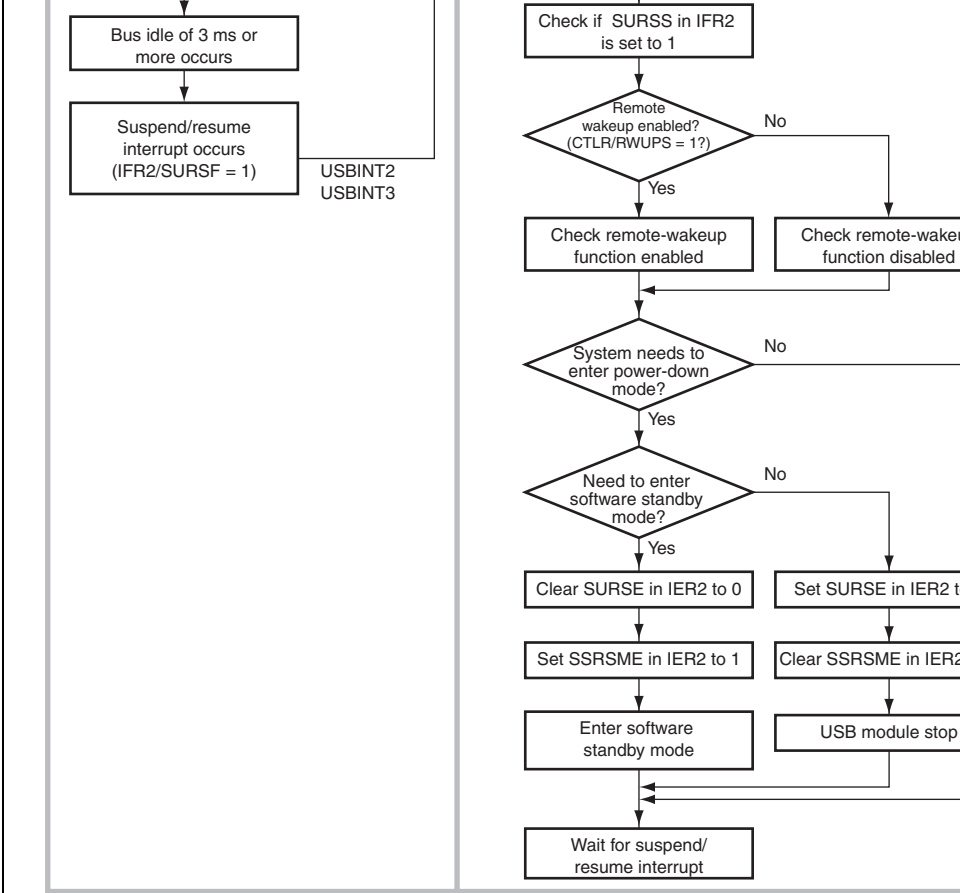


Figure 22.4 Suspend Operation

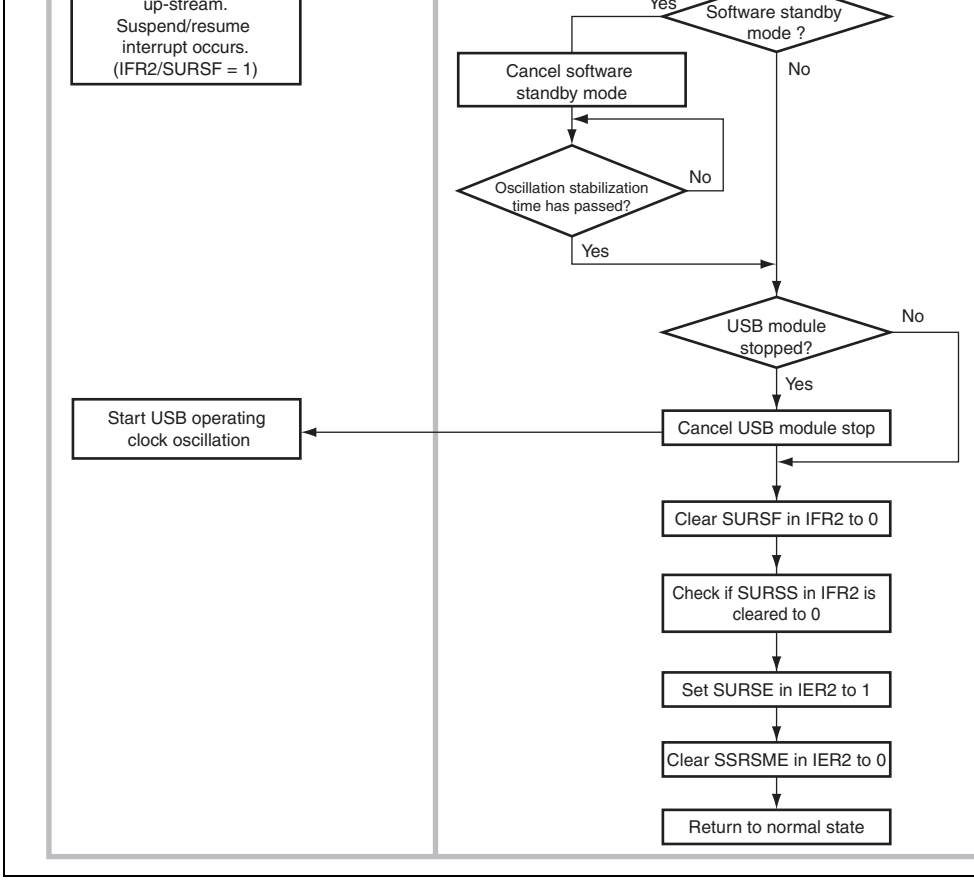


Figure 22.5 Resume Operation from Up-Stream

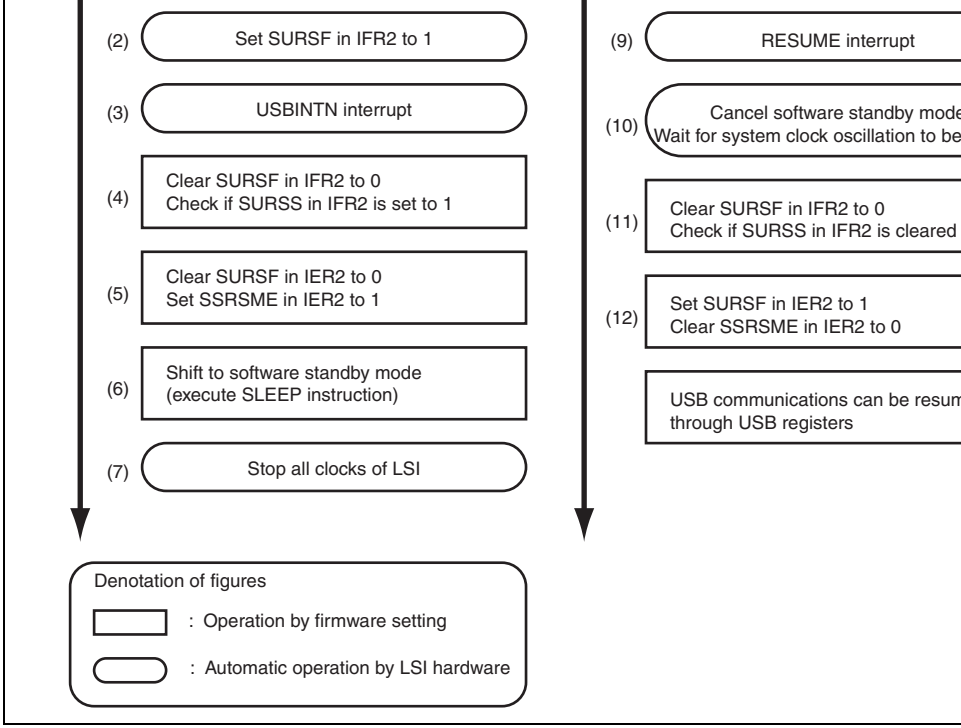


Figure 22.6 Flow of Transition to and Canceling Software Standby Mod

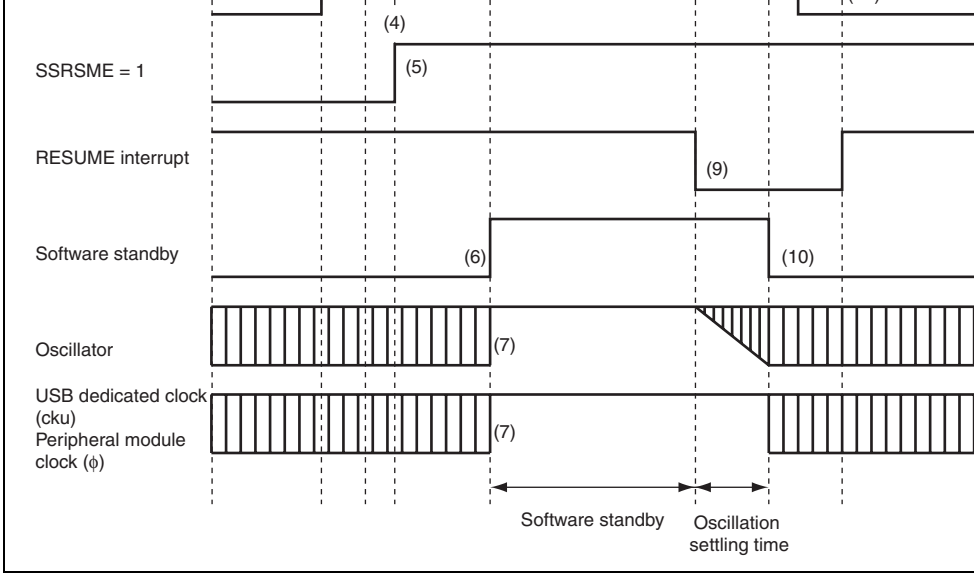


Figure 22.7 Timing of Transition to and Canceling Software Standby Mod

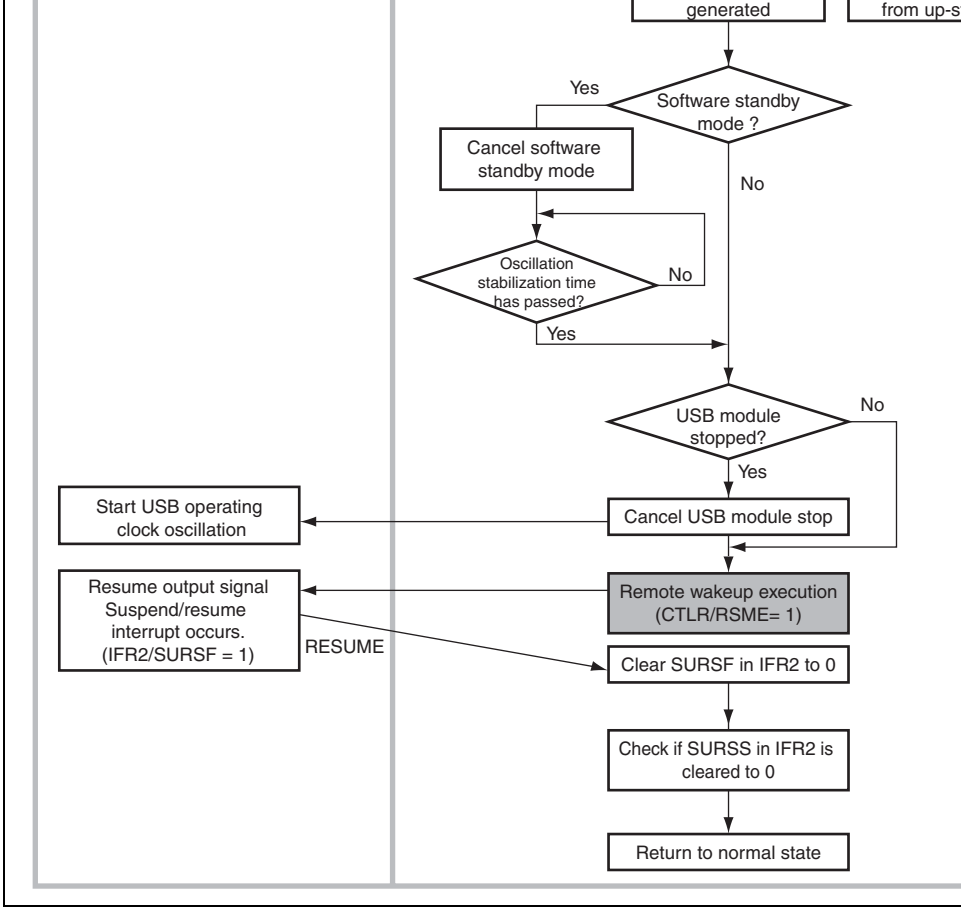


Figure 22.8 Remote-Wakeup

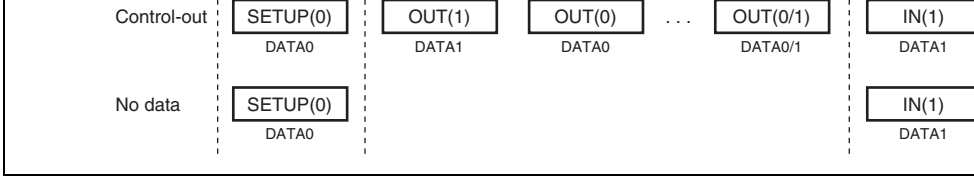
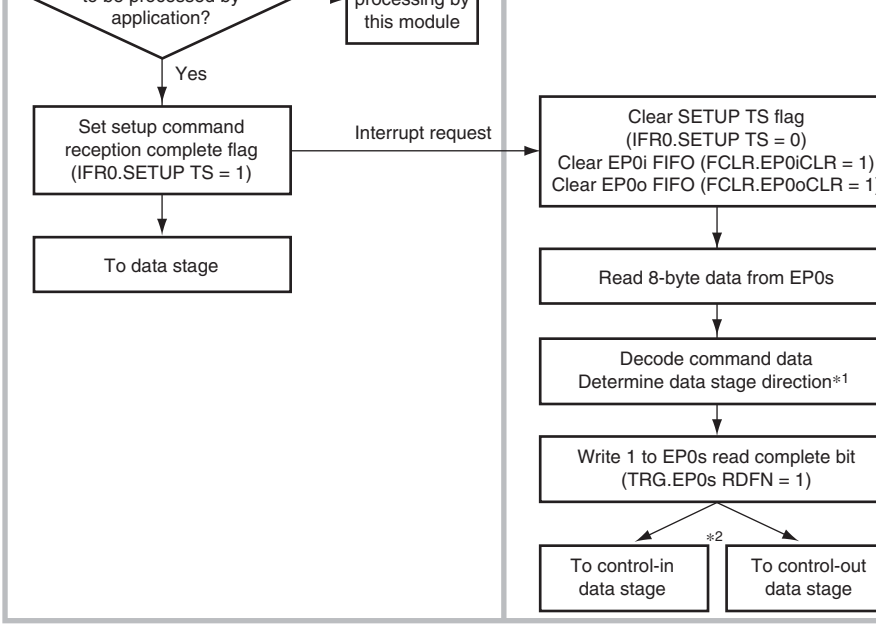


Figure 22.9 Transfer Stages in Control Transfer



Notes: 1. In the setup stage, the application analyzes command data from the host requiring processing
the application, and determines the subsequent processing (for example, data stage direction,
2. When the transfer direction is control-out, the EP0i transfer request interrupt required in the sta
stage should be enabled here. When the transfer direction is control-in, this interrupt is not req
and should be disabled.

Figure 22.10 Setup Stage Operation

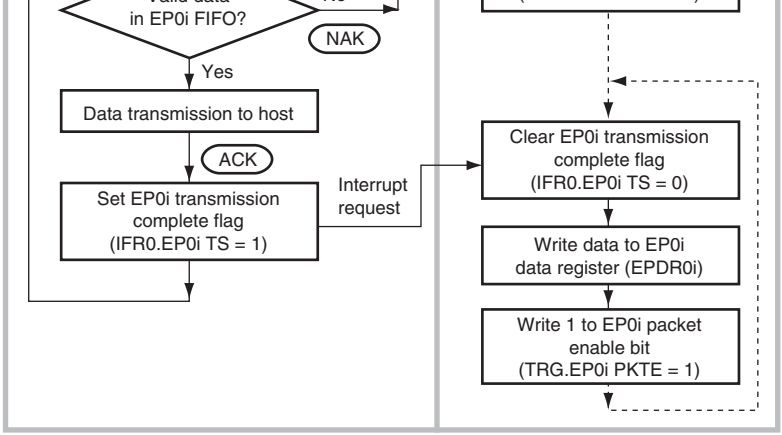


Figure 22.11 Data Stage (Control-In) Operation

The application first analyzes command data from the host in the setup stage, and determ subsequent data stage direction. If the result of command data analysis is that the data sta transfer, one packet of data to be sent to the host is written to the FIFO. If there is more d sent, this data is written to the FIFO after the data written first has been sent to the host (I bit in IFR0 = 1).

The end of the data stage is identified when the host transmits an OUT token and the statu is entered.

Note: If the size of the data transmitted by the function is smaller than the data size requ the host, the function indicates the end of the data stage by returning to the host a shorter than the maximum packet size. If the size of the data transmitted by the fu an integral multiple of the maximum packet size, the function indicates the end o stage by transmitting a zero-length packet.

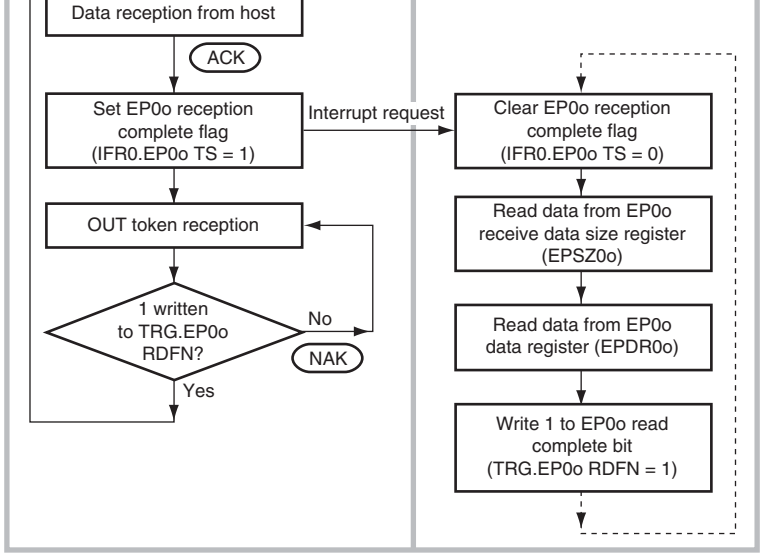


Figure 22.12 Data Stage (Control-Out) Operation

The application first analyzes command data from the host in the setup stage, and determ subsequent data stage direction. If the result of command data analysis is that the data st transfer, the application waits for data from the host, and after data is received (EP0oTS IFR0 = 1), reads data from the FIFO. Next, the application writes 1 to the EP0o read com empties the receive FIFO, and waits for reception of the next data.

The end of the data stage is identified when the host transmits an IN token and the status entered.

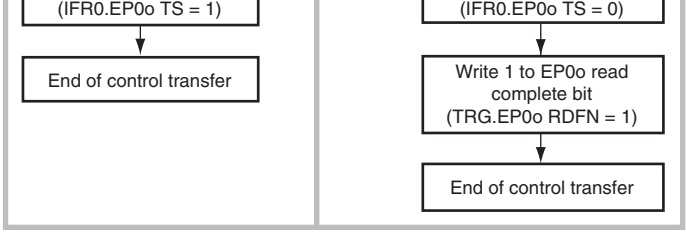


Figure 22.13 Status Stage (Control-In) Operation

The control-in status stage starts with an OUT token from the host. The application receiv
byte data from the host, and ends control transfer.

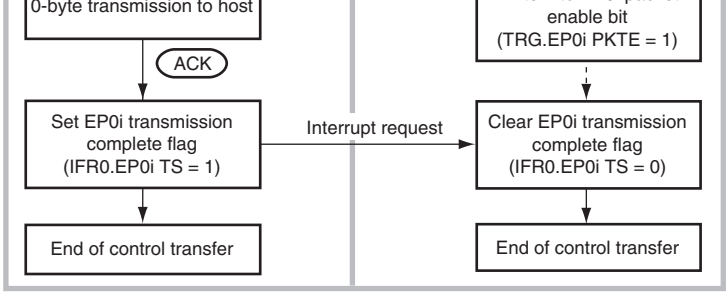


Figure 22.14 Status Stage (Control-Out) Operation

The control-out status stage starts with an IN token from the host. When an IN-token is
the start of the status stage, there is not yet any data in the EP0i FIFO, and so an EP0i tra
request interrupt is generated. The application recognizes from this interrupt that the stat
has started. Next, in order to transmit 0-byte data to the host, 1 is written to the EP0i pac
bit but no data is written to the EP0i FIFO. As a result, the next IN token causes 0-byte d
transmitted to the host, and control transfer ends.

After the application has finished all processing relating to the data stage, 1 should be w
the EP0i packet enable bit.

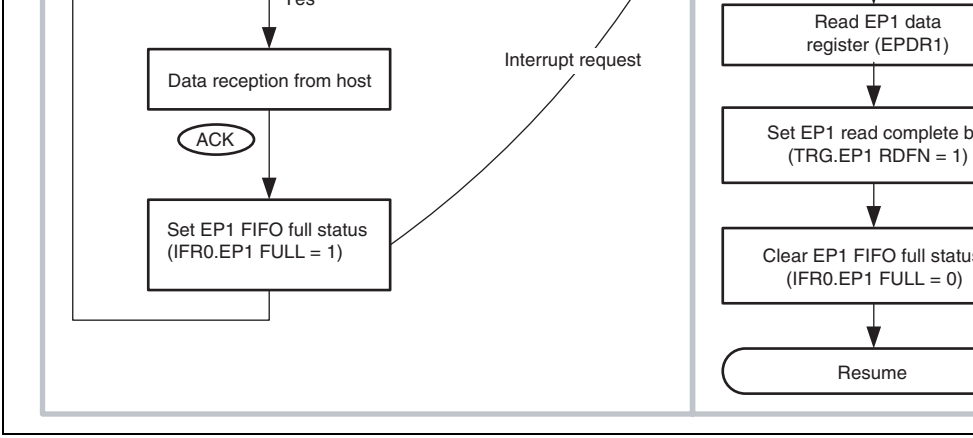


Figure 22.15 EP1 Bulk-Out Transfer Operation

EP1 has two 64-byte FIFOs, but the user can receive data and read receive data without b
aware of this dual-FIFO configuration.

When one FIFO is full after reception is completed, the EP1FULL bit in IFR0 is set. Afte
receive operation into one of the FIFOs when both FIFOs are empty, the other FIFO is en
so the next packet can be received immediately. When both FIFOs are full, NAK is return
host automatically. When reading of the receive data is completed following data receptio
written to the EP1RDFN bit in TRG and 0 is written to the EP1FULL bit in IFR0. This op
empties the FIFO that has just been read, and makes it ready to receive the next packet.

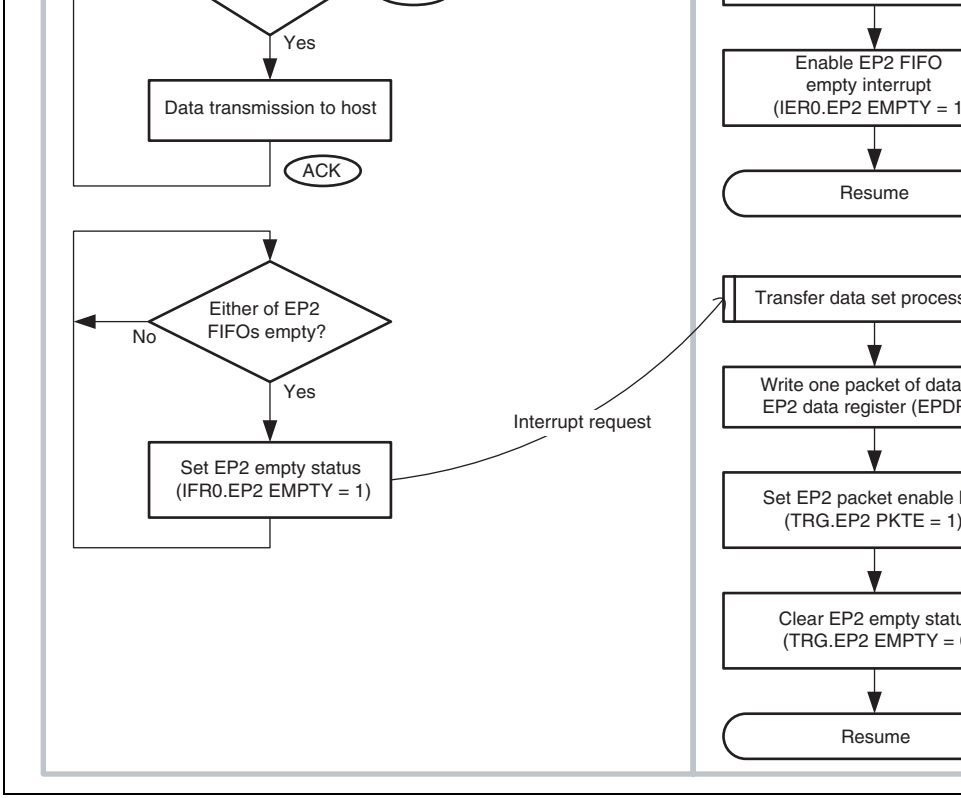
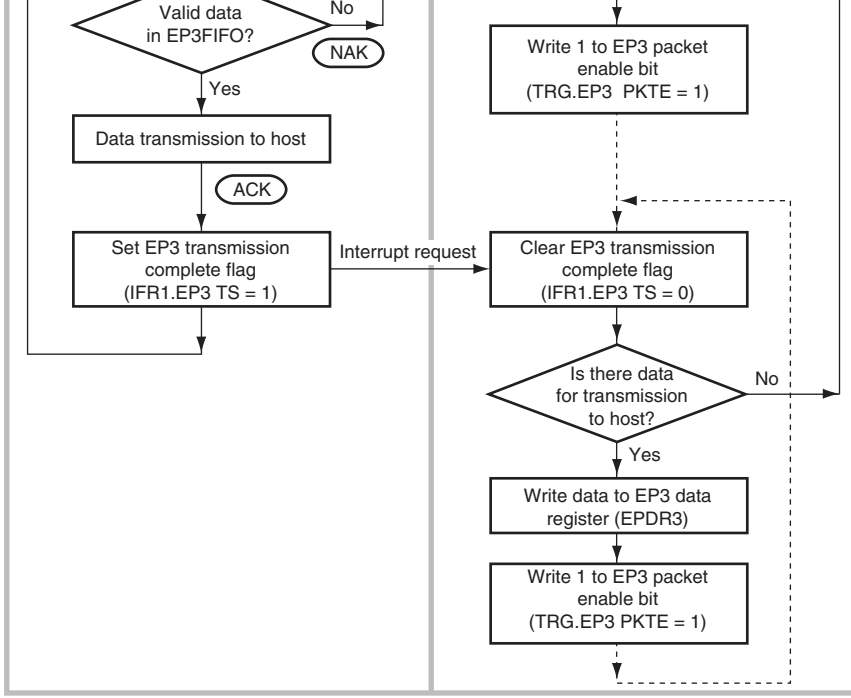


Figure 22.16 EP2 Bulk-In Transfer Operation

EP2 has two 64-byte FIFOs, but the user can transmit data and write transmit data witho
aware of this dual-FIFO configuration. However, one data write is performed for one FI
example, even if both FIFOs are empty, it is not possible to perform EP2PKTE at one ti
consecutively writing 128 bytes of data. EP2PKTE must be performed for each 64-byte

When transmission of all data has been completed, write 0 to the EP2EMPTY bit in IER0
disable interrupt requests.



Note: This flowchart shows just one example of interrupt transfer processing. Other possibilities include operation flow in which, if there is data to be transferred, the EP3 DE bit in the data status register referenced to confirm that the FIFO is empty, and then data is written to the FIFO.

Figure 22.17 Operation of EP3 Interrupt-In Transfer

Decoding not Necessary on Application Side	Decoding Necessary on Applicatio
Clear Feature	Get Descriptor
Get Configuration	Class/Vendor command
Get Interface	Set Descriptor
Get Status	Sync Frame
Set Address	
Set Configuration	
Set Feature	
Set Interface	

If decoding is not necessary on the application side, command decoding and data stage ar
stage processing are performed automatically. No processing is necessary by the user. An
is not generated in this case.

If decoding is necessary on the application side, this module stores the command in the E
FIFO. After reception is completed successfully, the IFR0/SETUP TS flag is set and an in
request is generated. In the interrupt routine, eight bytes of data must be read from the EP
register (EPDR0s) and decoded by firmware. The necessary data stage and status stage pr
should then be carried out according to the result of the decoding operation.

The USB function module has internal status bits that hold the status (stall or non-stall) endpoint. When a transaction is sent from the host, the module references these internal and determines whether to return a stall to the host. These bits cannot be cleared by the application; they must be cleared with a Clear Feature command from the host.

However, the internal status bit for EP0 is automatically cleared only when the setup co received.

22.7.2 Forcible Stall by Application

The application uses the EPSTL register to issue a stall request for the USB function mo When the application wishes to stall a specific endpoint, it sets the corresponding bit in 1 in figure 22.18). The internal status bits are not changed at this time. When a transacti from the host for the endpoint for which the EPSTL bit was set, the USB function modu references the internal status bit, and if this is not set, references the corresponding bit in (1-2 in figure 22.18). If the corresponding bit in EPSTL is set, the USB function module internal status bit and returns a stall handshake to the host (1-3 in figure 22.18). If the corresponding bit in EPSTL is not set, the internal status bit is not changed and the trans accepted.

Once an internal status bit is set, it remains set until cleared by a Clear Feature comman host, without regard to the EPSTL register. Even after a bit is cleared by the Clear Featu command (3-1 in figure 22.18), the USB function module continues to return a stall han while the bit in EPSTL is set, since the internal status bit is set each time a transaction is for the corresponding endpoint (1-2 in figure 22.18). To clear a stall, therefore, it is nece the corresponding bit in EPSTL to be cleared by the application, and also for the interna to be cleared with a Clear Feature command (2-1, 2-2, and 2-3 in figure 22.18).

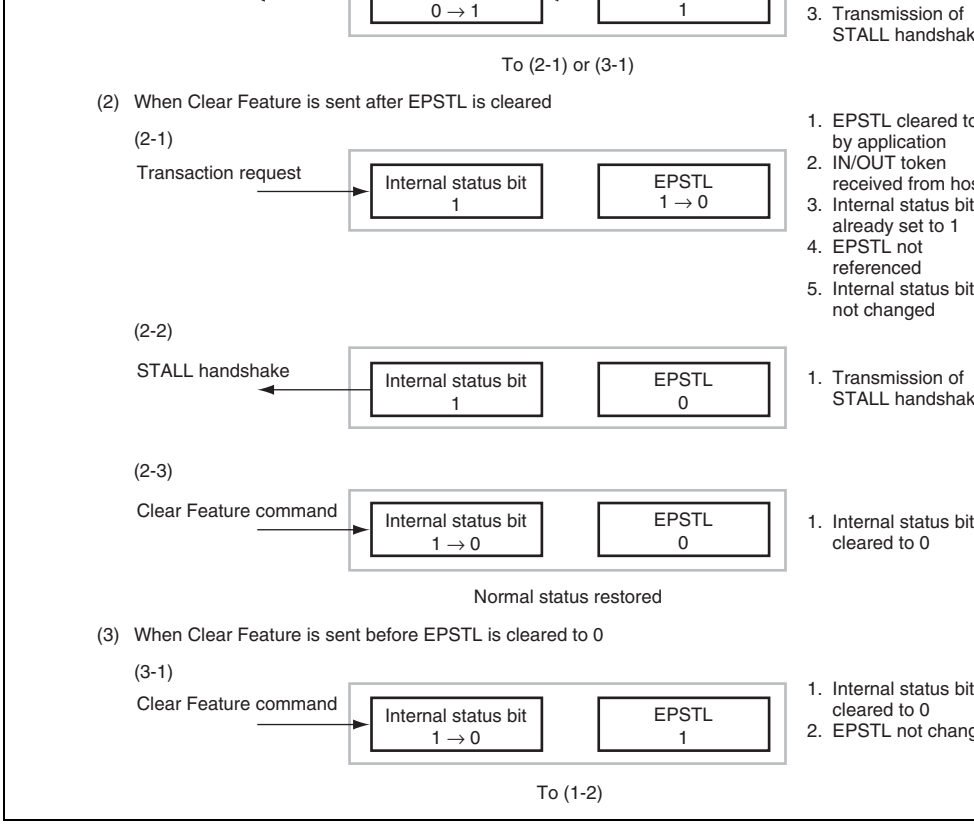


Figure 22.18 Forcible Stall by Application

the internal status bit must be cleared with a Clear Feature command (3-1 in figure 22.19
by the application, EPSTL should also be cleared (2-1 in figure 22.19).

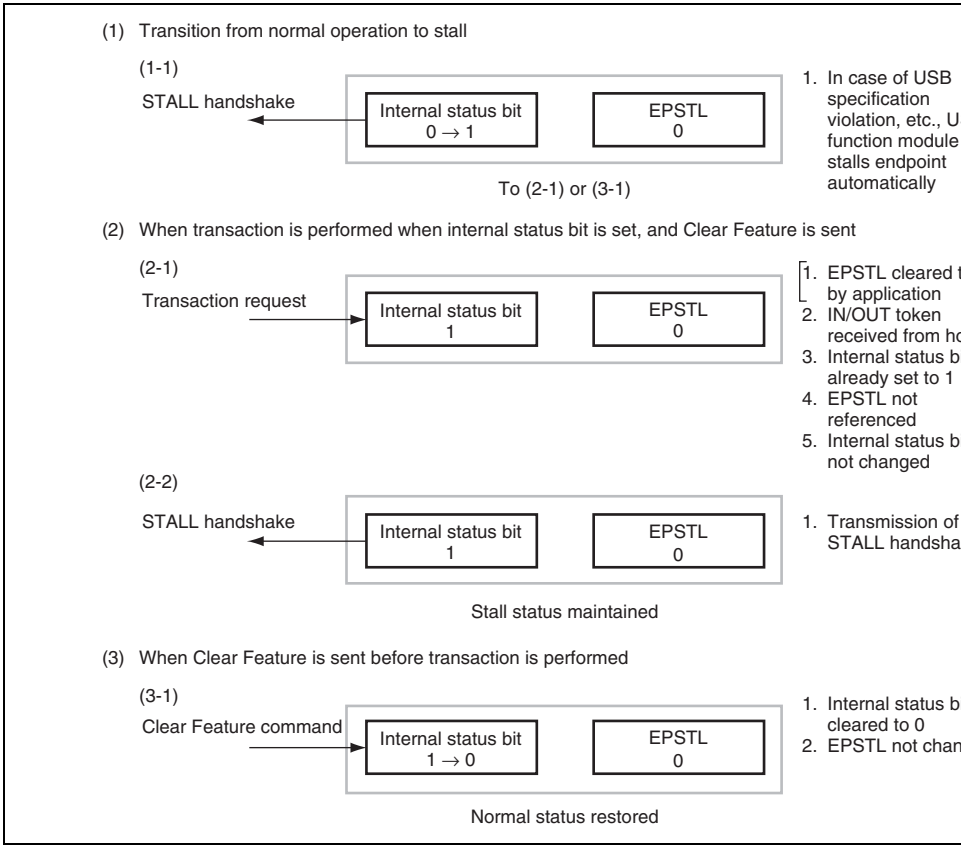


Figure 22.19 Automatic Stall by USB Function Module

If the DTC transfer is enabled by setting the EPTDMAE bit in the DTC transfer setting re
1, zero-length data reception at endpoint 1 is ignored. When the DTC transfer is enabled,
RDFN bit for EP1 and PKTE bit for EP2 do not need to be set to 1 in TRG (note that the
must be set to 1 when the transfer data is less than the maximum number of bytes). When
data received at EP1 is read, the FIFO automatically enters the EMPTY state. When the n
number of bytes (64 bytes) are written to the EP2 FIFO, the FIFO automatically enters th
state, and the data in the FIFO can be transmitted (see figures 22.20 and 22.21). Since the
for a DTC transfer end interrupt is not automatically cleared, it should be cleared within t
interrupt processing.

then set the EP1DMAE bit in DMA to 1.

Figure 22.20 shows an example of receiving 150 bytes of data from the host. In this case processing which is the same as writing 1 to the RDFN bit in TRG is automatically perf three times. This internal processing is performed when the currently selected data FIFO empty. Accordingly, this processing is automatically performed both when 64-byte data and when data less than 64 bytes is sent.

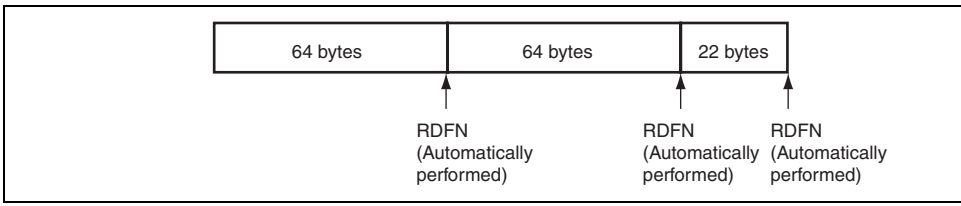


Figure 22.20 RDFN Bit Operation for EP1

omitted, the DTC transfer end interrupt is not cleared. To perform the DTC transfer again addition to the said procedure, set the number of transfers, set the DTCERF register, and the EP2DMAE bit in DMA to 1.

Figure 22.21 shows an example for transmitting 150 bytes of data to the host. In this case processing which is the same as writing 1 to the PKTE bit in TRG is automatically perfor twice. This internal processing is performed when the currently selected data FIFO becon Accordingly, this processing is automatically performed only when 64-byte data is sent.

When the last 22 bytes are sent, the internal processing for writing 1 to the PKTE bit is n performed, and the user must write 1 to the PKTE bit by software. In this case, the applic no more data to transfer but the USB function module continues to output DTC requests as long as the FIFO has an empty space. Therefore, when all data has been transferred, w the EP2DMAE bit in DMA to cancel DTC transfer requests, and then write 0 to the EP2E bit in IFR0 to clear the request for a DTC transfer end interrupt.

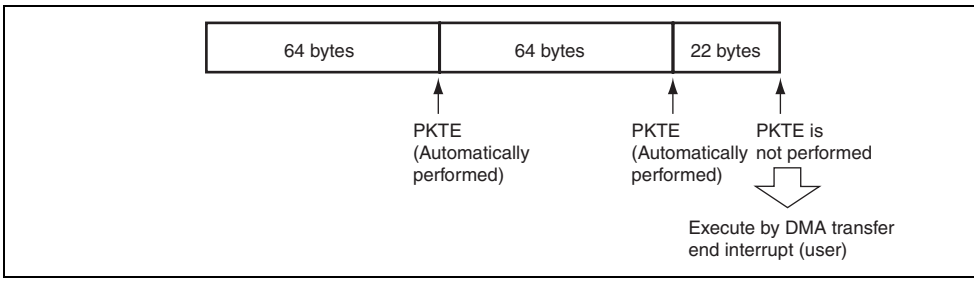


Figure 22.21 PKTE Bit Operation for EP2

When the DTC transfer is continuously performed:

- Set CRA and CRB of the DTC the number of transfers.
- Set the DTCERF register.
- Set 1 to the EP1DMAE bit in DMA.

(2) Endpoint 2

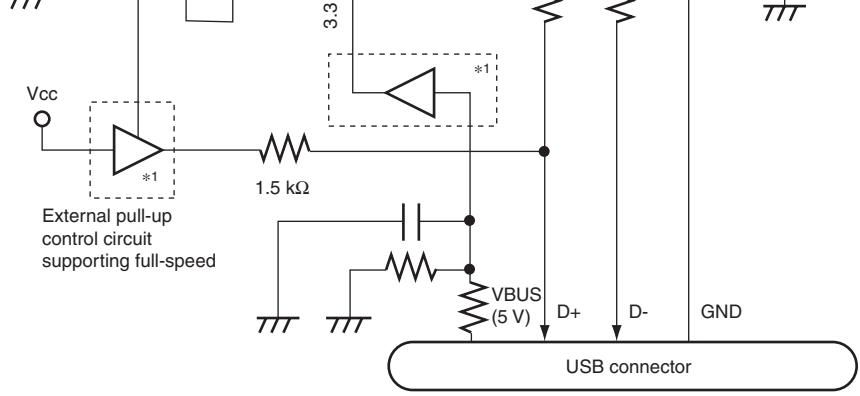
- Clear the EP2DMAE bit in DMA to 0.
- Write H'EF to the IFR0 register.

  Write 0 to the EP2EMPTY bit. The bit manipulation instruction should not be used f
  setting.

When the DTC transfer is continuously performed:

- Set CRA and CRB of the DTC the number of transfers.
- Set the DTCERF register.
- Set 1 to the EP2DMAE bit in DMA.

As USB states, etc., are managed by hardware in this module, a VBUS signal that rec
connection/disconnection is necessary. The power supply signal (VBUS) in the USB c
used for this purpose. However, if the cable is connected to the USB host/hub when th
function (system installing this LSI) power is off, a voltage (5 V) will be applied from
host/hub. Therefore, an IC (such as an HD74LV1G08A or 2G08A) that allows voltag
application when the system power is off should be connected externally.



Notes:
1. To protect this LSI from being damaged, use the IC (such as HD74LV-A Series) which ca be applied voltage even when the system power is turned off.
2. Prevent noise from the VBUS pin while the USB is performing communication.

Figure 22.22 Example of Circuitry in Self-Powered Mode

2. EPDR0s must always be read in 8-byte units. If the read is terminated at a midpoint, t received at the next setup cannot be read correctly.

22.10.2 Clearing the FIFO

If a USB cable is disconnected during data transfer, the data being received or transmitted remain in the FIFO. When disconnecting a USB cable, clear the FIFO.

While a FIFO is transferring data, it must not be cleared.

22.10.3 Overreading and Overwriting the Data Registers

Note the following when reading or writing to a data register of this module.

(1) Receive data registers

The receive data registers must not be read exceeding the valid amount of receive data, th number of bytes indicated by the receive data size register. Even for EPDR1 which has do FIFO buffers, the maximum data to be read at one time is 64 bytes. After the data is read current valid FIFO buffer, be sure to write 1 to EP1RDFN in TRG, which switches the va buffer, updates the receive data size to the new number of bytes, and enables the next dat received.

(2) Transmit data registers

The transmit data registers must not be written to exceeding the maximum packet size. Ev EPDR2 which has double FIFO buffers, write data within the maximum packet size at on After the data is written, write 1 to PKTE in TRG to switch the valid buffer and enable th data to be written. Data must not be continuously written to the two FIFO buffers.

22.10.6 Notes on TR Interrupt

Note the following when using the transfer request interrupt (TR interrupt) for IN transf
EP2, or EP3.

The TR interrupt flag is set if the FIFO for the target EP has no data when the IN token
from the USB host. However, at the timing shown in figure 22.24, multiple TR interrupt
successively. Take appropriate measures against malfunction in such a case.

Note: This module determines whether to return NAK if the FIFO of the target EP has
when receiving the IN token, but the TR interrupt flag is set after a NAK handsh
sent. If the next IN token is sent before PKTE of TRG is written to, the TR inter
set again.

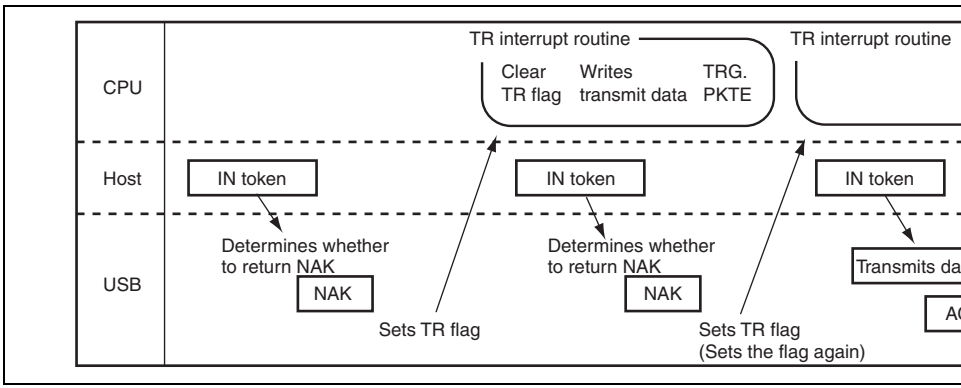


Figure 22.23 TR Interrupt Flag Set Timing

UXSEL	UEXTAL Input Frequency	EXTAL Input Frequency	USB Dedicated Clock (cku: 48 MHz)	ϕ
0	—	8.00 MHz	EXTAL × 6	EXTAL × 4 (
1	8.00 MHz	8.50 MHz	UEXTAL × 6	EXTAL × 4 (

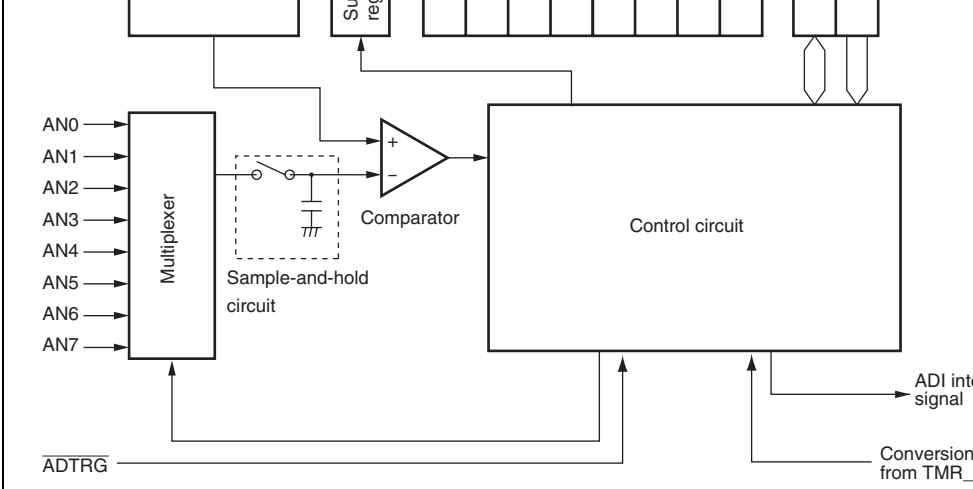
- Eight input channels
- Conversion time: 4.7 µs per channel (at 34-MHz operation)
- Two operating modes

  Single mode: Single-channel A/D conversion

  Scan mode: Continuous A/D conversion on 1 to 4 channels or continuous A/D conv
  1 to 8 channels
- Eight data registers

  Conversion results are held in a 16-bit data register for each channel
- Sample and hold function
- Three ways of starting A/D conversion
  - Software
  - Trigger from TMR_0
  - External trigger signal
- Interrupt request

  A/D conversion end interrupt (ADI) request can be generated
- Module stop mode can be set



[Legend]

ADCR: A/D control register	ADDRD: A/D data register D
ADCSR: A/D control/status register	ADDRE: A/D data register E
ADDRA: A/D data register A	ADDRF: A/D data register F
ADDRB: A/D data register B	ADDRG: A/D data register G
ADDRC: A/D data register C	ADDRH: A/D data register H

Note: * Supported only by the H8S/2472 Group and the H8S/2462 Group.

Figure 23.1 Block Diagram of the A/D Converter

Analog input pin 3	AN3	Input	
Analog input pin 4	AN4	Input	
Analog input pin 5	AN5	Input	
Analog input pin 6	AN6	Input	
Analog input pin 7	AN7	Input	
A/D external trigger input pin	$\overline{\text{ADTRG}}$	Input	External trigger input for starting A/D co
Analog power supply pin	AVcc	Input	Analog block power supply
Analog ground pin	AVss	Input	Analog block ground
Reference power supply pin	AVref	Input	Reference voltage for A/D converter Supported only by the H8S/2472 Group H8S/2462 Group.

- A/D data register G (ADDRG)
- A/D data register H (ADDRH)
- A/D control/status register (ADCSR)
- A/D control register (ADCR)

23.3.1 A/D Data Registers A to H (ADDRA to ADDRH)

The ADDR are eight 16-bit read-only registers, ADDRA to ADDRH, which store the res
A/D conversion. The ADDR registers, which store a conversion result for each channel, a
in table 23.2.

The converted 10-bit data is stored to bits 15 to 6. The lower 6-bit data is always read as

The data bus between the CPU and the A/D converter is 16-bit width and can be read dire
from the CPU. The ADDR must always be accessed in 16-bit unit. They cannot be access
bit unit.

The results of A/D conversion are stored in each registers, when the ADF flag is set to 1.

AN6	ADDRG
AN7	ADDRH

23.3.2 A/D Control/Status Register (ADCSR)

The ADCSR controls the operation of the A/D conversion.

Bit	Bit Name	Initial Value	R/W	Description
7	ADF	0	R/(W)*	A/D End Flag A status flag that indicates the end of A/D conve This flag indicates that the results of A/D conver stored in the A/D data registers. [Setting conditions] • When A/D conversion ends in single mode • When A/D conversion ends on all channels s in scan mode [Clearing conditions] • When 0 is written after reading ADF = 1 • When DTC starts by an ADI interrupt and AD read
6	ADIE	0	R/W	A/D Interrupt Enable Enables ADI interrupt by ADF when this bit is se

				This is a read-only bit and cannot be modified.
3	—	0	R/W	Reserved This bit is always read as 0. The initial value should changed.
2	CH2	All 0	R/W	Channel Select 2 to 0
1	CH1			Select analog input channels together with the SCA and the SCANS bit of ADCR.
0	CH0			When SCANE = 0, and SCANS = x: 000: AN0 001: AN1 010: AN2 011: AN3 100: AN4 101: AN5 110: AN6 111: AN7 When SCANE = 1 and SCANS = 0: 000: AN0 001: AN0 and AN1 010: AN0 to AN2 011: AN0 to AN3 100: AN4 101: AN4 and AN5 110: AN4 to AN6 111: AN4 to AN7 When SC and SCA: 000: AN0 001: AN1 010: AN0 011: AN0 100: AN0 101: AN0 110: AN0 111: AN0

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

[Legend] x: Don't care

				10: Enables starting by the ADTRG pin input. Other than above: Setting prohibited
5	SCANE	0	R/W	Scan Mode
4	SCANS	0	R/W	Select the operation mode of A/D conversion 0x: Single mode 10: Scan mode (consecutive A/D conversion of channels 1 t 11: Scan mode (consecutive A/D conversion of channels 1 t
3	CKS1	0	R/W	Clock Select 1 and 0
2	CKS0	0	R/W	Set the A/D conversion time. Setting should be while the conversion is stopped (ADST = 0). 00: Setting prohibited 01: Conversion time = 80 states (max) (20 MHz 10: Conversion time = 160 states (max) 11: Conversion time = 320 states (max)
1	ADSTCLR	0	R/W	A/D Start Clear Sets automatic clearing of the ADST bit in scan 0: Disables automatic clearing of ADST in scan 1: ADST is automatically cleared when A/D conv all the selected channels has been completed mode.

[Legend]

x: Don't care

In single mode, A/D conversion is performed only once on the specified single channel. Operations are as follows.

1. A/D conversion on the specified channel is started when the ADST bit in ADCSR is s by software or by the input of trigger signal.
2. When A/D conversion is completed, the result is transferred to the A/D data register corresponding to the channel.
3. On completion of A/D conversion, the ADF bit in ADCSR is set to 1. If the ADIE bit 1 at this time, an ADI interrupt request is generated.
4. The ADST bit remains set to 1 during A/D conversion. When conversion ends, the AI is automatically cleared to 0, and the A/D converter enters the idle state. If the ADST cleared during A/D conversion, the A/D converter stops conversion and enters the idl

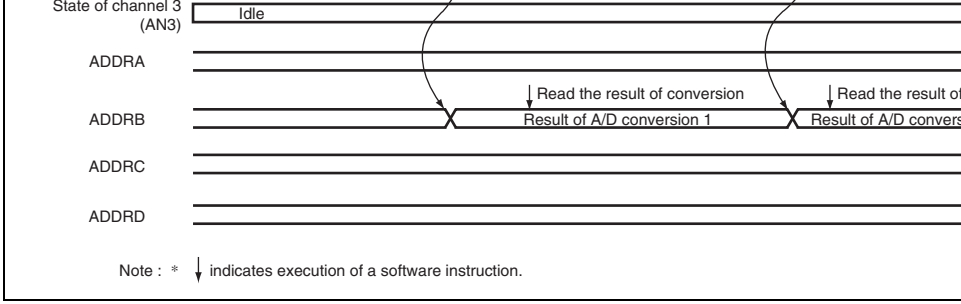
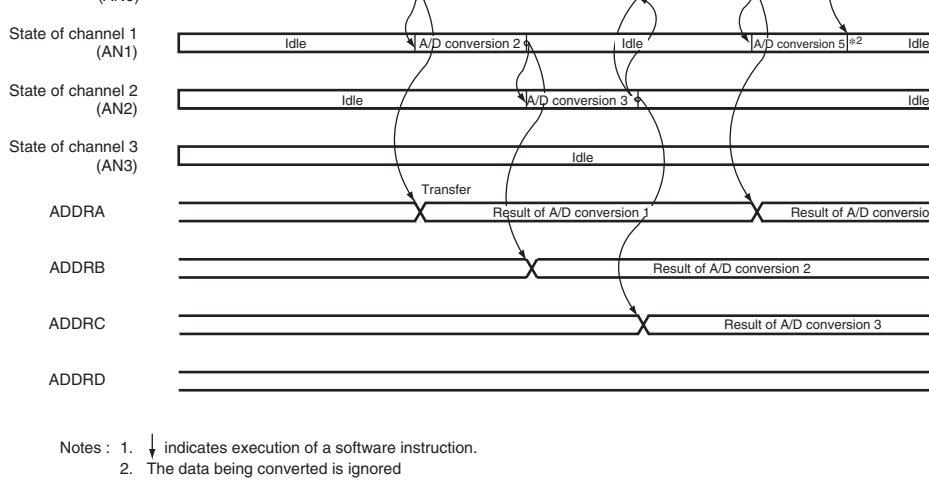


Figure 23.2 Example of A/D Converter Operation (When Channel 1 is Selected in Single Mode)

23.4.2 Scan Mode

In scan mode, A/D conversion is performed sequentially on the specified channels (four or eight channel maximum). Operations are as follows.

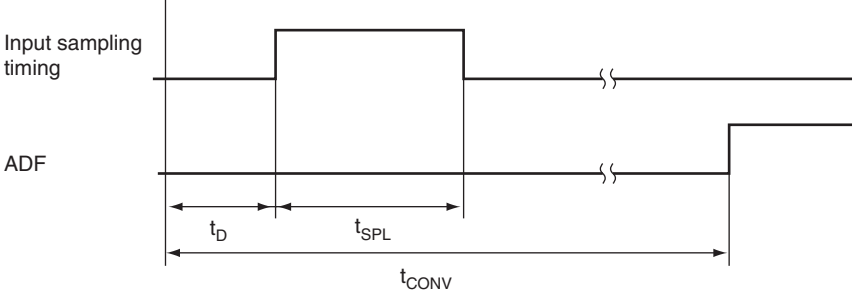
1. When the ADST bit in ADCSR is set to 1 by software or by the input of trigger signa conversion starts from the first channel of the selected channel. Consecutive A/D co either four channels maximum (SCANE and SCANS = B'10) or eight channels maxi (SCANE and SCANS = B'11) can be selected. In the case of consecutive A/D conve four channels, the operation starts from AN0 when CH2 = B'0, and starts from AN4 = B'1. In the case of consecutive A/D conversion on eight channels, the operation sta AN0.
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 bit in ADCSR i If the ADIE bit is set to 1 at this time, an ADI interrupt is requested after A/D conve Conversion of the first channel in the group starts again.



Notes : 1. ↓ indicates execution of a software instruction.
2. The data being converted is ignored

Figure 23.3 Example of A/D Converter Operation (When Channels AN0 to AN3 are Selected in Scan Mode)

In scan mode, the values given in table 23.3 apply to the first conversion time. The valu
table 23.4 apply to the second and subsequent conversions. In either case, bits CKS1 an
ADCR should be set so that the conversion time is within the ranges indicated by the A/
conversion characteristics.



[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 23.4 A/D Conversion Timing

Note: Values in the table are the number of states.

Table 23.4 A/D Conversion Characteristics (Scan Mode)

CKS1	CKS0	Conversion Time (Number of States)
0	0	Setting prohibited
	1	80 (Fixed)
1	0	160 (Fixed)
	1	320 (Fixed)

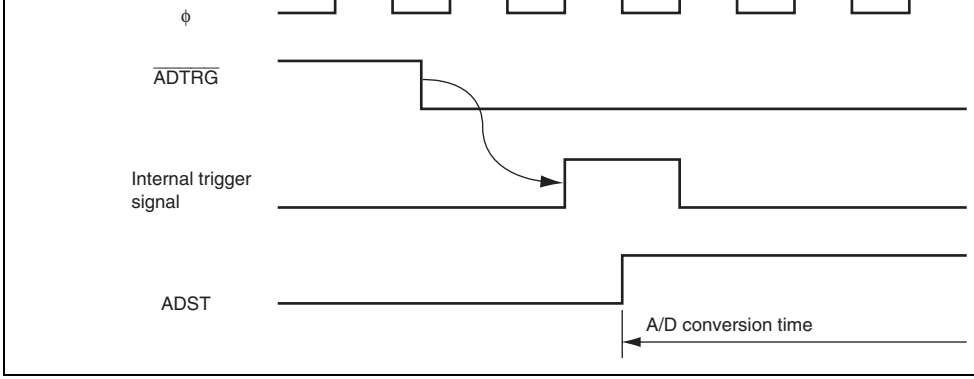


Figure 23.5 Timing of External Trigger Input

ADI	A/D conversion end	ADF	Possible

23.6 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 figure23.6).
- 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'00 0000 0000 (H
  B'00 0000 0001 (H'001) (see figure 23.7).
- 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'11 1111 1110 (H'3FE) to B'11 1111 1111 (H
  figure 23.7).
- Nonlinearity error

  The error with respect to the ideal A/D conversion characteristics between the zero v
  the full-scale voltage. Does not include the offset error, full-scale error, or quantizati
  (see figure 23.7).
- Absolute accuracy

  The deviation between the digital value and the analog input value. Includes the offs
  full-scale error, quantization error, and nonlinearity error.

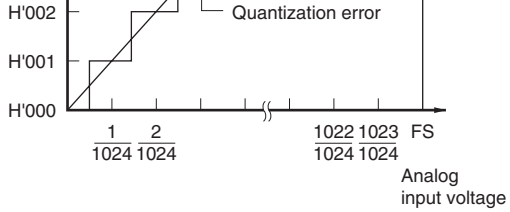


Figure 23.6 A/D Conversion Accuracy Definitions

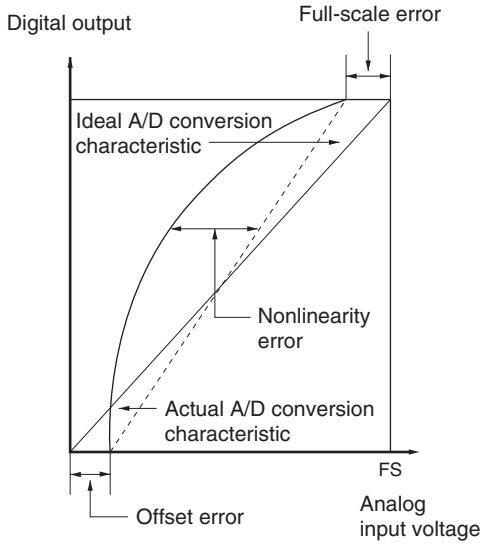


Figure 23.7 A/D Conversion Accuracy Definitions

This LSI's analog input is designed so that the conversion accuracy is guaranteed for an signal for which the signal source impedance is 5 kΩ or less. This specification is provid enable the A/D converter's sample-and-hold circuit input capacitance to be charged with sampling time; if the sensor output impedance exceeds 5 kΩ, charging may be insufficie may not be possible to guarantee the A/D conversion accuracy. However, if a large capa provided externally in single mode, the input load will essentially comprise only the inte resistance of 10 kΩ, and the signal source impedance is ignored. However, since a low-p effect is obtained in this case, it may not be possible to follow an analog signal with a la differential coefficient (voltage fluctuation ratio of 5 mV/μs or greater for example) (see 23.8). When converting a high-speed analog signal or converting in scan mode, a low-in buffer should be inserted.

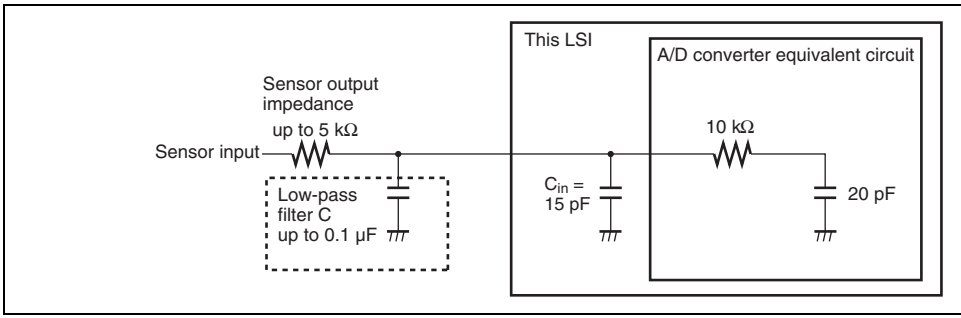


Figure 23.8 Example of Analog Input Circuit

If conditions shown below are not met, the reliability of this LSI 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 ≤ V_{AN} ≤ AVref.
- Relation between AVcc, AVss and Vcc, Vss

  The relationship between AVcc, AVss and Vcc, Vss should be Avcc = Vcc ± 0.3V an
  Vss. When the A/D converter is not used, set AVcc = Vcc and Avss = Vss.
- AVref pin reference voltage specification range

  The reference voltage of the AVref pin should be in the range AVref ≤ AVcc.

23.7.5 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. Also
circuitry must be isolated from the analog input signals (AN0 to AN7), the analog referen
voltage (AVref) and analog power supply (AVcc) by the analog ground (AVss). Also, the
ground (AVss) should be connected at one point to a stable digital ground (Vss) on the bo

the analog input pin voltage. Therefore, careful consideration is required upon deciding constants.

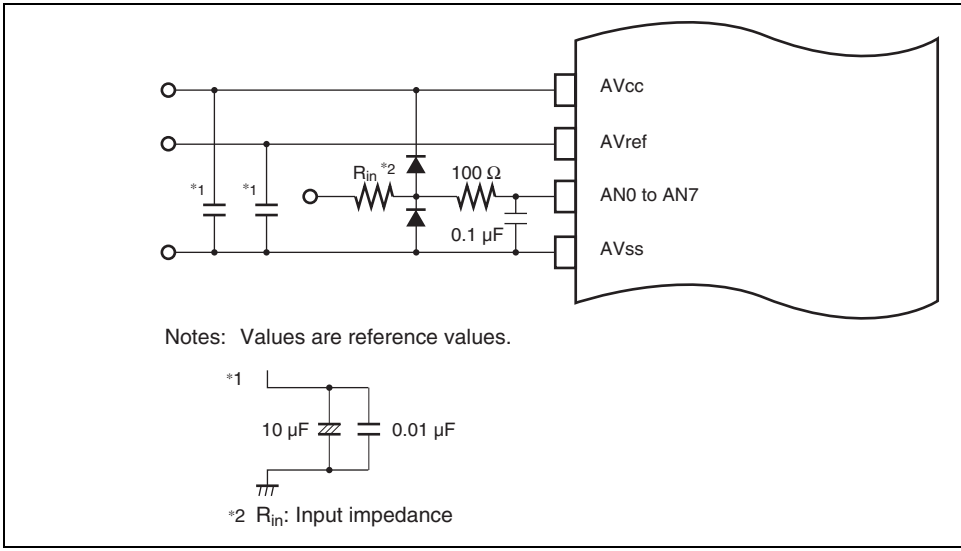


Figure 23.9 Example of Analog Input Protection Circuit

Table 23.6 Standard of Analog Pins

Item	Min.	Max.	Unit
Analog input capacitance	—	20	pF
Acceptable signal source impedance	—	5	kΩ

If this LSI enters software standby mode with the A/D conversion enabled, the content of converter is retained and about the same amount of analog supply current may flow as tha when A/D conversion in progress. If the analog supply current must be reduced in softwa standby mode, clear the ADST bit to disable the A/D conversion.

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

- Programming/erasing time

  The flash memory programming time is 1 ms (typ) in 128-byte simultaneous prograı
  approximately 7.8 µs per byte. The erasing time is 600 ms (typ) per 64-Kbyte block.

- Number of programming

  The number of flash memory programming can be up to 100 times at the minimum.
  ranged from 1 to 100 is guaranteed.)

- Four on-board programming modes

  — SCI boot mode

    This mode uses an on-chip SCI_1 interface to program and erase the user MAT.
    can automatically adjust the bit rate between the host and this LSI.

  — USB boot mode (only in the H8S/2472 Group)

    This mode uses an on-chip USB to program and erase the user MAT.

  — User program mode

    The user MAT can be programmed by using the optional interface.

  — User boot mode

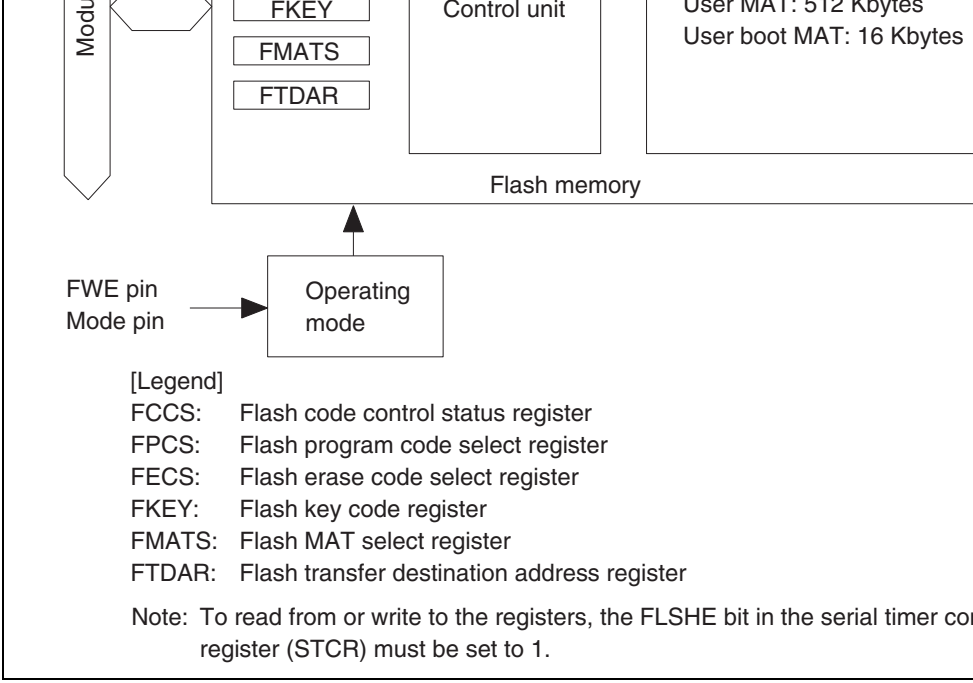
    The user boot program of the optional interface can be made and the user MAT c
    programmed.

- Programming/erasing protection

  Sets protection against flash memory programming/erasing via hardware, software, c
  protection.

- Programmer mode

  This mode uses the PROM programmer. The user MAT and user boot MAT can be
  programmed.



[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

Note: To read from or write to the registers, the FLSHE bit in the serial timer co register (STCR) must be set to 1.

Figure 25.1 Block Diagram of Flash Memory

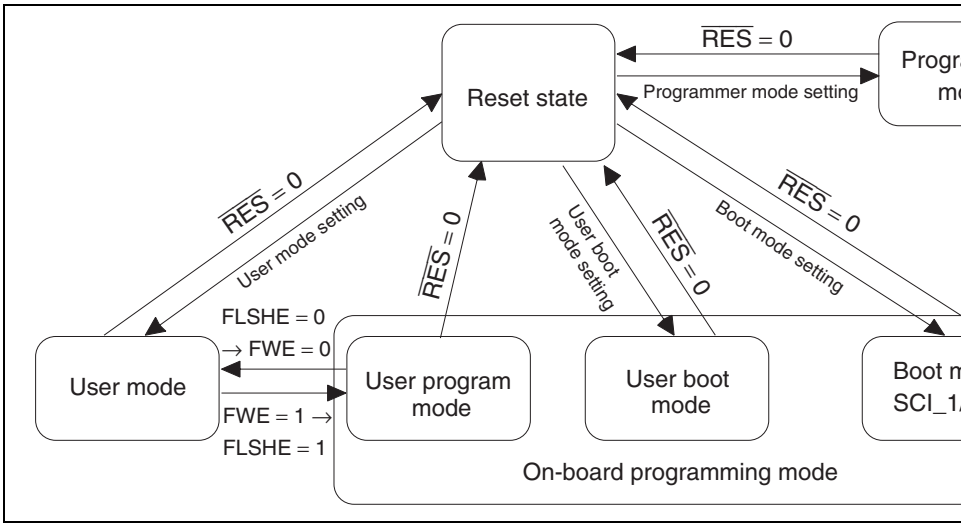


Figure 25.2 Mode Transition of Flash Memory

Programming/ erasing enable MAT	User MAT User boot MAT	User MAT	User MAT	User MAT User boot
All erasure	○ (Automatic)	○	○	○ (Autom
Block division erasure	○ *1	○	○	×
Program data transfer	From host via SCI or USB	Via optional device	Via optional device	Via progr
Reset initiation MAT	Embedded program storage MAT	User MAT	User boot MAT*2	—
Transition to user mode	Changing mode setting and reset	Changing FLSHE bit and FWE pin	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.

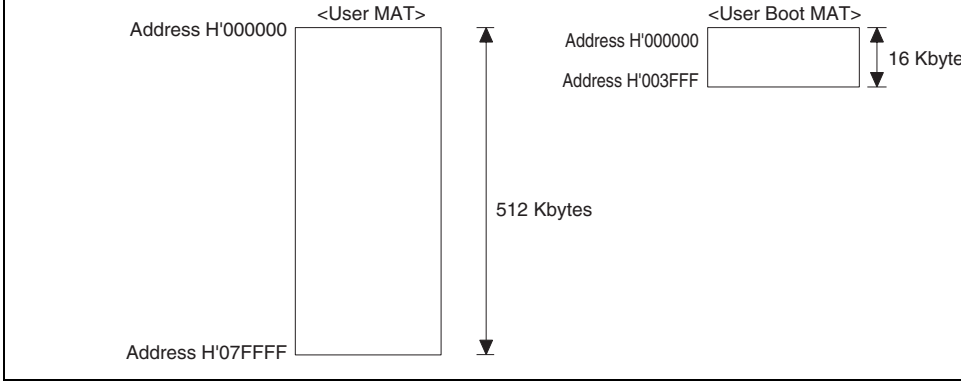


Figure 25.3 Flash Memory Configuration

The size of the user MAT is different from that of the user boot MAT. An address which the size of the 16-Kbyte user boot MAT should not be accessed. If the attempt is made, read as undefined value.

25.1.4 Block Division

The user MAT is divided into seven 64-Kbyte blocks, one 32-Kbyte block, and eight 4-K blocks as shown in figure 25.4. The user MAT can be erased in this divided-block units, erase-block number of EB0 to EB15 is specified when erasing. Programming is perform byte units starting at the addresses whose lowest-order byte is H'00 or H'80.

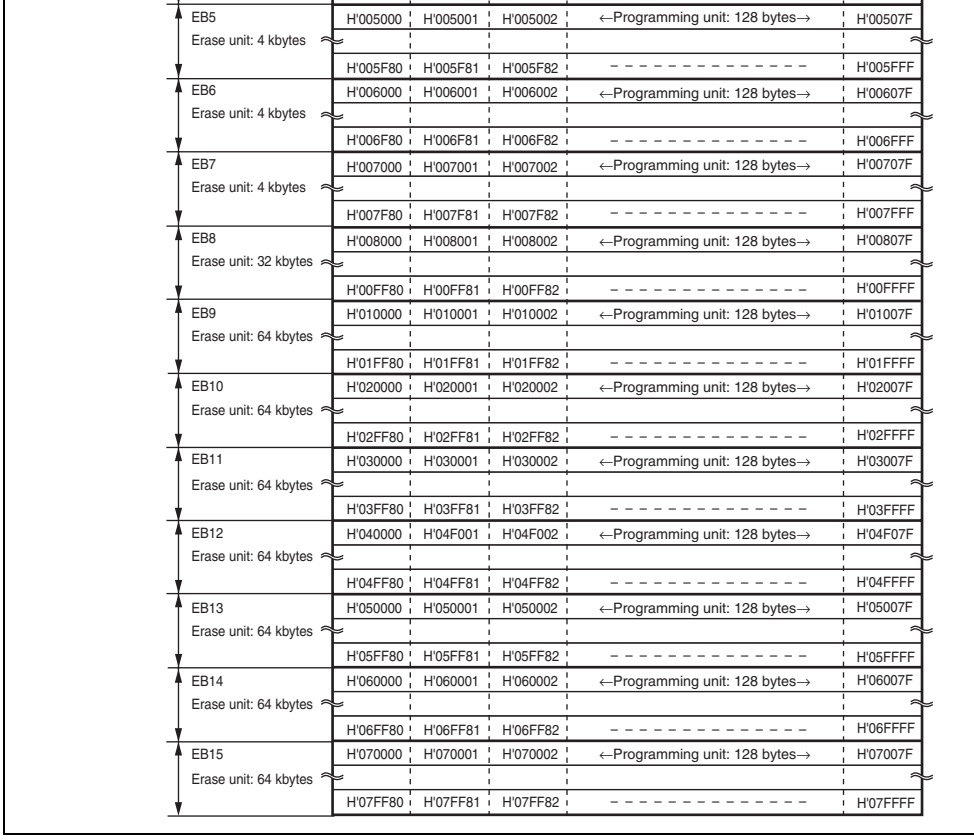


Figure 25.4 Block Division of User MAT

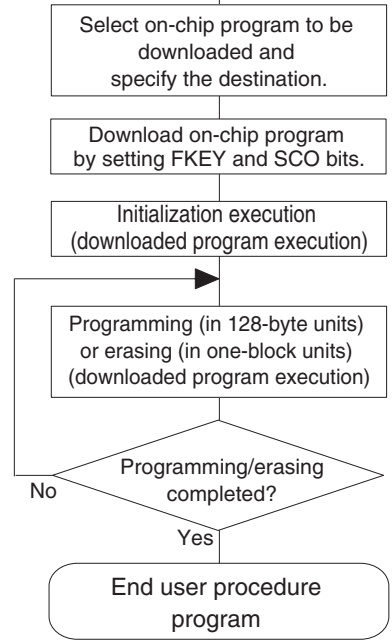


Figure 25.5 Overview of User Procedure Program

1. Selection of on-chip program to be downloaded

   For programming/erasing execution, the FLSHE bit in STCR must be set to 1 to tran user program mode.

   This LSI has programming/erasing programs which can be downloaded to the on-chi 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 flash transfer destination address register (FTDAR).

3. Initialization of programming/erasing

   The operating frequency is set before execution of programming/erasing. This setting
   performed by using the programming/erasing interface parameter.

4. Programming/erasing execution

   For programming/erasing execution, the FLSHE bit in STCR and the FWE pin must b
   to transition to user program mode.

   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

   All interrupts are prohibited during programming and erasing. Interrupts must be mas
   within the user system.

5. When programming/erasing is executed consecutively

   When the processing is not ended by the 128-byte programming or one-block erasure,
   program address/data and erase-block number must be updated and consecutive
   programming/erasing is required.

   Since the downloaded on-chip program is left in the on-chip RAM after the processin
   download and initialization are not required when the same processing is executed
   consecutively.

TxD1	Output	Serial transmit data output (used in SCI boot mod
RxD1	Input	Serial receive data input (used in SCI boot mode)
USD+, USD−	Input/output	USB data input/output (used in USB boot mode)
VBUS	Input	Detects USB cable connection/disconnection (use boot mode)
PUPDPLS	Input	USD+ pull-up control
PF5	Input	Sets SCI boot mode or USB boot mode (used in b setting)

- Flash MAT select register (FMATS)
- Flash transfer destination address register (FTDAR)
- Download pass/fail result (DPFR)
- Flash pass/fail result (FPFR)
- Flash multipurpose address area (FMPAR)
- Flash multipurpose data destination area (FMPDR)
- Flash erase Block select (FEBS)
- Flash programming/erasing frequency control (FPEFEQ)

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 25.3.

Programming/ Erasing Interface Parameter	DPFR	○	—	—	—
	FPFR	—	○	○	○
	FPEFEQ	—	○	—	—
	FMPAR	—	—	○	—
	FMPDR	—	—	○	—
	FEBS	—	—	—	○

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 a target MAT.

7	FWE	1/0	R	Flash Program Enable Monitors the signal level input to the FWE pin a enables or disables programming/erasing flash 0: Programming/erasing disabled 1: Programming/erasing enabled
6, 5	—	All 0	R/W	Reserved The initial value should not be changed.

Programming/erasing protection for flash me
(error protection) is invalid.

[Clearing condition]

- At a reset or in hardware standby mode

1: An error occurs during programming/erasing
memory.
Programming/erasing protection for flash me
(error protection) is valid.

[Setting conditions]

- When an interrupt, such as NMI, occurs du
programming/erasing flash memory.
- When the flash memory is read during
programming/erasing flash memory (includ
vector read or an instruction fetch).
- When the SLEEP instruction is executed d
programming/erasing flash memory (includ
software-standby mode)
- When a bus master other than the CPU, su
DTC, gets bus mastership during
programming/erasing flash memory.

				The interrupt exception handling on and after ve number 32 should not be used because the cor vector is not read, resulting in the CPU runaway 0: The space for the interrupt vector table is not When interrupt vector data is not read succes the operation for the interrupt exception hand cannot be guaranteed. An occurrence of any should be masked. 1: The space for the interrupt vector table is mo Even when interrupt vector data is not read successfully, the interrupt exception handling vector number 31 is enabled.
2, 1	—	All 0	R/W	Reserved The initial value should not be changed.

		after setting this bit to 1. Since this bit is cleared to 0 when download is completed, this bit cannot be read as 1. All interrupts must be disabled. This should be the user system. 0: Download of the on-chip programming/eras program to the on-chip RAM is not executed. [Clearing condition] When download is completed 1: Request that the on-chip programming/eras program is downloaded to the on-chip RAM occurred. [Setting conditions] When all of the following conditions are satisfi set to this bit • H'A5 is written to FKEY • During execution in the on-chip RAM

Note: * This bit is a write only bit. This bit is always read as 0.

				[Clearing 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 erasing program.

Bit	Bit Name	Initial Value	R/W	Description
7 to 1	—	All 0	R/W	Reserved
				The initial value should not be changed.
0	EPVB	0	R/W	Erase Pulse Verify Block
				Selects the erasing program.
				0: On-chip erasing program is not selected.
				[Clearing condition] When transfer is completed
				1: On-chip erasing program is selected.

4	K4	0	R/W	
3	K3	0	R/W	
2	K2	0	R/W	
1	K1	0	R/W	
0	K0	0	R/W	

cannot be set to the SCO bit. Therefore downl
the on-chip RAM cannot be executed.

Only when H'5A is written, programming/erasi
executed. Even if the on-chip programming/er
program is executed, the flash memory cannot
programmed or erased when the value other t
is written to FKEY.

H'A5: Writing to the SCO bit is enabled. (The S
cannot be set by the value other than H'

H'5A: Programming/erasing is enabled. (The v
than H'A5 is in software protection state

H'00: Initial value

2	MS2	0	R/W	Switching between User MAT and User Boot MA user boot MAT cannot be programmed in user p mode if user boot MAT is selected by FMATS. T boot MAT must be programmed in boot mode o programmer mode.) H'AA: The user boot MAT is selected (in user-M selection state when the value of these bi other than H'AA) Initial value when these bits are initiated i boot mode. H'00: Initial value when these bits are initiated i except for user boot mode (in user-MAT s state) [Programmable condition] These bits are in the execution state in the on-c
1	MS1	0/1*	R/W	
0	MS0	0	R/W	

Note: * Set to 1 when in user boot mode, otherwise set to 0.

				of H'00 to H'03 is determined when an on-chip downloaded by setting the SCO bit in FCCS to sure that this bit is cleared to 0 before setting t to 1 and the value specified by TDA6 to TDA0 the range of H'00 to H'03. 0: The value specified by bits TDA6 to TDA0 is range. 1: The value specified by is TDA6 to TDA0 is o range (H'04 to H'FF) and the download is st
6	TDA6	0	R/W	Transfer Destination Address
5	TDA5	0	R/W	Specifies the start address to download an on- program. H'00 to H'03 can be specified as the address in the on-chip RAM space. H'00: H'FFE080 is specified as a start address download an on-chip program. H'01: H'FF0800 is specified as a start address download an on-chip program. H'02: H'FF1800 is specified as a start address download an on-chip program. H'03: H'FF8800 is specified as a start address download an on-chip program. H'04 to H'FF: Setting prohibited. Specifying th sets the TDER bit to 1 and stop download.
4	TDA4	0	R/W	
3	TDA3	0	R/W	
2	TDA2	0	R/W	
1	TDA1	0	R/W	
0	TDA0	0	R/W	

(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
3. Programming
4. Erasing

These items use different parameters. The correspondence table is shown in table 25.4. Th
meaning of the bits in FPFR varies in each processing program: initialization, programmi
erasure. For details, see descriptions of FPFR for each process.

Flash multipurpose address area	FMPAR	—	—	○	—	R/W	Undefined
Flash multipurpose 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 s
FTDAR

be used to determine if downloading is executed or not. Since the confirmation whether t
bit is set to 1 is difficult, the certain determination must be performed by writing the sing
the start address specified by FTDAR to the value other than the return value of downloa
example, H'FF) before the download start (before setting the SCO bit to 1).

Bit	Bit Name	Initial Value	R/W	Description
7 to 3	—	—	—	Unused Return 0
2	SS	—	R/W	Source Select Error Detect Only one type for the on-chip program which ca downloaded can be specified. When more than types of the program are selected, the program selected, or the program is selected without ma error is occurred. 0: Download program can be selected normally 1: Download error is occurred (multi-selection o program which is not mapped is selected)
1	FK	—	R/W	Flash Key Register Error Detect Returns the check result whether the value of F set to H'A5. 0: KEY setting is normal (FKEY = H'A5) 1: Setting value of FKEY becomes error (FKEY other than H'A5)

CPU)

This parameter sets the operating frequency of the CPU. The settable range of the operati
frequency in this LSI is 20 to 34 MHz.

Bit	Bit Name	Initial Value	R/W	Description
31 to 16	—	—	—	Unused This bit should be cleared to 0.
15 to 0	F15 to F0	—	R/W	Frequency Set Set the operating frequency of the CPU. With th multiplication function, set the frequency multipl setting value must be calculated as the followin methods. 1. The operating frequency which is shown in M must be rounded in a number to three decim and be shown in a number of two decimal pl 2. The value multiplied by 100 is converted to t digit and is written to the FPEFEQ paramete (general register ER0). For example, when the operating frequency of t is 33.000 MHz, the value is as follows. 1. The number to three decimal places of 34.0 rounded and the value is thus 34.00. 2. The formula that 34.00 × 100 = 3400 is conv the binary digit and B'0000,1101,0100,1000 is set to ER0.

				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 no 0: Initialization is ended normally (no error) 1: Initialization is ended abnormally (error occu

(3) Programming Execution

When flash memory is programmed, the programming destination address on the user M be passed to the programming program in which the program data is downloaded.

1. The start address of the programming destination on the user MAT must be stored in register ER1. This parameter is called as flash multipurpose address area parameter (
   Since the program data is always in units of 128 bytes, the lower eight bits (A7 to A0 H'00 or H'80 as the boundary of the programming start address on the user MAT.
2. The program data for the user MAT must be prepared in the consecutive area. The p data must be in the consecutive space which can be accessed by using the MOV.B in 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 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 st general register ER0. This parameter is called as flash multipurpose data destination parameter (FMPDR).

For details on the program processing procedure, see section 25.4.3, User Program Mod

31 to 0	MOA31 to MOA0	—	R/W	Store the start address of the programming des on the user MAT. The consecutive 128-byte programming is executed starting from the spec address of the user MAT. Therefore, the specifi programming start address becomes a 128-byte boundary and MOA6 to MOA0 are always 0.

(b) Flash multipurpose data destination parameter (FMPDR: general register ER0 CPU):

This parameter stores the start address in the area which stores the data to be programmed user MAT. When the storage destination of the program data is in flash memory, an error The error occurrence is indicated by the WD bit in FPFR.

Bit	Bit Name	Initial Value	R/W	Description
31 to 0	MOD31 to MOD0	—	R/W	Store the start address of the area which stores program data for the user MAT. The consecutiv byte data is programmed to the user MAT starti the specified start address.

(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	—	—	—	Unused Return 0.

				1: Programming cannot be performed (FWE FLER = 1)
5	EE	—	R/W	Programming Execution Error Detect 1 is returned to this bit when the specified data be written because the user MAT was not eras bit is set to 1, there is a high possibility that the is partially rewritten. In this case, after removin factor, erase the user MAT. If FMATS is set to H'AA and the user boot MA selected, an error occurs when programming is performed. In this case, both the user MAT an MAT are not rewritten. Programming of the us MAT should be performed in boot mode or pro mode. 0: Programming has ended normally 1: Programming has ended abnormally (prog result is not guaranteed)
4	FK	—	R/W	Flash Key Register Error Detect Returns the check result of the value of FKEY start of the programming processing. 0: FKEY setting is normal (FKEY = H'5A) 1: FKEY setting is error (FKEY = value other th
3	—	—	—	Unused Returns 0.

				• When the programming destination address area other than flash memory is specified • When the specified address is not in a 128-b boundary. (The lower eight bits of the addre other than H'00 and H'80.) 0: Setting of programming destination address i 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

Bit	Bit Name	Initial Value	R/W	Description
31 to 16	—	—	—	Unused These bits should be cleared to H'0.
15	EB15	—	R/W	Erase Block Set the erase-block number in the range from corresponds to the EB0 block, and 15 corresp the EB15 block.
14	EB14	—	R/W	
13	EB13	—	R/W	
12	EB12	—	R/W	
11	EB11	—	R/W	
10	EB10	—	R/W	
9	EB9	—	R/W	
8	EB8	—	R/W	
7	EB7	—	R/W	
6	EB6	—	R/W	
5	EB5	—	R/W	
4	EB4	—	R/W	
3	EB3	—	R/W	
2	EB2	—	R/W	
1	EB1	—	R/W	
0	EB0	—	R/W	

				entered. When the low-level signal is input to th pin or the error protection state is entered, 1 is this bit. The state can be confirmed with the FW FLER bits in FCCS. For conditions to enter the protection state, see section 25.5.3, Error Prote 0: FWE and FLER settings are normal (FWE = FLER = 0) 1: Programming cannot be performed (FWE = FLER = 1)
5	EE	—	R/W	Erasure Execution Error Detect 1 is returned to this bit when the user MAT coul erased or when flash-memory related register s are partially changed. If this bit is set to 1, there possibility that the user MAT is partially erased. case, after removing the error factor, erase the MAT. If FMATS is set to H'AA and the user boo selected, an error occurs when erasure is perfo this case, both the user MAT and user boot MA erased. Erasing of the user boot MAT should be performed in boot mode or programmer mode. 0: Erasure has ended normally 1: Erasure has ended abnormally (erasure res 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 tha

	Indicates whether the erasing processing is en normally or not. 0: Erasure is ended normally (no error) 1: Erasure is ended abnormally (error occurs)

25.4 On-Board Programming Mode

When the pin is set in on-board programming mode and the reset start is executed, the o programming state that can program/erase the on-chip flash memory is entered. On-boar programming mode has three operating modes: boot mode, user program mode, and use mode.

For details of the pin setting for entering each mode, see table 25.5. For details of the sta transition of each mode for flash memory, see figure 25.2.

Table 25.5 Setting On-Board Programming Mode

Mode Setting		FWE	$\overline{\text{MD2}}$	MD1	NMI	P
Boot mode	SCI_1	1	0	0	1	0
	USB	1	0	0	1	1
User program mode		1*	1	1	0/1	–
User boot mode		1	0	0	0	–

Note: * Before downloading the programming/erasing programs, the FLSHE bit must 1 to transition to user program mode.

setting in boot mode, see table 25.5. The NMI and other interrupts are ignored in boot mo
However, the NMI and other interrupts should be disabled in the user system.

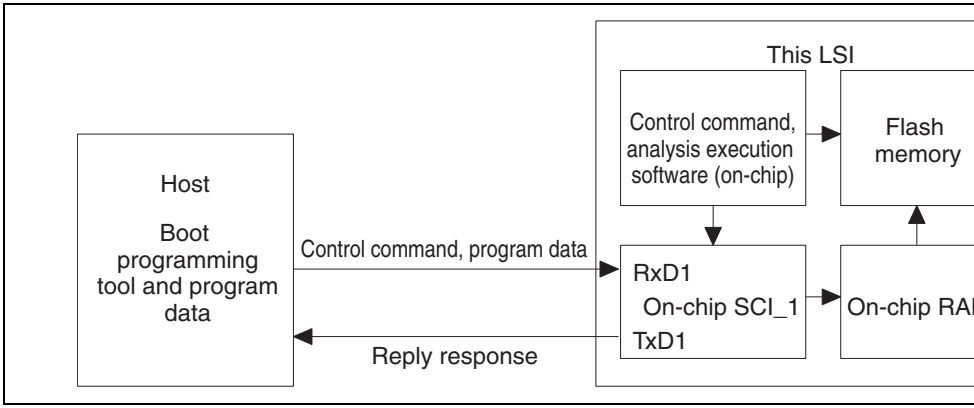


Figure 25.6 System Configuration in Boot Mode

set to 9,600 bps or 19,200 bps.

The system clock frequency, which can automatically adjust the transfer bit rate of the h
the bit rate of this LSI, is shown in table 25.6. Boot mode must be initiated in the range
system clock.

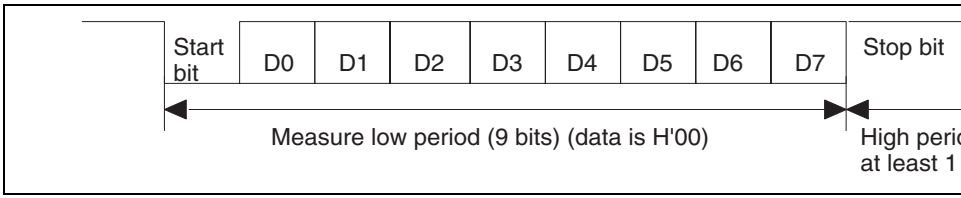


Figure 25.7 Automatic-Bit-Rate Adjustment Operation of SCI

Table 25.6 System Clock Frequency for Automatic-Bit-Rate Adjustment by This

Bit Rate of Host	System Clock Frequency
9,600 bps	20 to 34 MHz
19,200 bps	

After inquiries have finished, all user MAT and user boot MAT are automatically era

4. Waiting for programming/erasing command
   - When the program preparation notice is received, the state for waiting program da entered. The programming start address and program data must be transmitted foll the programming command. When programming is finished, the programming sta must be set to H'FFFFFFFF and transmitted. Then the state for waiting program d returned to the state of programming/erasing command wait.
   - When the erasure preparation notice is received, the state for waiting erase-block d entered. The erase-block number must be transmitted following the erasing comm When the erasure is finished, the erase-block number must be set to H'FF and tran 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 blo programmed without a reset start after programming is executed in boot mode. Wh programming can be executed by only one operation, all blocks are erased before for waiting programming/erasing/other command is entered. The erasing operation required.
   - There are many commands other than programming/erasing. Examples are sum ch blank check (erasure check), and memory read of the user MAT/user boot MAT a acquisition of current status information.

Note that memory read of the user MAT/user boot MAT can only read the programmed d all user MAT/user boot MAT has automatically been erased.

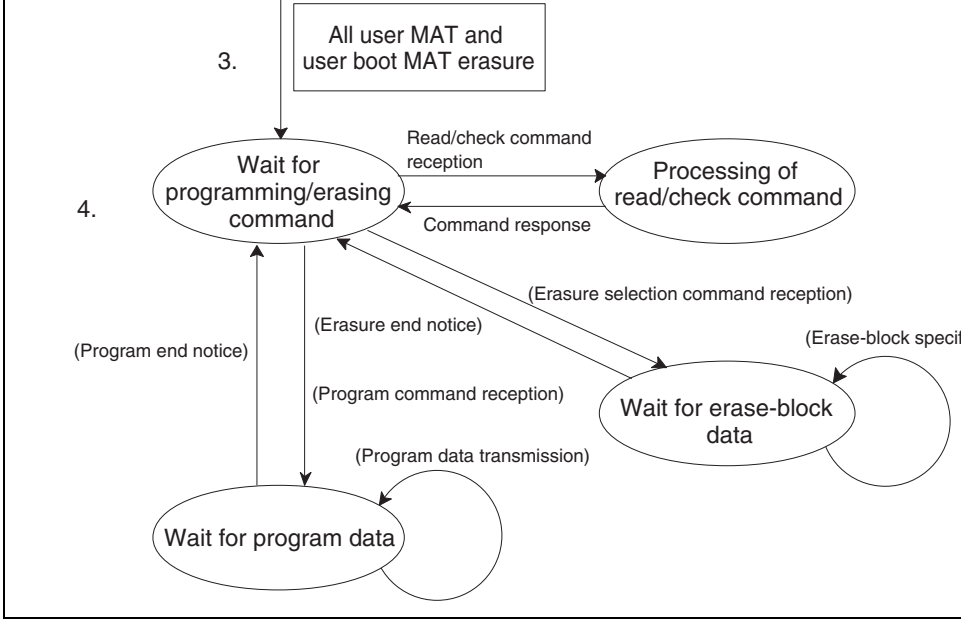


Figure 25.8 Overview of Boot Mode State Transition Diagram

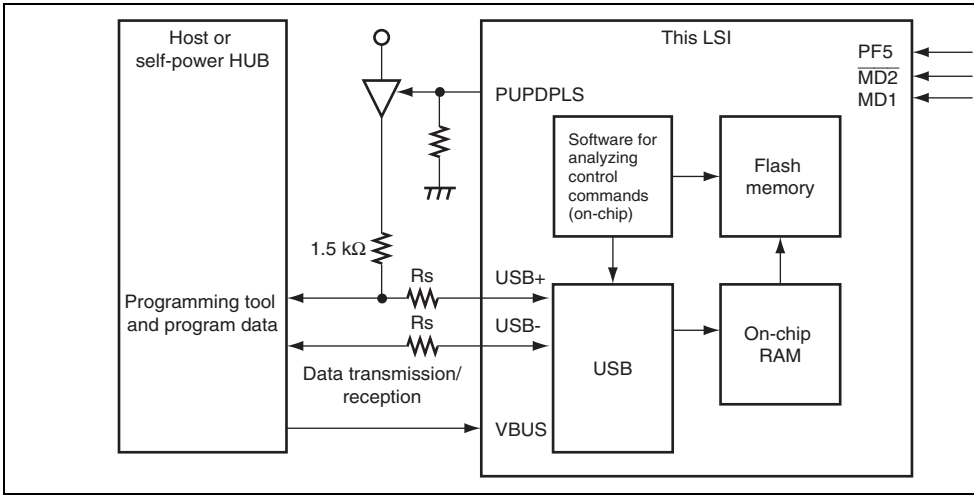
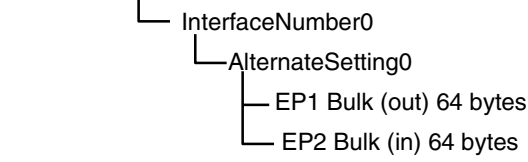


Figure 25.9 System Configuration in USB Boot Mode


InterfaceNumber0
AlternateSetting0
EP1 Bulk (out) 64 bytes
EP2 Bulk (in) 64 bytes

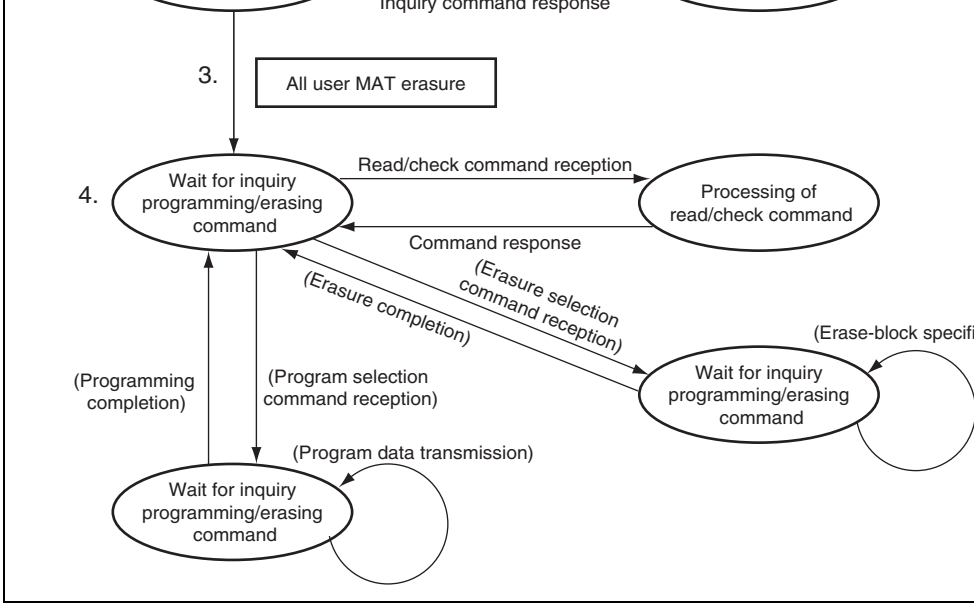


Figure 25.10 USB Boot Mode State Transition Diagram

1. After a transition to the USB boot mode is made, the boot program embedded in this l initialized. This LSI performs enumeration to the host after the USB boot program is initialized.
2. Inquiry information about the size, configuration, start address, and support status of t MAT is transmitted to the host.
3. After inquiries have finished, all user MAT are automatically erased.

- If the bus powered HUB is disconnected during the flash memory programming and permanent damage to the LSI may result.

period of 100 µs which is longer than normal.

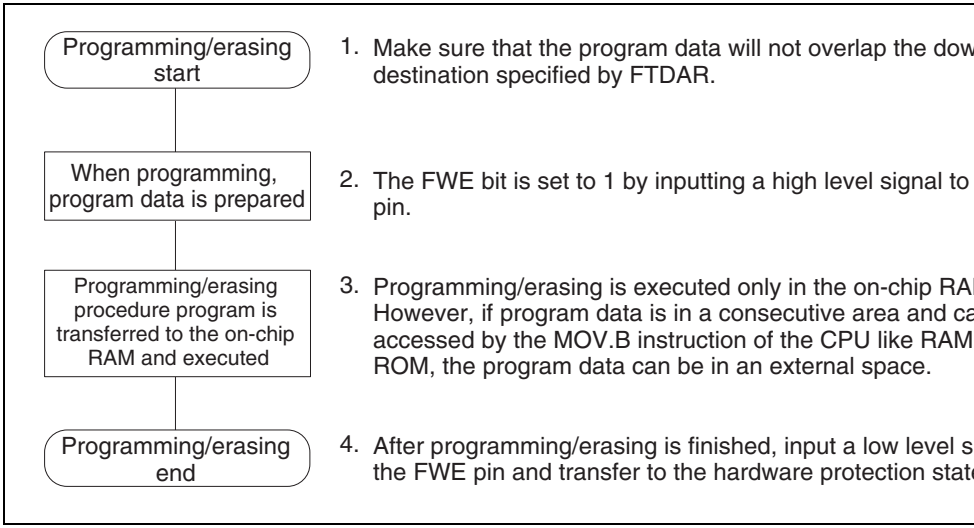


Figure 25.11 Programming/Erasing Overview Flow

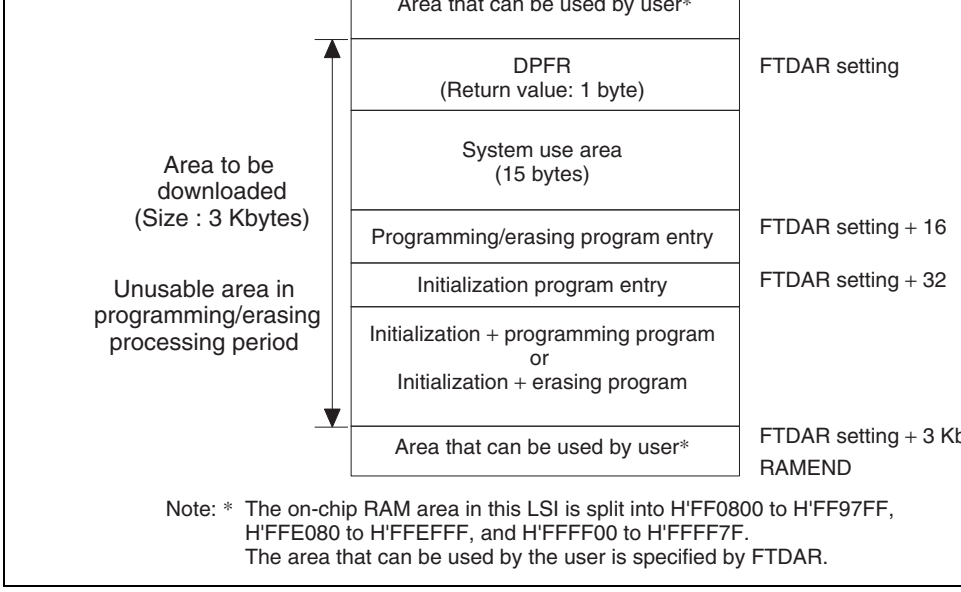


Figure 25.12 RAM Map When Programming/Erasing is Executed

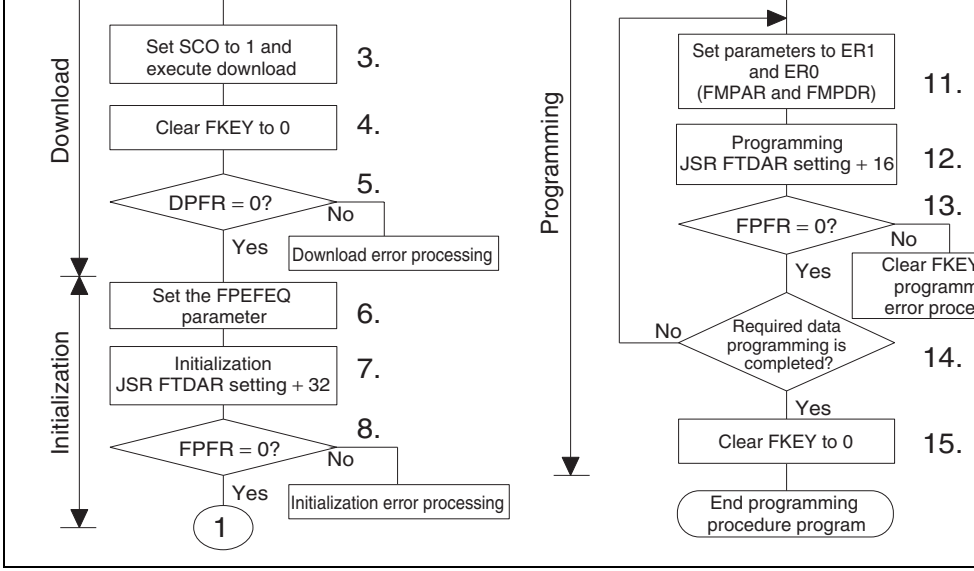


Figure 25.13 Programming Procedure

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 25.4.5, Procedure Program and Storable Ar Programming Data.

The following description assumes the area to be programmed on the user MAT is erased program data is prepared in the consecutive area. When erasing is not executed, erasing is 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.

2. Program H'A5 in FKEY

   If H'A5 is not written to FKEY for protection, 1 cannot be set to the SCO bit for dow
   request.

3. 1 is set to the SCO bit of FCCS and then download is executed.

   To set 1 to the SCO bit, the following conditions must be satisfied.

   — 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 SCO bit i
   to the user procedure program, the SCO is cleared to 0. Therefore, the SCO bit canno
   confirmed to be 1 in the user procedure program.

   The download result can be confirmed only by the return value of DPFR. Before the
   is set to 1, incorrect determination must be prevented by setting the one byte of the s
   address (to be used as DPFR) specified by FTDAR to a value other than the return v
   H'FF).

   When download is executed, particular interrupt processing, which is accompanied b
   switch as described below, is performed as an internal microcomputer processing. Fo
   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 ar
   the transfer processing to the on-chip RAM specified by FTDAR is executed.

   — The SCO bit in FCCS is cleared to 0.

   — The return value is set to the DPFR parameter.

— Since a stack area of 128 bytes at the maximum is used, the area must be allocated setting the SCO bit to 1.

— If a flash memory access by the DTC signal is requested during downloading, the cannot be guaranteed. Therefore, an access request by the DTC signal must not be generated.

4. FKEY is cleared to H'00 for protection.

5. 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 downloac destination specified by FTDAR). If the value is H'00, download has been perforn 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 c confirm the setting of the TDER bit (bit 7) in FTDAR.

   — If the value of the DPFR parameter is different from before downloading, check th (bit 2) and the FK bit (bit 1) in the DPFR parameter to ensure that the download p selection and FKEY setting were normal, respectively.

6. The operating frequency is set in the FPEFEQ parameter for initialization.

   — The current frequency of the CPU clock is set to the FPEFEQ parameter value (ge register ER0).

   The settable range of the FPEFEQ parameter is 20 to 34 MHz. When the frequenc out of this range, an error is returned to the FPFR parameter of the initialization pr and initialization is not performed. For details on the frequency setting, see the de in 25.3.2 (2) (a), Flash programming/erasing frequency parameter (FPEFEQ: gene register ER0 of CPU).

— R0L is a return value of the FPFR parameter.
— Since the stack area is used in the initialization program, 128-byte stack area at th
maximum 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 d

8. The return value in the initialization program, FPFR (general register R0L) is determ

9. 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 bus mastership to other than the CPU, such as to the DTC,
prohibited.

   To disable interrupts, bit 7 (I) in the condition code register (CCR) of the CPU shoul
B'1 in interrupt control mode 0 or bits 7 and 6 (I and UI) should be set to B'11 in inte
control mode 1. Interrupts other than NMI are held and not executed.

   The NMI interrupts must be masked within the user system.

   The interrupts that are held must be executed after all program processing.

   When the bus mastership is moved to other than the CPU, such as to the DTC, the en
protection state is entered. Therefore, taking bus mastership by the DTC is prohibited

10. FKEY must be set to H'5A and the user MAT must be prepared for programming.

— Example of the FMPDR setting

When the storage destination of the program data is flash memory, even if the pro
execution routine is executed, programming is not executed and an error is returne
FPFR parameter. In this case, the program data must be transferred to the on-chip
and then programming must be executed.

12. Programming

There is an entry point of the programming program in the area from the start address
by FTDAR + 16 bytes of the on-chip RAM. The subroutine is called and programmin
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.

— Since the stack area is used in the programming program, a stack area of 128 byte
maximum must be allocated in RAM.

13. The return value in the programming program, FPFR (general register R0L) is determ

14. Determine whether programming of the necessary data has finished.
If more than 128 bytes of data are to be programmed, specify FMPAR and FMPDR in
byte units, and repeat steps 12 to 14. Increment the programming destination address
bytes and update the programming data pointer correctly. If an address which has alre
programmed is written to again, not only will a programming error occur, but also fla
memory will be damaged.

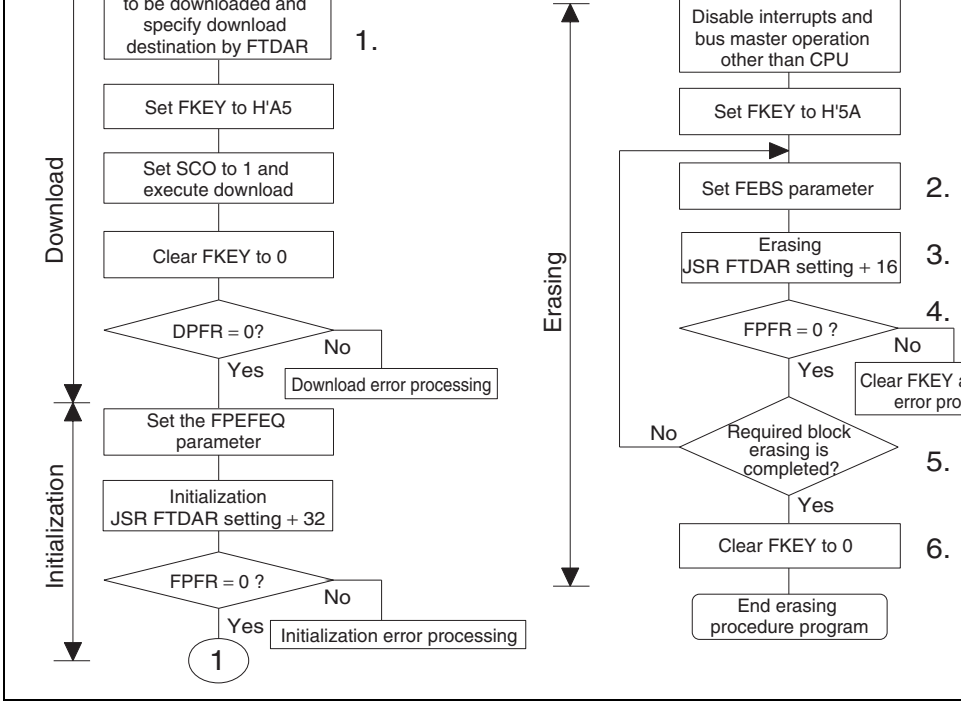


Figure 25.14 Erasing Procedure

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 25.4.5, Procedure Program and Storable A
Programming Data.

parameter.

Specify the start address of a download destination by FTDAR.

The procedures to be carried out after setting FKEY, e.g. download and initialization, same as those in the programming procedure. For details, refer to Programming Proce User Program Mode in section 25.4.3 (2), Programming Procedure in User Program M

The procedures after setting parameters for erasing programs are as follows:

2. 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.

3. 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 128 bytes at the maximum must be allocated in RAM.

4. The return value in the erasing program, FPFR (general register R0L) is determined.

Figure 25.15 shows a repeating procedure of erasing and programming.

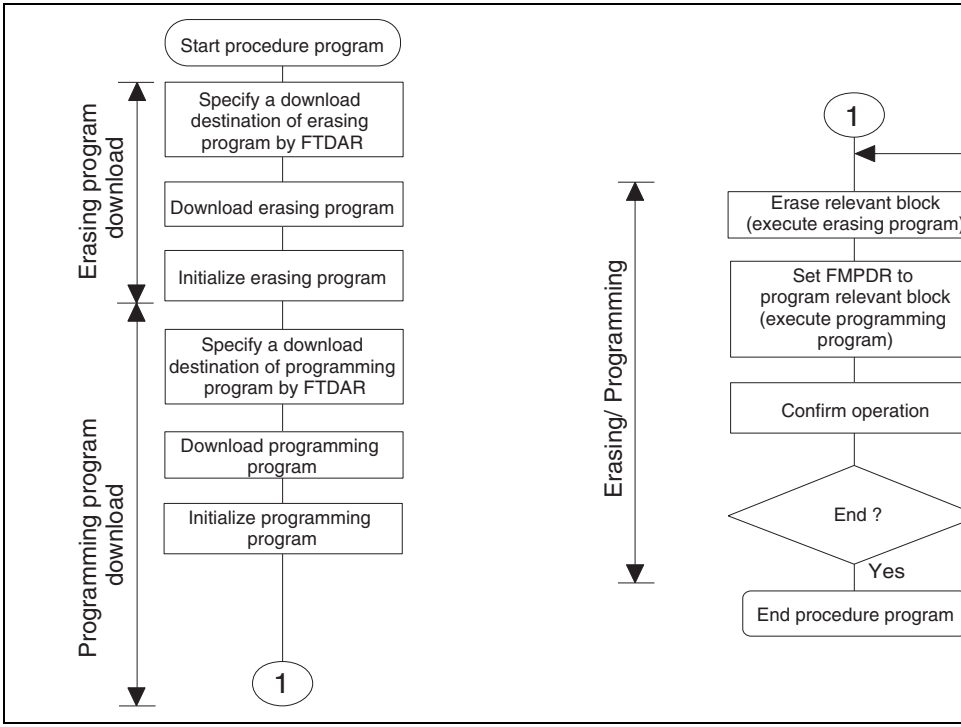


Figure 25.15 Repeating Procedure of Erasing and Programming

In the above procedure, download and initialization are performed only once at the begi

In this kind of operation, note the following:

For the mode pin settings to start up user boot mode, see table 25.5.

When the reset start is executed in user boot mode, the built-in check routine runs. The u
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 bo
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 settin
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 25.16 shows the procedure for programming the user MAT in user boot mode.

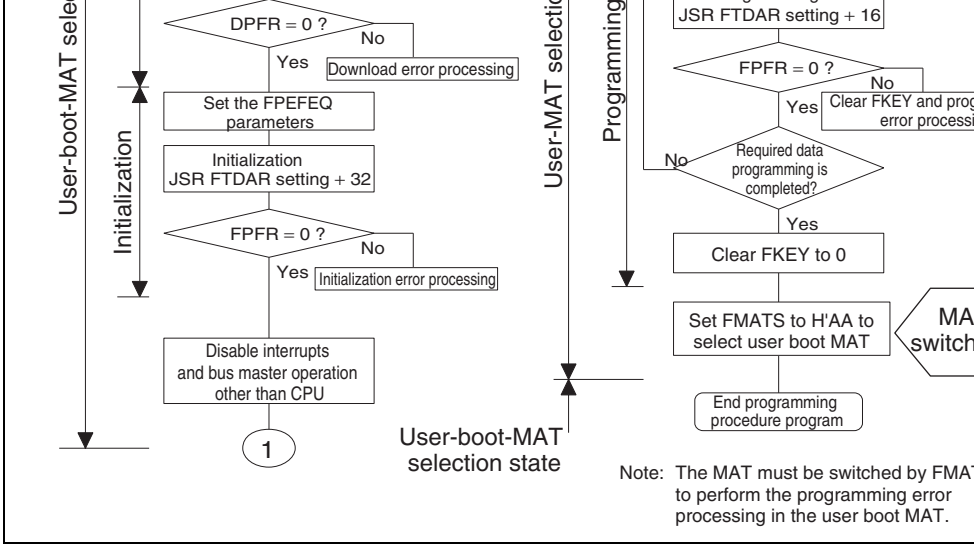


Figure 25.16 Procedure for Programming User MAT in User Boot Mode

The difference between the programming procedures in user program mode and user boo whether the MAT is switched or not as shown in figure 25.16.

In user boot mode, the user boot MAT can be seen in the flash memory space with the us hidden in the background. The user MAT and user boot MAT are switched only while the MAT is being programmed. Because the user boot MAT is hidden while the user MAT is 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 whi MATs are being switched, the LSI is in an unstable state, e.g. access to a MAT is not allo MAT switching is completed, and if an interrupt occurs, from which MAT the interrupt v

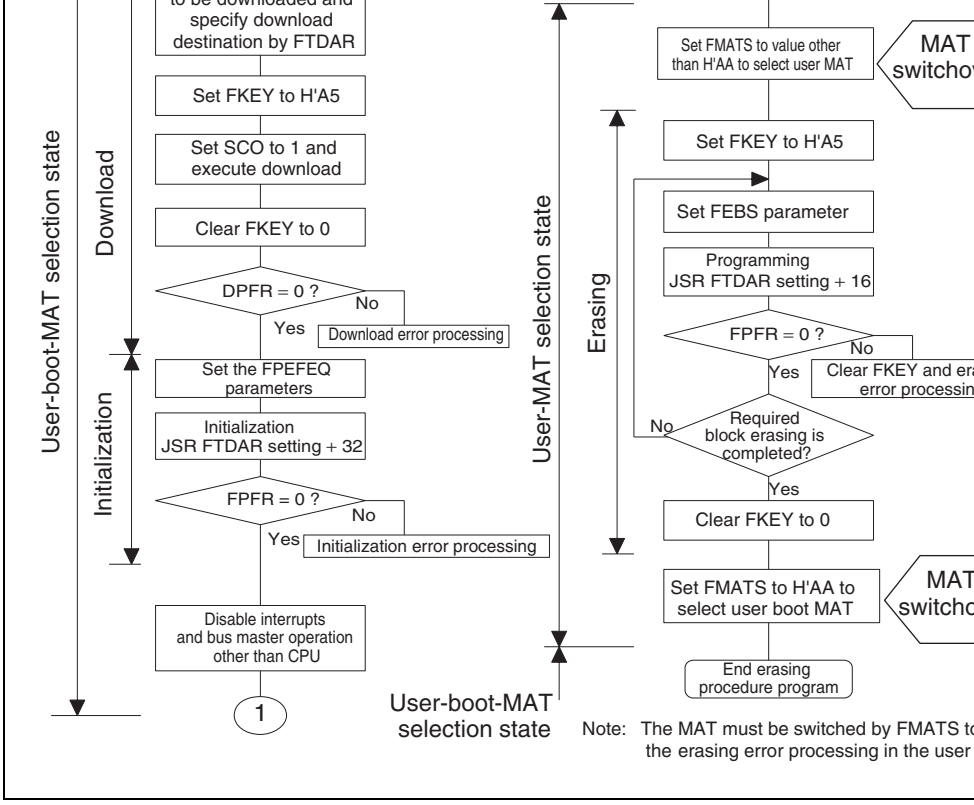


Figure 25.17 Procedure for Erasing User MAT in User Boot Mode

The difference between the erasing procedures in user program mode and user boot mode on whether the MAT is switched or not as shown in figure 25.17.

2. The on-chip programming/erasing program will use 128 bytes at the maximum as a st
make sure that 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 a mode in which the external ad
space is not accessible, such as single-chip mode, the required procedure programs, N
handling vector and NMI handler 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 addr
space.
6. After programming/erasing, the flash memory should be inhibited until FKEY is clear
The reset state ($\overline{RES}$ = 0) must be in place for more than 100 µs when the LSI mode is
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
boot MAT is operated in user-boot mode. The program which switches the MATs sho
executed from the on-chip RAM. See section 25.6, 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 th
memory area, an error will occur even when the data stored is normal. Therefore, the

Programming	Table 25.9 (1)	Table 25.9 (3)
Erasing	Table 25.9 (2)	Table 25.9 (4)

Note: * Programming/Erasing is possible to user MATs.

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	○	○	○	○
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 Program Parameter	○	×	○	○

	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	○	×	○	○
Execution of Erasure	○	×	×	○
Determination of Erasure Result	○	×	○	○

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	○	○	○		○
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	○	×	○	○	

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	○	×	○	○	

Operation after 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 u

user MAT, and the error in programming/erasing is reported in the parameter FPFR.

(including a 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 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, execute erasure and then execute program again.

		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.	○	○

25.5.3 Error Protection

Error protection is a mechanism for aborting programming or erasure when an error occu
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 exces
programming or erasing.

If the microcomputer malfunctions during programming/erasing of the flash memory, the
bit in the FCCS register is set to 1 and the error-protection state is entered, and this aborts
programming or erasure.

The FLER bit is set in the following conditions:

1. When an interrupt such as NMI occurs during programming/erasing.
2. When the flash memory is read during programming/erasing (including a vector read
   instruction fetch).
3. When a SLEEP instruction (including software-standby mode) is executed during
   programming/erasing.

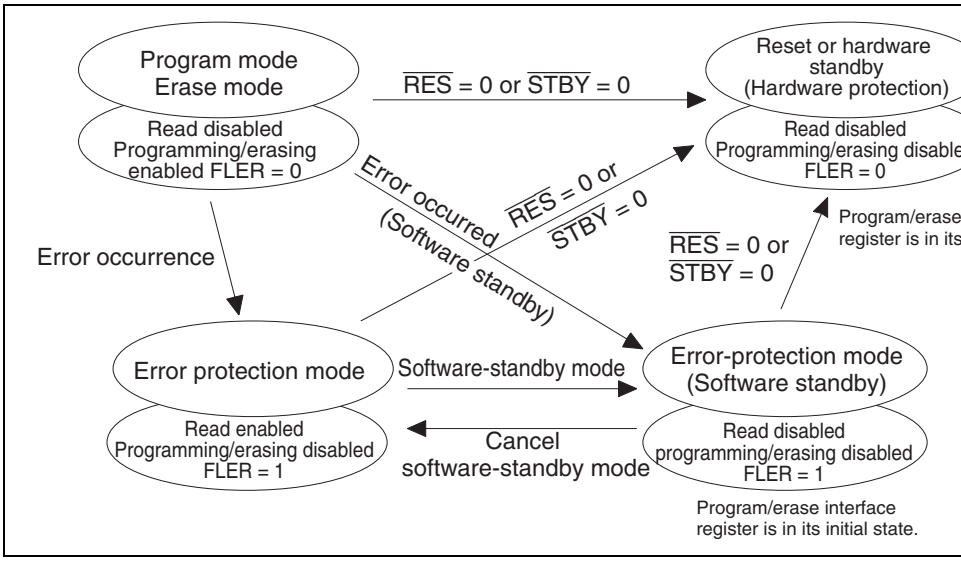


Figure 25.18 Transitions to Error-Protection State

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 set the WEINTE bi FCCS 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 16-Kbyte memory space. If ac goes beyond the 16-Kbyte space, the values read are undefined.

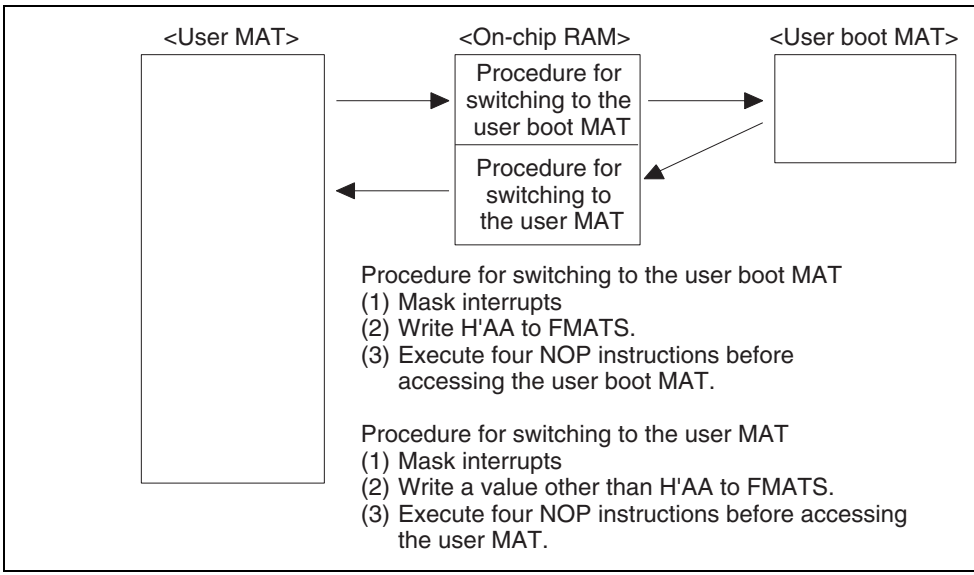


Figure 25.19 Switching between the User MAT and User Boot MAT

MHz input-clock signal.

Notes: 1. For the PROM programmer and the version of its program, see the instructio
for socket adapter.
2. In this LSI, set the programming voltage of the PROM programmer to 3.3 V.

mode enables starting of the boot program and entry to the bit-rate-adjustment state. T
program receives the command from the host to adjust the bit rate. After adjusting the
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 devi
   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 to
   chip 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 progr
   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 host

These boot program states are shown in figure 25.20.

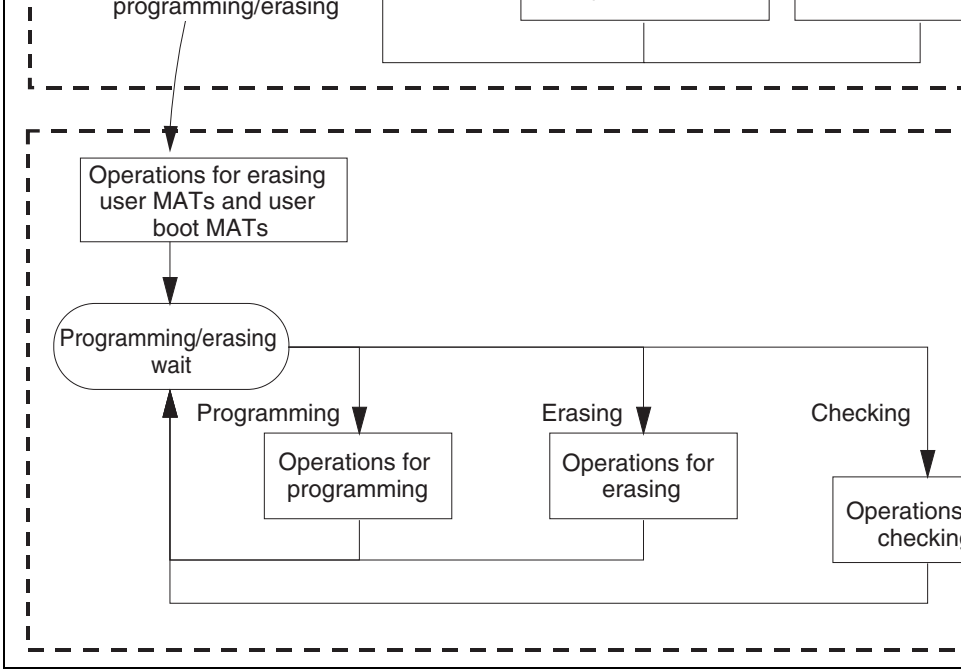


Figure 25.20 Boot Program States

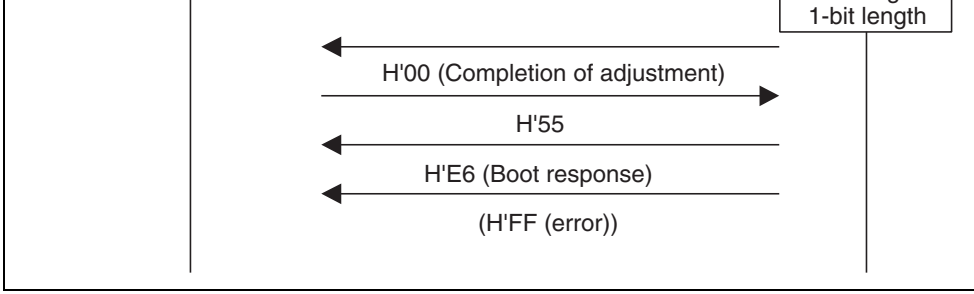


Figure 25.21 Bit-Rate-Adjustment Sequence

(3) Communications Protocol

After adjustment of the bit rate, the protocol for communications between the host and th program is as shown below.

1. 1-byte commands and 1-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 selections responses to inquiries.

   The amount of programming data is not included under this heading because it is dete in another command.
3. Error response

   The error response is a response to inquiries. It consists of an error response and an er and comes two bytes.

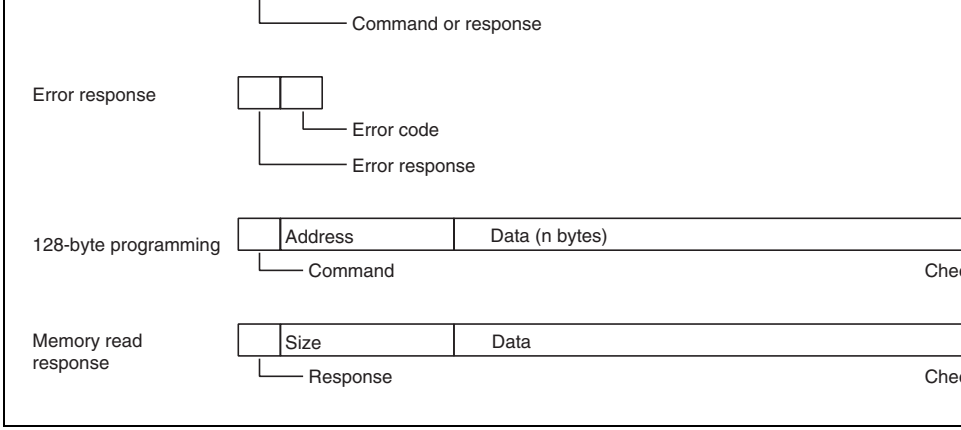


Figure 25.22 Communication Protocol Format

- Command (1 byte): Commands including inquiries, selection, programming, erasing checking
- Response (1 byte): Response to an inquiry
- Size (1 byte): The amount of data for transmission excluding the command, amount and checksum
- Checksum (1 byte): The checksum is calculated so that the total of all values from th command byte to the SUM byte becomes H′00.
- Data (n bytes): Detailed data of a command or response
- Error response (1 byte): Error response to a command
- Error code (1 byte): Type of the error
- Address (4 bytes): Address for programming
- Data (n bytes): Data to be programmed (the size is indicated in the response to the programming unit inquiry.)

Command	Command Name	Description
H'20	Supported Device Inquiry	Inquiry regarding device codes
H'10	Device Selection	Selection of device code
H'21	Clock Mode Inquiry	Inquiry regarding numbers of clock m values of each mode
H'11	Clock Mode Selection	Indication of the selected clock mode
H'22	Multiplication Ratio Inquiry	Inquiry regarding the number of frequ multiplied clock types, the number of multiplication ratios, and the values o multiple
H'23	Operating Clock Frequency Inquiry	Inquiry regarding the maximum and m values of the main clock and peripher
H'24	User Boot MAT Information Inquiry	Inquiry regarding the number of user MATs and the start and last addresse each MAT
H'25	User MAT Information Inquiry	Inquiry regarding the a number of use and the start and last addresses of ea
H'26	Block for Erasing Information Inquiry	Inquiry regarding the number of block the start and last addresses of each b
H'27	Programming Unit Inquiry	Inquiry regarding the unit of programm
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 M entry to programming/erasing state
H'4F	Boot Program Status Inquiry	Inquiry into the operated status of the program

(a) Supported Device Inquiry

The boot program will return the device codes of supported devices and the product cod response to the supported device inquiry.

Command	H'20

- Command, H'20, (1 byte): Inquiry regarding supported devices

Response	H'30	Size	Number of devices	
	Number of characters	Device code		Product name
	...			
	SUM			

- Response, H'30, (1 byte): Response to the supported device inquiry
- Size (1 byte): Number of bytes to be transmitted, excluding the command, size, and that is, the amount of data contributes by the number of devices, characters, device c product names
- Number of devices (1 byte): The number of device types supported by the boot prog
- Number of characters (1 byte): The number of characters in the device codes and bo program's name
- Device code (4 bytes): ASCII code of the supporting product
- Product name (n bytes): Type name of the boot program in ASCII-coded characters
- SUM (1 byte): Checksum

  The checksum is calculated so that the total number of all values from the command the SUM byte becomes H'00.

- SUM (1 byte): Checksum

Response	H'06

- Response, H'06, (1 byte): Response to the device selection command
  ACK will be returned when the device code matches.

Error response	H'90	ERROR

- Error response, H'90, (1 byte): Error response to the device selection command
  ERROR : (1 byte): Error code
  H'11: Sum check error
  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 in

Command	H'21

- Command, H'21, (1 byte): Inquiry regarding clock mode

Response	H'31	Size	Number of modes	Mode	···	SUM

- Response, H'31, (1 byte): Response to the clock-mode inquiry
- Size (1 byte): Amount of data that represents the number of modes and modes
- Number of clock modes (1 byte): The number of supported clock modes
  H'00 indicates no clock mode or the device allows to read the clock mode.
- Mode (1 byte): Values of the supported clock modes (i.e. H'01 means clock mode 1.)
- SUM (1 byte): Checksum

- SUM (1 byte): Checksum

Response	H'06

- Response, H'06, (1 byte): Response to the clock mode selection command
  ACK will be returned when the clock mode matches.

Error Response	H'91	ERROR

- Error response, H'91, (1 byte) : Error response to the clock mode selection comma
- ERROR : (1 byte): Error code
  H'11: Checksum error
  H'22: Clock mode error, that is, the clock mode does not match.

Even if the clock mode numbers are H'00 and H'01 by a clock mode inquiry, the clock m
be selected using these respective values.

...						
SUM						

- Response, H'32, (1 byte): Response to the multiplication ratio inquiry
- Size (1 byte): The amount of data that represents the number of clock sources and multiplication ratios and the multiplication ratios
- Number of types (1 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 (1 byte): The number of multiplication ratios for each (e.g. the number of multiplication ratios to which the main clock can be set and the pe clock can be set.)
- Multiplication ratio (1 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.)

  Division ratio: The inverse of the division ratio, i.e. a negative number (e.g. when the divided by two, the value of division ratio will be H'FE. H'FE = D'-2)

  The number of multiplication ratios returned is the same as the number of multiplicati and as many groups of data are returned as there are types.
- SUM (1 byte): Checksum

clock frequency	frequency
...	
SUM	

- Response, H'33, (1 byte): Response to operating clock frequency inquiry
- Size (1 byte): The number of bytes that represents the minimum values, maximum v the number of frequencies.
- Number of operating clock frequencies (1 byte): The number of supported operating 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 (2 bytes): The minimum value of the m or divided clock frequency.

  The minimum and maximum values represent the values in MHz, valid to the hundre of MHz, and multiplied by 100. (e.g. when the value is 20.00 MHz, it will be 2000, v H'07D0.)
- Maximum value (2 bytes): Maximum value among the multiplied or divided clock fr

  There are as many pairs of minimum and maximum values as there are operating clo frequencies.
- SUM (1 byte): Checksum

- Response, H'34, (1 byte): Response to user boot MAT information inquiry
- Size (1 byte): The number of bytes that represents the number of areas, area-start addr and area-last address
- Number of Areas (1 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 (4 byte): Start address of the area
- Area-last address (4 byte): Last address of the area
  There are as many groups of data representing the start and last addresses as there are
- SUM (1 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, (1 byte): Inquiry regarding user MAT information

Response

H'35	Size	Number of areas	
Start address area			Last address area
...			
SUM			

- Response, H'35, (1 byte): Response to the user MAT information inquiry
- Size (1 byte): The number of bytes that represents the number of areas, area-start addr area-last address
- Number of areas (1 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 (4 bytes): Start address of the area

	Block start address	Block last address
	...	
	SUM	

- Response, H'36, (1 byte): Response to the number of erased blocks and addresses
- Size (three bytes): The number of bytes that represents the number of blocks, block-s addresses, and block-last addresses.
- Number of blocks (1 byte): The number of erased blocks
- Block start address (4 bytes): Start address of a block
- Block last Address (4 bytes): Last address of a block
  There are as many groups of data representing the start and last addresses as there ar
- SUM (1 byte): Checksum

(j) Programming Unit Inquiry

The boot program will return the programming unit used to program data.

Command	H'27

- Command, H'27, (1 byte): Inquiry regarding programming unit

Response	H'37	Size	Programming unit	SUM

- Response, H'37, (1 byte): Response to programming unit inquiry
- Size (1 byte): The number of bytes that indicate the programming unit, which is fixe
- Programming unit (2 bytes): A unit for programming
  This is the unit for reception of programming.
- SUM (1 byte): Checksum

- Size (1 byte): The number of bytes that represents the bit rate, input frequency, numb multiplication ratios, and multiplication ratio
- Bit rate (2 bytes): New bit rate
  One hundredth of the value (e.g. when the value is 19200 bps, it will be 192, which is H′00C0.)
- Input frequency (2 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, it will be 2000, which is H'07D0.)
- Number of multiplication ratios (1 byte): The number of multiplication ratios to whic device can be set.
- Multiplication ratio 1 (1 byte) : The value of multiplication or division ratios for the operating frequency
  Multiplication ratio (1 byte): The value of the multiplication ratio (e.g. when the cloc frequency is multiplied by four, the multiplication ratio will be H'04.)
  Division ratio: The inverse of the division ratio, as a negative number (e.g. when the frequency is divided by two, the value of division ratio will be H'FE. H'FE = D'-2)
- Multiplication ratio 2 (1 byte): The value of multiplication or division ratios for the p frequency
  Multiplication ratio (1 byte): The value of the multiplication ratio (e.g. when the cloc frequency is multiplied by four, the multiplication ratio will be H'04.)
  (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)
- SUM (1 byte): Checksum

Response	H'06

- Response, H'06, (1 byte): Response to selection of a new bit rate
  When it is possible to set the bit rate, the response will be ACK.

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 range minimum to maximum frequencies which matches the clock modes of the specified dev 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 that i the clock modes of the specified device. When the value is out of this range, an multipli ratio error is generated.

3. Operating frequency

Operating frequency is calculated from the received value of the input frequency and the multiplication or division ratio. The input frequency is input to the LSI and the LSI is op 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 rang minimum to maximum frequencies which are available with the clock modes of the spec 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, (1 byte): Confirmation of a new bit rate

Response	H'06

- Response, H'06, (1 byte): Response to confirmation of a new bit rate

The sequence of new bit-rate selection is shown in figure 25.23.

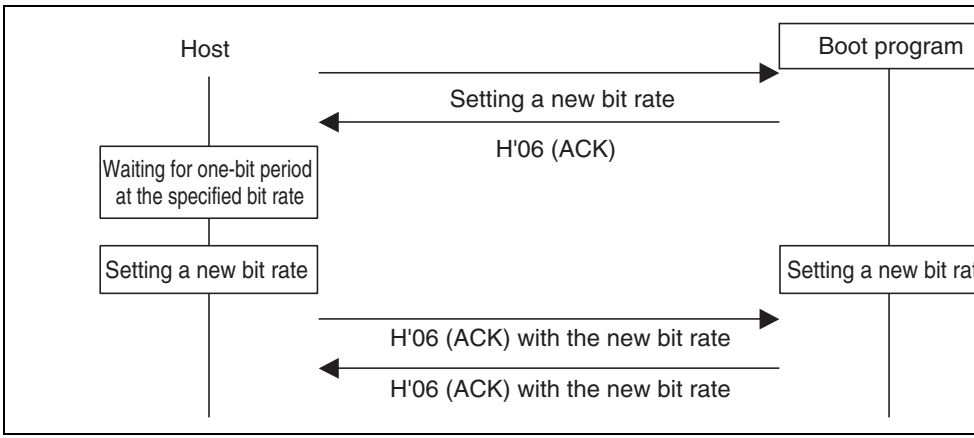


Figure 25.23 New Bit-Rate Selection Sequence

Command H'40

- Command, H'40, (1 byte): Transition to programming/erasing state

Response	H'06

- Response, H'06, (1 byte): Response to transition to 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, (1 byte): Error response for user boot MAT blank check
- Error code, H'51, (1 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 i
or a command is unacceptable. Issuing a clock-mode selection command before a devic
or an inquiry command after the transition to programming/erasing state command, are

Error Response	H'80	H'xx

- Error response, H'80, (1 byte): Command error
- Command, H'xx, (1 byte): Received command

be made, such as the multiplication-ratio inquiry (H'22) or operating frequency inquir which are needed for a new bit-rate selection.

6. A new bit rate should be selected with the new bit-rate selection (H'3F) command, acc to the returned information on multiplication ratios and operating frequencies.
7. After selection of the device and clock mode, the information of the user boot MAT a MAT should be made to inquire about the user boot MATs information inquiry (H'24 MATs information inquiry (H'25), erased block information inquiry (H'26), and progr unit inquiry (H'27).
8. After making inquiries and selecting a new bit rate, issue the transition to programming/erasing state command (H'40). The boot program will then enter the programming/erasing state.

H'43	User MAT programming selection	Transfers the user MAT programm program
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
H'4B	User MAT sum check	Checks the checksum of the user
H'4C	User boot MAT blank check	Checks whether the contents of th boot MAT are blank
H'4D	User MAT blank check	Checks whether the contents of th MAT are blank
H'4F	Boot program status inquiry	Inquires into the boot program's st

programming command. The 128-byte programming command that follows the select
command represents the data programmed according to the method specified by the se
command. When more than 128-byte data is programmed, 128-byte commands should
repeatedly be executed. Sending a 128-byte programming command with H'FFFFFFF
address will stop the programming. On completion of programming, the boot program
wait for selection of programming or erasing.

Where the sequence of programming operations that is executed includes programmin
another method or of another MAT, the procedure must be repeated from the program
selection command.

The sequence for programming-selection and 128-byte programming commands is sh
figure 25.24.

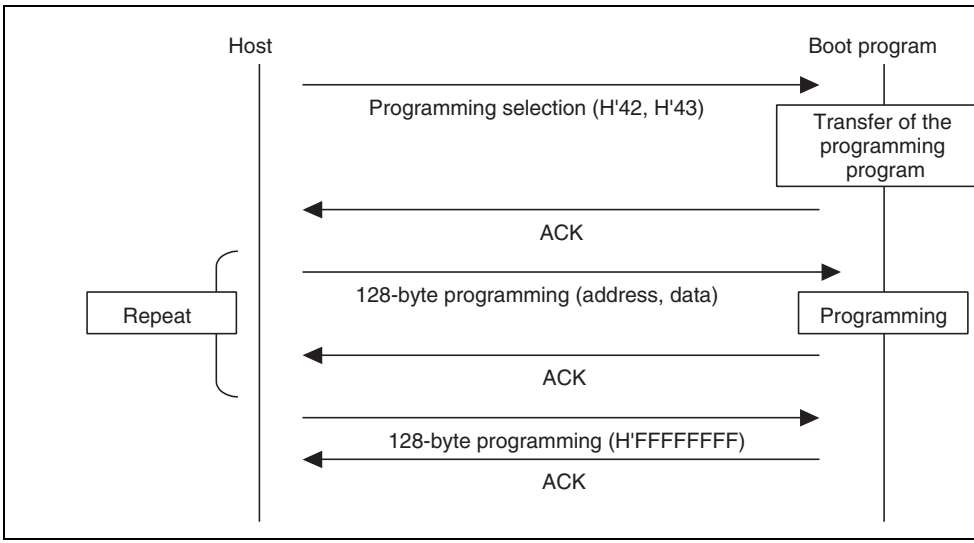


Figure 25.24 Programming Sequence

Error Response	H'C2	ERROR

- Error response : H'C2 (1 byte): Error response to user boot MAT programming selec
- ERROR : (1 byte): Error code
  H'54: Selection processing error (transfer error occurs and processing is not complet

- User MAT programming selection
  The boot program will transfer a program for programming. The data is programmed
  user MATs by the transferred program for programming.

Command	H'43

- Command, H'43, (1 byte): User MAT programming selection

Response	H'06

- Response, H'06, (1 byte): Response to user MAT programming selection
  When the programming program has been transferred, the boot program will return A

Error Response	H'C3	ERROR

- Error response : H'C3 (1 byte): Error response to user MAT programming selection
- ERROR : (1 byte): Error code
  H'54: Selection processing error (transfer error occurs and processing is not complet

(b) 128-byte programming

The boot program will use the programming program transferred by the programming se
program the user boot MATs or user MATs in response to 128-byte programming.

Command	H'50	Address						
	Data	···						
	···							
	SUM							

On completion of programming, the boot program will return ACK.

Error Response	H'D0	ERROR

- Error response, H'D0, (1 byte): Error response for 128-byte programming
- ERROR: (1 byte): Error code

  H'11: Checksum Error

  H'2A: Address Error

  H'53: Programming error

  A programming error has occurred and programming cannot be continued.

The specified address should match the unit for programming of data. For example, when programming is in 128-byte units, the lower 8 bits of the address should be H'00 or H'80. When there are less than 128 bytes of data to be programmed, the host should fill the rest H'FF.

Sending the 128-byte programming command with the address of H'FFFFFFFF will stop programming operation. The boot program will interpret this as the end of the programmi wait for selection of programming or erasing.

Command	H'50	Address	SUM

- Command, H'50, (1 byte): 128-byte programming
- Programming Address (4 bytes): End code is H'FF, H'FF, H'FF, H'FF.
- SUM (1 byte): Checksum

Response	H'06

- Response, H'06, (1 byte): Response to 128-byte programming
  On completion of programming, the boot program will return ACK.

Firstly, erasure is selected by the erasure selection command and the boot program then
specified block. The command should be repeatedly executed if two or more blocks are
erased. Sending a block-erasure command from the host with the block number H'FF wi
erasure operating. On completion of erasing, the boot program will wait for selection of
programming or erasing.

The sequences of issuing the erasure selection command and block-erasure command ar
figure 25.25.

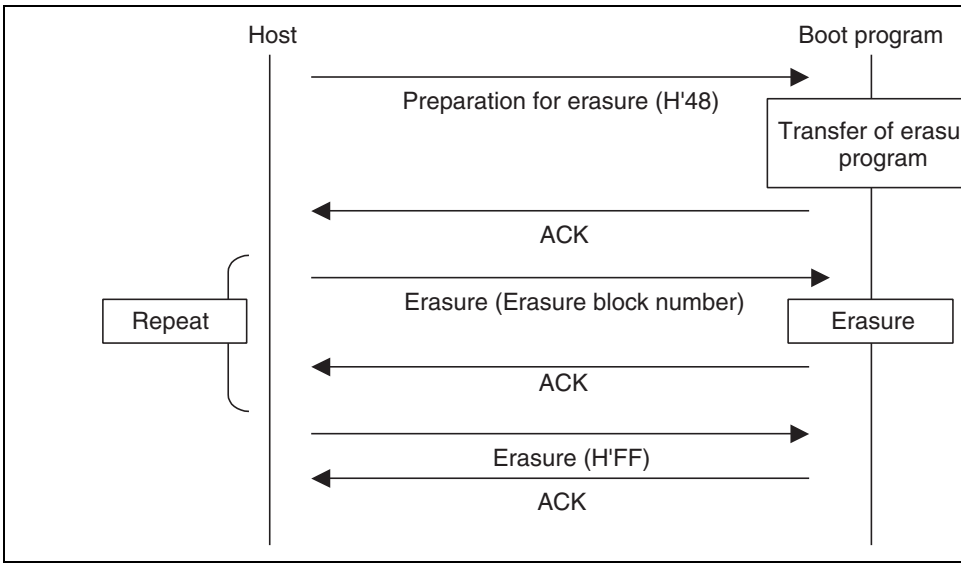


Figure 25.25 Erasure Sequence

Error Response	H'C8	ERROR

- Error Response, H'C8, (1 byte): Error response to erasure selection
- ERROR: (1 byte): Error code

  H'54: Selection processing error (transfer error occurs and processing is not complet

(b) Block Erasure

The boot program will erase the contents of the specified block.

Command	H'58	Size	Block number	SUM

- Command, H'58, (1 byte): Erasure
- Size (1 byte): The number of bytes that represents the erasure block number
  This is fixed to 1.
- Block number (1 byte): Number of the block to be erased
- SUM (1 byte): Checksum

Response	H'06

- Response, H'06, (1 byte): Response to Erasure
  After erasure has been completed, the boot program will return ACK.

Error Response	H'D8	ERROR

- Error Response, H'D8, (1 byte): Response to Erasure
- ERROR (1 byte): Error code

  H'11: Sum check error
  H'29: Block number error
  Block number is incorrect.
  H'51: Erasure error
  An error has occurred during erasure.

Response	H'06

- Response, H'06, (1 byte): Response to end of erasure (ACK)
  When erasure is to be performed after the block number H'FF has been sent, the proc
  should be executed from the erasure selection command.

(11) 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 (fi
- Area (1 byte)
  H'00: User boot MAT
  H'01: User MAT
  An address error occurs when the area setting is incorrect.
- Read address (4 bytes): Start address to be read from
- Read size (4 bytes): Size of data to be read
- SUM (1 byte): Checksum

Response	H'52	Read size						
	Data	···						
	SUM							

- 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

The boot program will return the byte-by-byte total of the contents of the bytes of the use MAT, as a 4-byte value.

Command	H'4A

- Command, H'4A, (1 byte): Sum check for user-boot MAT

Response	H'5A	Size	Checksum of user boot program	SUM

- Response, H'5A, (1 byte): Response to the sum check of user-boot MAT
- Size (1 byte): The number of bytes that represents the checksum
  This is fixed to 4.
- Checksum of user boot program (4 bytes): Checksum of user boot MATs
  The total of the data is obtained in byte units.
- SUM (1 byte): Sum check for data being transmitted

(13) User MAT Sum Check

The boot program will return the byte-by-byte total of the contents of the bytes of the use

Command	H'4B

- Command, H'4B, (1 byte): Sum check for user MAT

Response	H'5B	Size	Checksum of user program	SUM

- Response, H'5B, (1 byte): Response to the sum check of the user MAT
- Size (1 byte): The number of bytes that represents the checksum
  This is fixed to 4.
- Checksum of user boot program (4 bytes): Checksum of user MATs
  The total of the data is obtained in byte units.
- SUM (1 byte): Sum check for data being transmitted

Error Response	H'CC	H'52

- Error Response, H'CC, (1 byte): Response to blank check for user boot MAT
- Error Code, H'52, (1 byte): Erasure has not been completed.

(15) User MAT Blank Check

The boot program will check whether or not all user MATs are blank and return the resu

Command	H'4D

- Command, H'4D, (1 byte): Blank check for user MATs

Response	H'06

- Response, H'06, (1 byte): Response to the blank check for user boot MATs
  If the contents of all user MATs are blank (H'FF), the boot program will return ACK

Error Response	H'CD	H'52

- Error Response, H'CD, (1 byte): Error response to the blank check of user MATs.
- Error code, H'52, (1 byte): Erasure has not been completed.

- Status (1 byte): State of the boot program
- ERROR (1 byte): Error status

  ERROR = 0 indicates normal operation.
  ERROR = 1 indicates error has occurred.

- SUM (1 byte): Sum check

Table 25.14 Status Code

Code	Description
H'11	Device Selection Wait
H'12	Clock Mode Selection Wait
H'13	Bit Rate Selection Wait
H'1F	Programming/Erasing State Transition Wait (Bit rate selection is completed)
H'31	Programming State for Erasure
H'3F	Programming/Erasing Selection Wait (Erasure is completed)
H'4F	Programming Data Receive Wait (Programming is completed)
H'5F	Erasure Block Specification Wait (Erasure is completed)

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

PROM programmer that supports the 512-Kbyte flash memory on-chip MCU device a
Do not set the programmer to HN28F101 or the programming voltage to 5.0 V. Use o
specified socket adapter. If other adapters are used, the product may be damaged.

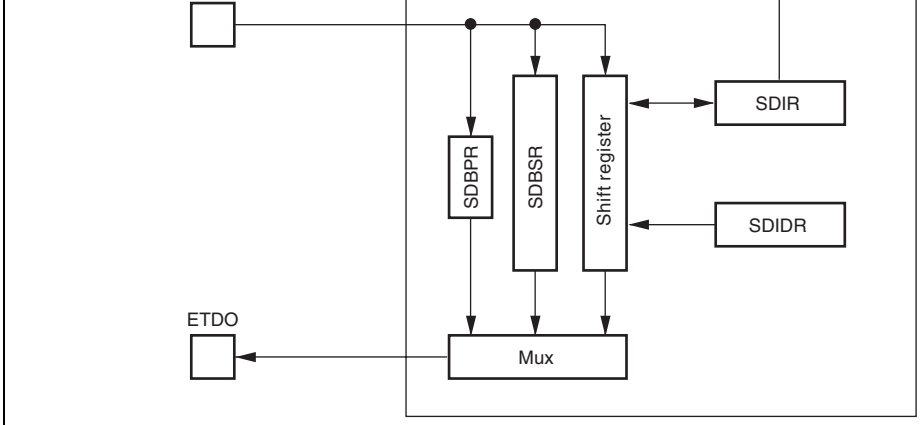
5. Do not remove the chip from the PROM programmer nor input a reset signal during programming/erasing. As a high voltage is applied to the flash memory during programming/erasing, doing so may damage or destroy flash memory permanently. If executed accidentally, reset must be released after the reset input period of 100 μs wh longer than normal.
6. The flash memory is not accessible until FKEY is cleared after programming/erasing completes. If this LSI is restarted by a reset immediately after programming/erasing h finished, secure the reset period (period of $\overline{RES}$ = 0) of more than 100 μs. Though tra the reset state or hardware standby state during programming/erasing is prohibited, if executed accidentally, reset must be released after the reset input period of 100 μs wh longer than normal.
7. At powering on or off the Vcc power supply, fix the $\overline{RES}$ pin to low and set the flash to hardware protection state. This power on/off timing must also be satisfied at a pow power-on caused by a power failure and other factors.
8. Program the area with 128-byte programming-unit blocks in on-board programming o programmer mode only once. Perform programming in the state where the programm block is fully erased.
9. When the chip is to be reprogrammed with the programmer after execution of progra erasure in on-board programming mode, it is recommend that automatic programming performed after execution of automatic erasure.
10. To write data or programs to the flash memory, data or programs must be allocated to addresses higher than that of the external interrupt vector table (H'000040) and H'FF r written to the areas that are reserved for the system in the exception handling vector ta

FCCS that is used for a download request or FMATS that is used for MAT switching 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 get out control. Do not use DTC to program flash related registers.

14. A programming/erasing program for flash memory used in the conventional H8S F-Z microcomputer which does not support download of the on-chip program by a SCO request cannot run in this LSI. Be sure to download the on-chip program to execute programming/erasing of flash memory in this LSI.
15. Unlike the conventional H8S F-ZTAT microcomputer, no countermeasures are avail runaway by WDT during programming/erasing. Prepare countermeasures (e.g. use o periodic timer interrupts) for WDT with taking the programming/erasing time into consideration as required.

26.1 Features

- Five test pins (ETCK, ETDI, ETDO, ETMS, and $\overline{\text{ETRST}}$)
- TAP controller
- Six instructions
  BYPASS mode
  EXTEST mode
  SAMPLE/PRELOAD mode
  CLAMP mode
  HIGHZ mode
  IDCODE mode
  (These instructions are test modes corresponding to IEEE 1149.1.)



[Legend]
SDIR: Instruction register
SDBPR: Bypass register
SDBSR: Boundary scan register
SDIDR: ID code register

Figure 26.1 JTAG Block Diagram

			with a duty cycle close to 50% should b For details, see section 31, Electrical Characteristics.
Test mode select	ETMS	Input	Test mode select input Sampled on the rise of the ETCK pin. T pin controls the internal state of the TA controller.
Test data input	ETDI	Input	Serial data input Performs serial input of instructions an JTAG registers. ETDI is sampled on th the ETCK pin.
Test data output	ETDO	Output	Serial data output Performs serial output of instructions a from JTAG registers. Transfer is perfor synchronization with the ETCK pin. If t output, the ETDO pin goes to the high-impedance state.
Test reset	$\overline{\text{ETRST}}$	Input	Test reset input signal Initializes the JTAG asynchronously.

input pin (ETDI). Data from SDIR can be output via the test data output pin (ETDO). The
register (SDBPR) is a 1-bit register to which the ETDI and ETDO pins are connected in E
CLAMP, or HIGHZ mode. The boundary scan register (SDBSR) is a 346-bit register in H
group, 333-bit register in H8S/2462 group to which the ETDI and ETDO pins are connec
SAMPLE/PRELOAD or EXTEST mode. The ID code register (SDIDR) is a 32-bit regist
fixed code can be output via the ETDO pin in IDCODE mode. All registers cannot be acc
directly by the CPU.

Table 26.2 shows the kinds of serial transfer possible with each JTAG register.

Table 26.2 JTAG Register Serial Transfer

Register	Serial Input	Serial Output
SDIR	Possible	Possible
SDBPR	Possible	Possible
SDBSR	Possible	Possible
SDIDR	Impossible	Possible

31	TS3	1	R/W	Test Set Bits
30	TS2	1	R/W	0000: EXTEST mode
29	TS1	1	R/W	0001: Setting prohibited
28	TS0	0	R/W	0010: CLAMP mode
				0011: HIGHZ mode
				0100: SAMPLE/PRELOAD mode
				0101: Setting prohibited
				: :
				1101: Setting prohibited
				1110: IDCODE mode (Initial value)
				1111: BYPASS mode
27 to 14	—	All 0	R	Reserved These bits are always read as 0 and cannot be m
13	—	1	R	Reserved This bit is always read as 1 and cannot be modi
12	—	0	R	Reserved This bit is always read as 0 and cannot be modi
11	—	1	R	Reserved This bit is always read as 1 and cannot be modi
10 to 1	—	All 0	R	Reserved These bits are always read as 0 and cannot be m
0	—	1	R	Reserved This bit is always read as 1 and cannot be modi

IEEE1149.1 standard can be performed.

Table 26.3 shows the relationship between the pins of this LSI and the boundary scan reg

		Enable	344			—
		Output	343			—
B1	P46	Input	342	F4	NC	—
		Enable	341			—
		Output	340			—
C2	P47	Input	339	F3	VCL	—
		Enable	338			—
		Output	337			—
D3	P56	Input	336	F1	$\overline{\text{MD2}}$	Input
		Enable	335			—
		Output	334			—
C1	P57	Input	333	F2	P51	Input
		Enable	332			Enable
		Output	331			Output
D2	VSS	—	—	G4	P50	Input
		—	—			Enable
		—	—			Output
E4	$\overline{\text{RES}}$	—	—	G3	P97	Input
		—	—			Enable
		—	—			Output
D1	MD1	Input	330	G1	P96	Input
		—	—			Enable
		—	—			Output

		Output	304			Output	2
H1	NC	—	—	L3	NC	—	-
		—	—			—	-
		—	—			—	-
H2	P92	Input	303	L1	PC2	Input	2
		Enable	302			Enable	2
		Output	301			Output	2
J4	P91	Input	300	L2	NC	—	-
		Enable	299			—	-
		Output	298			—	-
J3	P90	Input	297	L4	PC1	Input	2
		Enable	296			Enable	2
		Output	295			Output	2
J1	NC	—	—	M1	NC	—	-
		—	—			—	-
		—	—			—	-
J2	PC7	Input	294	M2	PC0	Input	2
		Enable	293			Enable	2
		Output	292			Output	2
K4	PC6	Input	291	M3	PA7	Input	2
		Enable	290			Enable	2
		Output	289			Output	2

		—	—				—
P1	PA4	Input	261	R4	NC	—	
		Enable	260			—	
		Output	259			—	
P2	PA3	Input	258	N5	P87	Input	
		Enable	257			Enable	
		Output	256			Output	
R1	NC	—	—	P5	P86	Input	
		—	—			Enable	
		—	—			Output	
N3	PA2	Input	255	R5	P85	Input	
		Enable	254			Enable	
		Output	253			Output	
R2	NC	—	—	M6	P84	Input	
		—	—			Enable	
		—	—			Output	
P3	PA1	Input	252	N6	P83	Input	
		Enable	251			Enable	
		Output	250			Output	
N4	NC	—	—	R6	P82	Input	
		—	—			Enable	
		—	—			Output	

		—	—			—	
R7	PE7	Input	222	M10	PD7	Input	1
		Enable	221			Enable	1
		Output	220			Output	1
P7	NC	—	—	N10	PD6	Input	1
		—	—			Enable	1
		—	—			Output	1
M8	PE6	Input	219	R10	PD5	Input	1
		Enable	218			Enable	1
		Output	217			Output	1
N8	PE5	Input	216	P10	PD4	Input	1
		Enable	215			Enable	1
		Output	214			Output	1
R8	PE4	Input	213	N11	PD3	Input	1
		Enable	212			Enable	1
		Output	211			Output	1
P8	PE3	Input	210	R11	PD2	Input	1
		Enable	209			Enable	1
		Output	208			Output	1
M9	PE2	Input	207	P11	PD1	Input	1
		Enable	206			Enable	1
		Output	205			Output	1

		—	—				—
N12	P70	Input	174	M13	P60	Input	
		—	—			Enable	
		—	—			Output	
R13	P71	Input	173	N15	P61	Input	
		—	—			Enable	
		—	—			Output	
M12	P72	Input	172	M14	P62	Input	
		—	—			Enable	
		—	—			Output	
P13	P73	Input	171	L12	P63	Input	
		—	—			Enable	
		—	—			Output	
R14	P74	Input	170	M15	P64	Input	
		—	—			Enable	
		—	—			Output	
P14	P75	Input	169	L13	P65	Input	
		—	—			Enable	
		—	—			Output	
R15	P76	Input	168	L14	P66	Input	
		—	—			Enable	
		—	—			Output	

		—	—			—	-
K15	USD-	—	—		G12	$\overline{\text{ETRST}}$	—	-
		—	—				—	-
		—	—				—	-
K14	USD+	—	—		G13	PF2	Input	1
		—	—				Enable	1
		—	—				Output	1
J12	NC	—	—		G15	PF1	Input	1
		—	—				Enable	1
		—	—				Output	1
J13	DrVSS	—	—		G14	PF0	Input	1
		—	—				Enable	1
		—	—				Output	1
J15	PUPDPLS	—	—		F12	NC	—	-
		—	—				—	-
		Output	142				—	-
J14	VBUS	—	—		F13	VSS	—	-
		—	—				—	-
		—	—				—	-
H12	ETMS	—	—		F15	P27	Input	1
		—	—				Enable	1
		—	—				Output	1

		Output	121			Output
E14	P23	Input	120	A15	P11	Input
		Enable	119			Enable
		Output	118			Output
E12	P22	Input	117	C13	VSS	—
		Enable	116			—
		Output	115			—
D15	P21	Input	114	A14	P10	Input
		Enable	113			Enable
		Output	112			Output
D14	P20	Input	111	B13	PB7	Input
		Enable	110			Enable
		Output	109			Output
D13	P17	Input	108	C12	PB6	Input
		Enable	107			Enable
		Output	106			Output
C15	P16	Input	105	A13	PB5	Input
		Enable	104			Enable
		Output	103			Output
D12	P15	Input	102	B12	PB4	Input
		Enable	101			Enable
		Output	100			Output

		Output	64			Output	3
B11	PB0	Input	63	D8	P40	Input	3
		Enable	62			Enable	3
		Output	61			Output	3
A11	VCC	—	—	C8	P41	Input	3
		—	—			Enable	3
		—	—			Output	3
D10	P30	Input	60	A8	P42	Input	3
		Enable	59			Enable	2
		Output	58			Output	2
C10	P31	Input	57	B8	P43	Input	2
		Enable	56			Enable	2
		Output	55			Output	2
A10	P32	Input	54	D7	PEVref	—	-
		Enable	53			—	-
		Output	52			—	-
B10	P33	Input	51	C7	PECI	—	-
		Enable	50			—	-
		Output	49			—	-
D9	P34	Input	48	A7	P52	Input	2
		Enable	47			Enable	2
		Output	46			Output	2

		Output	15			—
A6	P55	Input	14	A3	VSS	—
		Enable	13			—
		Output	12			—
B6	P44	Input	11	D4	PF3	Input
		Enable	10			Enable
		Output	9			Output
C5	VCC	—	—	B3	$\overline{\text{RESO}}$	—
		—	—			—
		—	—			—
A5	UXTAL	—	—	A2	XTAL	—
		—	—			—
		—	—			—
B5	UEXTAL	—	—	B2	EXTAL	—
		—	—			—
		—	—			—
D5	UXSEL	—	—			to ETDO
		—	—			
		—	—			

Though the pin no. for the H8S/2462 Group and the H8S/2463 Group differs, the bit no.
products is the same. The following table is listed with the pin no. of the H8S/2462 Grou

2	P45	Input	332	12	$\overline{\text{STBY}}$	—	-
		Enable	331			—	-
		Output	330			—	-
3	P46	Input	329	13	VCL	—	-
		Enable	328			—	-
		Output	327			—	-
4	P47	Input	326	14	$\overline{\text{MD2}}$	Input	3
		Enable	325			—	-
		Output	324			—	-
5	P56	Input	323	15	P51	Input	3
		Enable	322			Enable	3
		Output	321			Output	3
6	P57	Input	320	16	P50	Input	3
		Enable	319			Enable	3
		Output	318			Output	3
7	VSS	—	—	17	P97	Input	3
		—	—			Enable	3
		—	—			Output	3
8	$\overline{\text{RES}}$	—	—	18	P96	Input	3
		—	—			Enable	3
		—	—			Output	3
9	MD1	Input	317	19	P95	Input	2
		—	—			Enable	2
		—	—			Output	2

		Output	288			Output	
23	P91	Input	287	33	PA7	Input	
		Enable	286			Enable	
		Output	285			Output	
24	P90	Input	284	34	PA6	Input	
		Enable	283			Enable	
		Output	282			Output	
25	PC7	Input	281	35	PA5	Input	
		Enable	280			Enable	
		Output	279			Output	
26	PC6	Input	278	36	VCC	—	
		Enable	277			—	
		Output	276			—	
27	PC5	Input	275	37	PA4	Input	
		Enable	274			Enable	
		Output	273			Output	
28	PC4	Input	272	38	PA3	Input	
		Enable	271			Enable	
		Output	270			Output	
29	PC3	Input	269	39	PA2	Input	
		Enable	268			Enable	
		Output	267			Output	

		—	—			Output	2
43	P87	Input	233	53	PE5	Input	2
		Enable	232			Enable	2
		Output	231			Output	2
44	P86	Input	230	54	PE4	Input	2
		Enable	229			Enable	1
		Output	228			Output	1
45	P85	Input	227	55	PE3	Input	1
		Enable	226			Enable	1
		Output	225			Output	1
46	P84	Input	224	56	PE2	Input	1
		Enable	223			Enable	1
		Output	222			Output	1
47	P83	Input	221	57	PE1	Input	1
		Enable	220			Enable	1
		Output	219			Output	1
48	P82	Input	218	58	PE0	Input	1
		Enable	217			Enable	1
		Output	216			Output	1
49	P81	Input	215	59	PD7	Input	1
		Enable	214			Enable	1
		Output	213			Output	1

		Output	174			—	
63	PD3	Input	173	73	P75	Input	
		Enable	172			—	
		Output	171			—	
64	PD2	Input	170	74	P76	Input	
		Enable	169			—	
		Output	168			—	
65	PD1	Input	167	75	P77	Input	
		Enable	166			—	
		Output	165			—	
66	PD0	Input	164	76	AVCC	—	
		Enable	163			—	
		Output	162			—	
67	AVSS	—	—	77	AVref	—	
		—	—			—	
		—	—			—	
68	P70	Input	161	78	P60	Input	
		—	—			Enable	
		—	—			Output	
69	P71	Input	160	79	P61	Input	
		—	—			Enable	
		—	—			Output	

		Output	139			Output	1
83	P65	Input	138	93	PF0	Input	1
		Enable	137			Enable	1
		Output	136			Output	1
84	P66	Input	135	94	VSS	—	–
		Enable	134			—	–
		Output	133			—	–
85	P67	Input	132	95	P27	Input	1
		Enable	131			Enable	1
		Output	130			Output	1
86	VCC	—	—	96	P26	Input	1
		—	—			Enable	1
		—	—			Output	1
87	ETMS	—	—	97	P25	Input	1
		—	—			Enable	1
		—	—			Output	1
88	ETDO	—	—	98	P24	Input	1
		—	—			Enable	1
		—	—			Output	1
89	ETDI	—	—	99	P23	Input	1
		—	—			Enable	1
		—	—			Output	1

		Output	100			Output
103	P17	Input	99	113	PB6	Input
		Enable	98			Enable
		Output	97			Output
104	P16	Input	96	114	PB5	Input
		Enable	95			Enable
		Output	94			Output
105	P15	Input	93	115	PB4	Input
		Enable	92			Enable
		Output	91			Output
106	P14	Input	90	116	PB3	Input
		Enable	89			Enable
		Output	88			Output
107	P13	Input	87	117	PB2	Input
		Enable	86			Enable
		Output	85			Output
108	P12	Input	84	118	PB1	Input
		Enable	83			Enable
		Output	82			Output
109	P11	Input	81	119	PB0	Input
		Enable	80			Enable
		Output	79			Output

		Output	46			Output	1
123	P32	Input	45	133	PEVref	—	–
		Enable	44			—	–
		Output	43			—	–
124	P33	Input	42	134	PECI	—	–
		Enable	41			—	–
		Output	40			—	–
125	P34	Input	39	135	P52	Input	1
		Enable	38			Enable	1
		Output	37			Output	1
126	P35	Input	36	136	P53	Input	1
		Enable	35			Enable	1
		Output	34			Output	1
127	P36	Input	33	137	FWE	Input	9
		Enable	32			Enable	–
		Output	31			Output	–
128	P37	Input	30	138	P54	Input	8
		Enable	29			Enable	7
		Output	28			Output	6
129	P40	Input	27	139	P55	Input	5
		Enable	26			Enable	4
		Output	25			Output	3

		—	—
143	XTAL	—	—
		—	—
		—	—
144	EXTAL	—	—
		—	—
		—	—
		to ETDO	

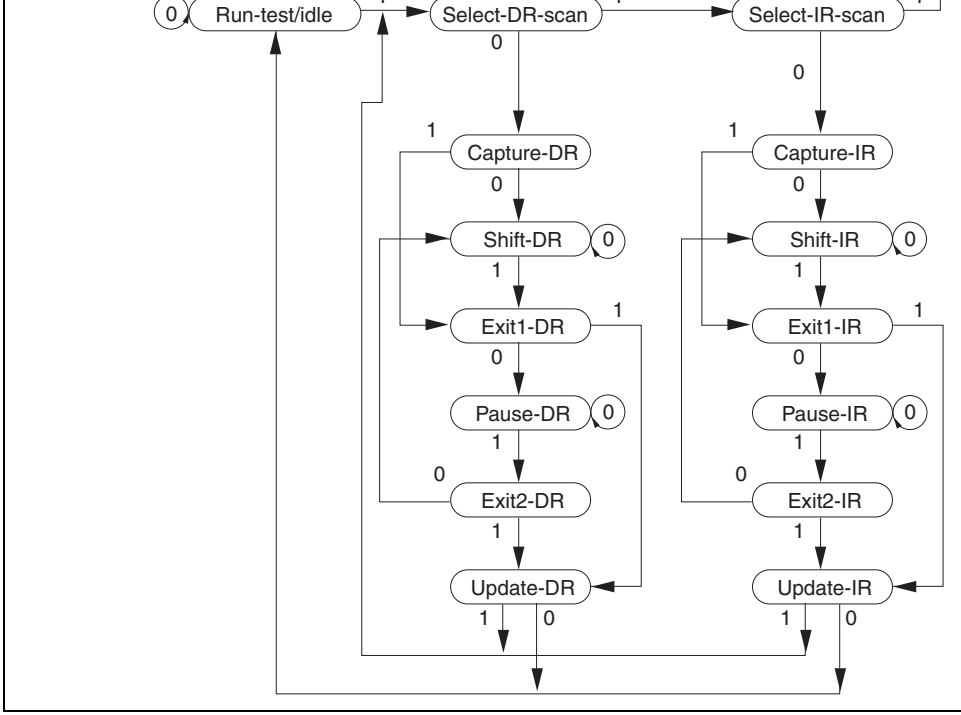


Figure 26.2 TAP Controller State Transitions

by setting a command in SDIR.

26.5.1 Supported Instructions

This LSI supports the three essential instructions defined in the IEEE1149.1 standard (BY
SAMPLE/PRELOAD, and EXTEST) and optional instructions (CLAMP, HIGHZ, and II

(1) BYPASS (Instruction code: B'1111)

The BYPASS instruction is an instruction that operates the bypass register. This instructi
shortens the shift path to speed up serial data transfer involving other chips on the printed
board. While this instruction is being executed, the test circuit has no effect on the syster
circuits.

(2) SAMPLE/PRELOAD (Instruction code: B'0100)

The SAMPLE/PRELOAD instruction inputs values from this LSI internal circuitry to the
boundary scan register, outputs values from the scan path, and loads data onto the scan pa
When this instruction is being executed, this LSI's input pin signals are transmitted direct
internal circuitry, and internal circuit values are directly output externally from the output
This LSI system circuits are not affected by execution of this instruction.

In a SAMPLE operation, a snapshot of a value to be transferred from an input pin to the i
circuitry, or a value to be transferred from the internal circuitry to an output pin, is latche
boundary scan register and read from the scan path. Snapshot latching does not affect nor
operation of this LSI.

printed circuit board, and input pins are used to latch test results into the boundary scan
from the printed circuit board. If testing is carried out by using the EXTEST instruction
the Nth test data is scanned in when test data (N-1) is scanned out.

Data loaded into the output pin boundary scan register in the Capture-DR state is not use
external circuit testing (it is replaced by a shift operation).

(4) CLAMP (Instruction code: B'0010)

When the CLAMP instruction is enabled, the output pin outputs the value of the bounda
register that has been previously set by the SAMPLE/PRELOAD instruction. While the
instruction is enabled, the state of the boundary scan register maintains the previous stat
regardless of the state of the TAP controller.

A bypass register is connected between the ETDI and ETDO pins. The related circuit op
the same way when the BYPASS instruction is enabled.

(5) HIGHZ (Instruction code: B'0011)

When the HIGHZ instruction is enabled, all output pins enter a high-impedance state. W
HIGHZ instruction is enabled, the state of the boundary scan register maintains the prev
regardless of the state of the TAP controller.

A bypass register is connected between the ETDI and ETDO pins. The related circuit op
the same way when the BYPASS instruction is enabled.

2. Boundary scan mode does not cover clock-related pins (EXTAL, XTAL, UEX
   UXTAL).
3. Boundary scan mode does not cover reset- and standby-related pins ($\overline{\text{RES}}$, $\overline{\text{ST}}$
   $\overline{\text{RESO}}$).
4. Boundary scan mode does not cover JTAG-related pins (ETCK, ETDI, ETDO
   and $\overline{\text{ETRST}}$).
5. Fix the $\overline{\text{MD2}}$ pin high.
6. Use the $\overline{\text{STBY}}$ pin in high state.
7. Boundary scan mode does not cover the PECI pin.
8. Boundary scan mode does not cover the USB pins (USB+, USB-).

— To prevent the LSI system operation from being affected by the ETRST pin of th
tester, circuits must be separated.

— Alternatively, to prevent the $\overline{\text{ETRST}}$ pin of the board tester from being affected b
system reset, circuits must be separated.

Figure 26.3 shows a design example of the reset signal circuit wherein no reset signa
interference occurs.

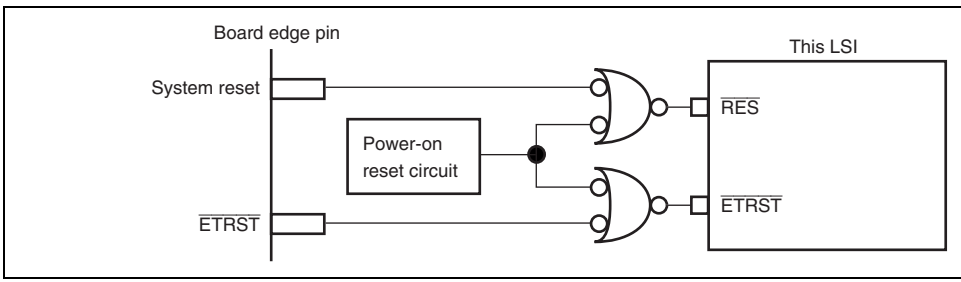


Figure 26.3 Reset Signal Circuit Without Reset Signal Interference

should then be retried, regardless of the transfer operation.

8. If a pin with a pull-up function is sampled while its pull-up function is enabled, 1 can detected at the corresponding input scan register. In this case, the corresponding enab register should be cleared to 0.
9. If a pin with an open-drain function is sampled while its open-drain function is enable corresponding output scan register is 1, 0 can be detected at the corresponding enable register.

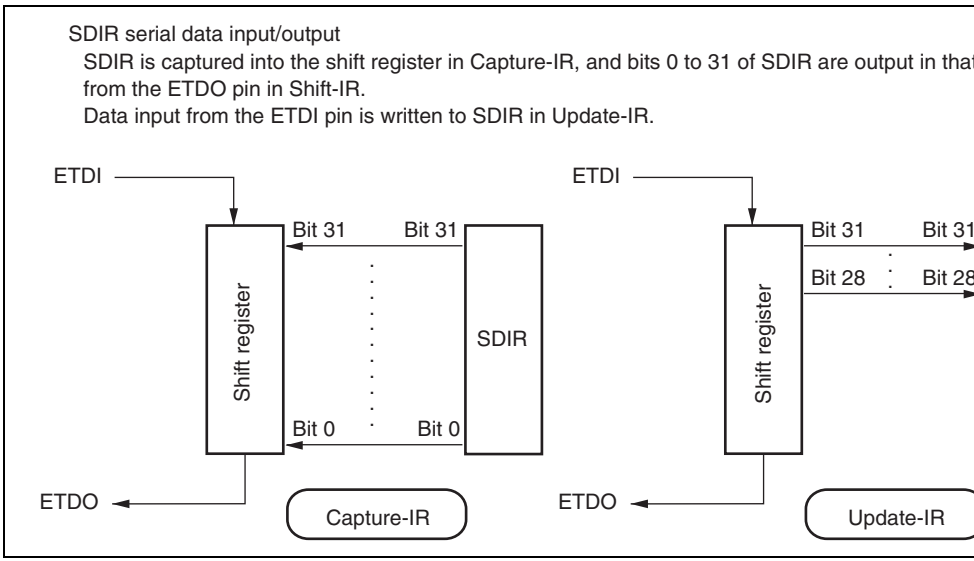


Figure 26.4 Serial Data Input/Output (1)

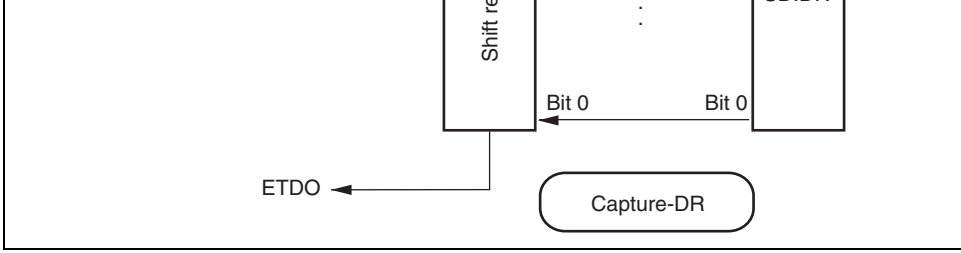


Figure 26.5 Serial Data Input/Output (2)

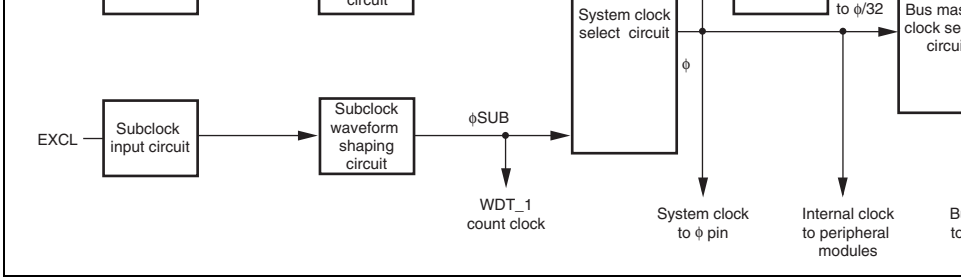


Figure 27.1 Block Diagram of Clock Pulse Generator

The bus master clock is selected as either high-speed mode or medium-speed mode by s
according to the settings of the SCK2 to SCK0 bits in the standby control register. Use o
medium-speed clock (ϕ/2 to ϕ/32) may be limited during CPU operation and when acces
internal memory of the CPU. The operation speed of the DTC and the external space acc
are thus stabilized regardless of the setting of medium-speed mode. For details on the sta
control register, see section 28.1.1, Standby Control Register (SBYCR).

The subclock input is controlled by software according to the EXCLE bit setting in the l
control register. For details on the low power control register, see section 28.1.2, Low-P
Control Register (LPWRCR).

Figure 27.3 shows the equivalent circuit of a crystal resonator. A crystal resonator having characteristics given in table 27.2 should be used.

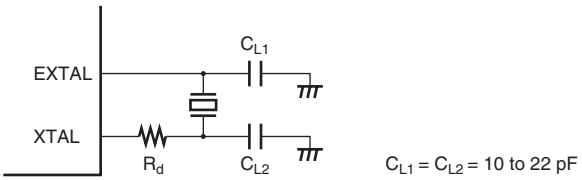


Figure 27.2 Typical Connection to Crystal Resonator

Table 27.1 Damping Resistance Values

Frequency (MHz)	5	8	8.5
R_d (Ω)	300	200	0

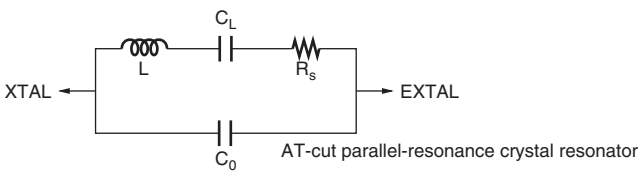


Figure 27.3 Equivalent Circuit of Crystal Resonator

To input an inverted clock to the XTAL pin, the external clock should be tied to high in
mode.

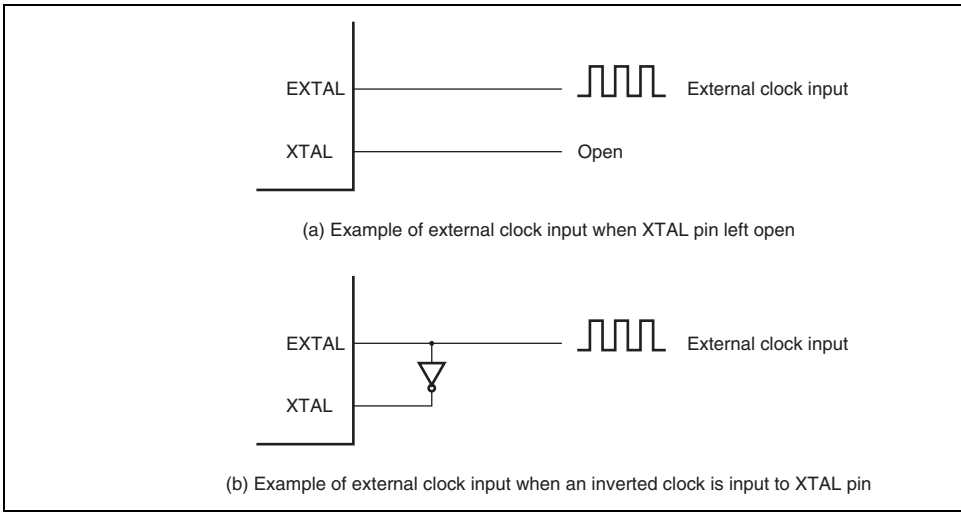


Figure 27.4 Example of External Clock Input

When a specified clock signal is input to the EXTAL pin, internal clock signal output is
determined after the external clock output stabilization delay time (t_{DEXT}) has passed. As
signal output is not determined during the t_{DEXT} cycle, a reset signal should be set to low
in reset state. For the external clock output stabilization delay time, refer to section 31, E
Characteristics.

27.3 Medium-Speed Clock Divider

The medium-speed clock divider divides the system clock (ϕ), and generates ϕ/2, ϕ/4, ϕ/8 and ϕ/32 clocks.

27.4 Bus Master Clock Select Circuit

The bus master clock select circuit selects a clock to supply the bus master with either the clock (ϕ) or medium-speed clock (ϕ/2, ϕ/4, ϕ/8, ϕ/16, or ϕ/32) by the SCK2 to SCK0 bits SBYCR.

27.5 Subclock Input Circuit

The subclock input circuit controls subclock input from the EXCL pin. To use the subclo 32.768-kHz external clock should be input from the EXCL pin. At this time, the P56DDR P5DDR should be cleared to 0, and the EXCLE bit in LPWRCR should be set to 1.

When the subclock is not used, subclock input should not be enabled.

27.6 Subclock Waveform Shaping Circuit

To remove noise from the subclock input at the EXCL pin, the subclock is sampled by a clock. The sampling frequency is set by the NESEL bit in LPWRCR.

27.8.2 Notes on Board Design

When using a crystal resonator, the crystal resonator and its load capacitors should be pla
close as possible to the EXTAL and XTAL pins. Other signal lines should be routed awa
the oscillation circuit to prevent inductive interference with the correct oscillation as sho
figure 27.5.

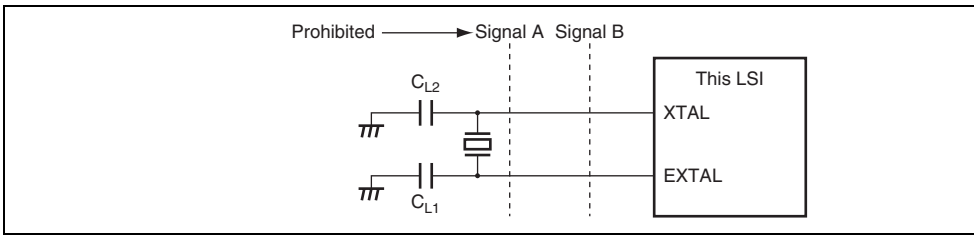


Figure 27.5 Note on Board Design of Oscillation Circuit Section

27.8.3 Note on Operation Check

This LSI may oscillate at several kHz of frequency even when a crystal resonator is not c
to the EXTAL and XTAL pins or an external clock is not input. Use this LSI after confir
the LSI operates with appropriate frequency.

The CPU stops but on-chip peripheral modules continue operating.

- Software standby mode
  Clock oscillation stops, and the CPU and on-chip peripheral modules stop operating.
- Hardware standby mode
  Clock oscillation stops, and the CPU and on-chip peripheral modules enter reset stat
- Module stop mode
  Independently of above operating modes, on-chip peripheral modules that are not us
  stopped individually.

- Module stop control register A (MSTPCRA)
- Sub-chip module stop control register BH, BL (SUBMSTPBH, SUBMSTPBL)

28.1.1 Standby Control Register (SBYCR)

SBYCR controls power-down modes.

Bit	Bit Name	Initial Value	R/W	Description
7	SSBY	0	R/W	Software Standby Specifies the operating mode to be entered after executing the SLEEP instruction. When the SLEEP instruction is executed in high-s mode or medium-speed mode: 0: Shifts to sleep mode 1: Shifts to software standby mode Note that the SSBY bit is not changed even if a m transition occurs by an interrupt.

				STS0 values and wait time.
3	DTSPEED	0	R/W	DTC Speed Specifies the operating clock for the bus masters other than the CPU in medium-speed mode. 0: All bus masters operate based on the mediu clock. 1: The DTC operates based on the system cloc The operating clock is changed when a DTC tra requested even if the CPU operates based on th medium-speed clock.
2	SCK2	0	R/W	System Clock Select 2 to 0
1	SCK1	0	R/W	Select a clock for the bus master in high-speed medium-speed mode.
0	SCK0	0	R/W	000: High-speed mode (Initial value) 001: Medium-speed clock: ϕ/2 010: Medium-speed clock: ϕ/4 011: Medium-speed clock: ϕ/8 100: Medium-speed clock: ϕ/16 101: Medium-speed clock: ϕ/32 11x: Must not be set.

[Legend]

x: Don't care

Recommended specification

Note: * Setting prohibited.

[Legend] x: Don't care

				from the EXCL pin is sampled using the clock (ϕ generated by the system clock pulse generator. 0: Sampling using ϕ/32 clock 1: Sampling using ϕ/4 clock
4	EXCLE	0	R/W	Subclock Input Enable Enables/disables subclock input from the EXCL 0: Disables subclock input from the EXCL pin 1: Enables subclock input from the EXCL pin
3	—	0	R/W	Reserved The initial value should not be changed.
2	PNCCS	0	R/W	Address Multiplex Chip Select Controls the output polarity of chip select signals $\overline{IOS}$) in the address multiplex extended mode. 0: Outputs $\overline{CS256}$, and $\overline{IOS}$ 1: Outputs CS256, and IOS
1	PNCAH	0	R/W	Address Multiplex Address Hold Controls the output polarity of the address hold s in the address multiplex extended mode. 0: Outputs $\overline{AH}$ 1: Outputs AH
0	—	0	R/W	Reserved The initial value should not be changed.

5	MSTP13	1	R/W	16-bit free-running timer (FRT)
4	MSTP12	1	R/W	8-bit timers (TMR_0, TMR_1)
3	MSTP11	1	R/W	14-bit PWM timer (PWMX)
2	MSTP10	1	R/W	Reserved The initial value should not be changed.
1	MSTP9	1	R/W	A/D converter
0	MSTP8	1	R/W	8-bit timers (TMR_X, TMR_Y)

- MSTPCRL

Bit	Bit Name	Initial Value	R/W	Corresponding Module
7	MSTP7	1	R/W	Serial communication interface 3 (SCI_3)
6	MSTP6	1	R/W	Serial communication interface 1 (SCI_1)
5	MSTP5	1	R/W	Reserved The initial value should not be changed.
4	MSTP4	1	R/W	I^2C bus interface channel 0 (IIC_0)
3	MSTP3	1	R/W	I^2C bus interface channel 1 (IIC_1)
2	MSTP2	1	R/W	I^2C bus interface channel 2, 3 (IIC_2, IIC_3)
1	MSTP1	1	R/W	CRC operation circuit
0	MSTP0	1	R/W	I^2C bus interface channel 4, 5 (IIC_4, IIC_5)

MSTPCR sets operation and stop by the combination of bits as follows:

MSTPCRH (bit 3) MSTP11	MSTPCRA (bit 2) MSTPA2	Function
0	0	14-bit PWM timer (PWMX_1) operates.
0	1	14-bit PWM timer (PWMX_1) stops.
1	x	Reserved

MSTPCRH (bit 3) MSTP11	MSTPCRA (bit 1) MSTPA1	Function
0	0	14-bit PWM timer (PWMX_0) operates.
0	1	14-bit PWM timer (PWMX_0) stops.
1	x	Reserved

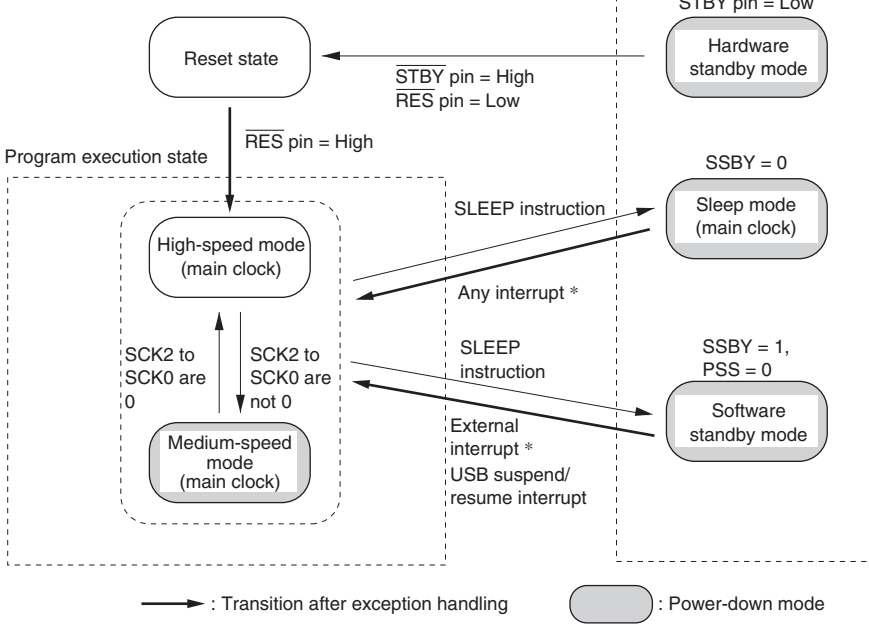
Note: Bit 3 of MSTPCRH is the module stop bit for PWMX_0 and PWMX_1.

[Legend] x: Don't care

6	SMSTPB14	1	R/W	Ethernet controller (EtherC)
5	SMSTPB13	1	R/W	DMAC for Ethernet (E-DMAC)
4	SMSTPB12	1	R/W	USB function module (USB) This bit is valid only in the H8S/2472 Group. The initial value should not be changed in the H Group.
3 to 0	SMSTPB11 to SMSTPB8	All 1	R/W	Reserved The initial values should not be changed.

- SUBMSTPBL

Bit	Bit Name	Initial Value	R/W	Corresponding Module
7 to 5	SMSTPB7 to SMSTPB5	All 1	R/W	Reserved The initial values should not be changed.
4	SMSTPB4	1	R/W	PECI This bit is not incorporated in the H8S/2463 Gr initial values should not be changed.
3	SMSTPB3	1	R/W	Serial communication interface with FIFO (SCI
2	SMSTPB2	1	R/W	Synchronous serial communication unit (SSU)
1	SMSTPB1	1	R/W	LPC interface (LPC)
0	SMSTPB0	1	R/W	Reserved The initial values should not be changed.



Note: * When a transition is made between modes by means of an interrupt, the transition cannot be m
on interrupt source generation alone. Ensure that interrupt handling is performed after acceptin
interrupt request.

Figure 28.1 Mode Transition Diagram

modules			medium-speed mode/ Functioning		/Halted (retained)	(retained)	
	WDT_1		Functioning		Functioning		
	WDT_0						
	TMR_0,TMR_1				Functioning /Halted (retained)		
	LPC						
	FRT						
	TMR_X, TMR_Y						
	IIC_0 to IIC_5						
	CRC						
	SCI_1, SCI_3	Functioning	Functioning	Functioning	Functioning /Halted (retained/ reset)	Halted (retained/ reset)	Hal
	SCIF, SSU, PECI						
	EtherC, E-DMAC, USB		Halted				
	PWMX_0,PWMX_1				Functioning/ Halted (reset)	Halted (reset)	
	A/D converter						
	RAM			Functioning (DTC)	Functioning	Retained	Ret
	I/O			Functioning			Hig

Notes: Halted (retained) means that internal register values are retained. The internal stat operation suspended.

Halted (reset) means that internal register values and internal states are initialized.

In module stop mode, only modules for which a stop setting has been made are ha (reset or retained).

In medium-speed mode, a bus access is executed in the specified number of states with r
the bus master operating clock. For example, if ϕ/4 is selected as the operating clock, on
memory is accessed in 4 states, and internal I/O registers in 8 states.

By clearing all of bits SCK2 to SCK0 to 0, a transition is made to high-speed mode at th
the current bus cycle.

If a SLEEP instruction is executed when the SSBY bit in SBYCR is cleared to 0, a trans
made to sleep mode. When sleep mode is cleared by an interrupt, medium-speed mode i
When the SLEEP instruction is executed with the SSBY bit set to 1 and the PSS bit in T
(WDT_1) cleared to 0, operation shifts to software standby mode. When software standb
cleared by an external interrupt, medium-speed mode is restored.

When the $\overline{\text{RES}}$ pin is set low, medium-speed mode is cancelled and 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 28.2 shows an example of medium-speed mode timing.

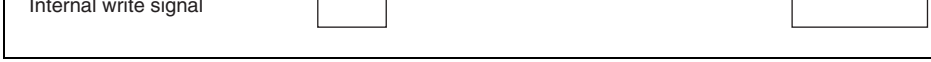


Figure 28.2 Medium-Speed Mode Timing

28.4 Sleep Mode

The CPU makes a transition to sleep mode if the SLEEP instruction is executed when the bit in SBYCR is cleared to 0. In sleep mode, CPU operation stops but the peripheral mod not stop. The contents of the CPU's internal registers are retained.

Sleep mode is exited by any interrupt, the $\overline{\text{RES}}$ pin, or the $\overline{\text{STBY}}$ pin.

When an interrupt occurs, sleep mode is exited and interrupt exception handling starts. Sl mode is not exited if the interrupt is disabled, or interrupts other than NMI are masked by CPU.

Setting the $\overline{\text{RES}}$ pin level low cancels sleep mode and selects the reset state. After the osc settling time has passed, driving the $\overline{\text{RES}}$ pin high causes the CPU to start reset exception handling.

When the $\overline{\text{STBY}}$ pin level is driven low, a transition is made to hardware standby mode.

suspend/resume interrupt (RESUME), the $\overline{\text{RES}}$ pin input, or $\overline{\text{STBY}}$ pin input.

When an external interrupt request signal is input, system clock oscillation starts, and af
elapse of the time set in bits STS2 to STS0 in SBYCR, software standby mode is cleared
interrupt exception handling is started. When exiting software standby mode by IRQ0 to
interrupt, set the corresponding enable bit to 1 and ensure that any interrupt with a highe
than IRQ0 to IRQ15 is not generated. Software standby mode is not exited if the corresp
enable bit is cleared to 0 or if the interrupt has been masked by the CPU.

When the $\overline{\text{RES}}$ pin is driven low, system clock oscillation is started. At the same time as
clock oscillation starts, the system clock is supplied to the entire LSI. Note that the $\overline{\text{RES}}$
be held low until clock oscillation settles. When the $\overline{\text{RES}}$ pin goes high after clock oscill
settles, the CPU begins reset exception handling.

When the $\overline{\text{STBY}}$ pin is driven low, software standby mode is cancelled and a transition i
hardware standby mode.

Figure 28.3 shows an example in which a transition is made to software standby mode a
falling edge of the NMI pin, and software standby mode is cleared at the rising edge of t
pin.

In this example, an NMI interrupt is accepted with the NMIEG bit in SYSCR cleared to
edge specification), then the NMIEG bit is set to 1 (rising edge specification), the SSBY
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 of the NMI pin.

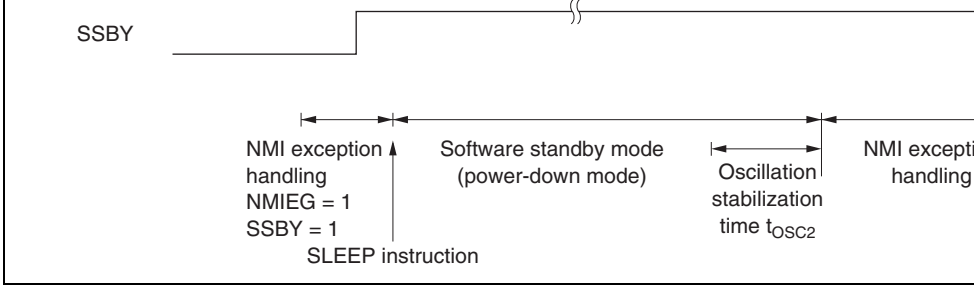


Figure 28.3 Software Standby Mode Application Example

Hardware standby mode is cleared by the $\overline{\text{STBY}}$ pin input or the $\overline{\text{RES}}$ pin input.

When the $\overline{\text{STBY}}$ pin is driven high while the $\overline{\text{RES}}$ pin is low, clock oscillation is started. that the $\overline{\text{RES}}$ pin is held low until system clock oscillation settles. When the $\overline{\text{RES}}$ pin is subsequently driven high after the clock oscillation settling time has passed, reset excep handling starts.

Figure 28.4 shows an example of hardware standby mode timing.

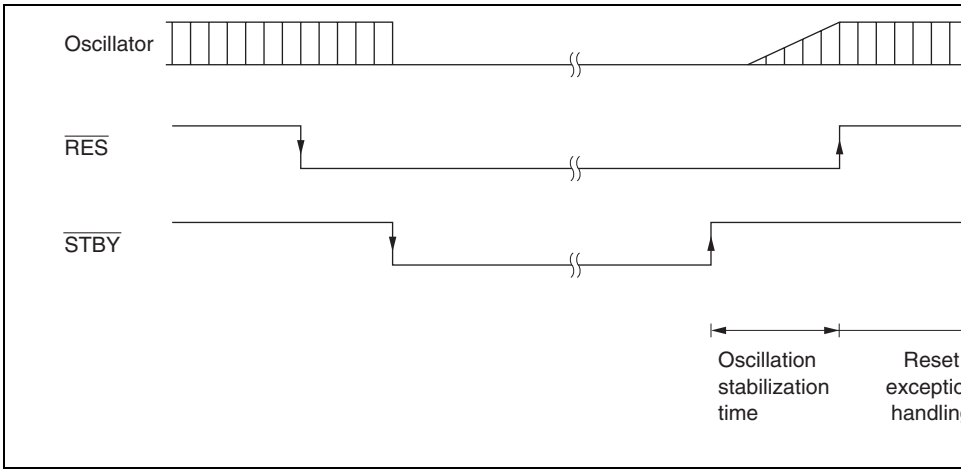


Figure 28.4 Hardware Standby Mode Timing

While an on-chip peripheral module is in module stop mode, read/write access to its regis
disabled.

28.8 Usage Notes

28.8.1 I/O Port Status

The status of the I/O ports is retained in software standby mode. Therefore, when a high
output, the current consumption is not reduced by the amount of current to support the hi
output.

28.8.2 Current Consumption when Waiting for Oscillation Settling

The current consumption increases during oscillation settling.

28.8.3 DTC Module Stop Mode

If the DTC module stop mode specification and DTC bus request occur simultaneously, t
released to the DTC and the MSTP bit cannot be set to 1. After completing the DTC bus
the MSTP bit to 1 again.

28.8.4 Notes on Subclock Usage

When using the subclock, make a transition to power-down mode after setting the EXCL
LPWRCR to 1 and loading the subclock two or more cycles. When not using the subcloc
EXCLE bit should not be set to 1.

- Registers are classified by functional modules.
- The access size is indicated.

2. Register Bits

- Bit configurations of the registers are described in the same order as the Register Ad
(address order) above.
- Reserved bits are indicated by — in the bit name column.
- The bit number in the bit-name column indicates that the whole register is allocated
counter or for holding data.
- 16-bit registers are indicated from the bit on the MSB side, in 2 lines of eight bits.
- 32-bit registers are indicated from the bit on the MSB side, in 4 lines of eight bits.

3. Register States in Each Operating Mode

- Register states are described in the same order as the Register Addresses (address or
above.
- The register states described here are for the basic operating modes. If there is a spec
for an on-chip peripheral module, refer to the section on that on-chip peripheral mod

EtherC mode register	ECMR	32	H'F900	EtherC	16
EtherC status register	ECSR	32	H'F904	EtherC	16
EtherC interrupt permission register	ECSIPR	32	H'F908	EtherC	16
PHY interface register	PIR	32	H'F90C	EtherC	16
MAC address high register	MAHR	32	H'F910	EtherC	16
MAC address low register	MALR	32	H'F914	EtherC	16
Receive frame length register	RFLR	32	H'F918	EtherC	16
PHY status register	PSR	32	H'F91C	EtherC	16
Transmit retry over counter register	TROCR	32	H'F920	EtherC	16
Delayed collision detect counter register	CDCR	32	H'F924	EtherC	16
Lost carrier counter register	LCCR	32	H'F928	EtherC	16
Carrier not detect counter register	CNDCR	32	H'F92C	EtherC	16
CRC error frame counter register	CEFCR	32	H'F934	EtherC	16
Frame receive error counter register	FRECR	32	H'F938	EtherC	16
Too-short frame receive counter register	TSFRCR	32	H'F93C	EtherC	16
Too-long frame receive counter register	TLFRCR	32	H'F940	EtherC	16
Residual-bit frame counter register	RFCR	32	H'F944	EtherC	16
Multicast address frame counter register	MAFCR	32	H'F948	EtherC	16
IPG register	IPGR	32	H'F954	EtherC	16
Automatic PAUSE frame set register	APR	32	H'F958	EtherC	16
Manual PAUSE frame set register	MPR	32	H'F95C	EtherC	16

Receive descriptor list address register	RDLAR	32	H'F990	E-DMAC	16
EtherC/E-DMAC status register	EESR	32	H'F994	E-DMAC	16
EtherC/E-DMAC status interrupt permission register	EESIPR	32	H'F998	E-DMAC	16
Transmit/receive status copy enable register	TRSCER	32	H'F99C	E-DMAC	16
Receive missed-frame counter register	RMFCR	32	H'F9A0	E-DMAC	16
Transmit FIFO threshold register	TFTR	32	H'F9A4	E-DMAC	16
FIFO depth register	FDR	32	H'F9A8	E-DMAC	16
Receiving method control register	RMCR	32	H'F9AC	E-DMAC	16
Flow control FIFO threshold register	FCFTR	32	H'F9B4	E-DMAC	16
Transmit Interrupt Register	TRIMD	32	H'F9BC	E-DMAC	16
Receiving-Buffer Write Address Register	RBWAR	32	H'F9C0	E-DMAC	16
Receiving-Descriptor Fetch Address Register	RDFAR	32	H'F9C4	E-DMAC	16
Transmission-Buffer Read Address Register	TBRAR	32	H'F9CC	E-DMAC	16
Transmission-Descriptor Fetch Address Register	TDFAR	32	H'F9D0	E-DMAC	16
Bit rate setting register	ECBRR	8	H'F9D4	E-DMAC	16
Interrupt flag register 0	IFR0	8	H'FA00	USB	16

Interrupt select register 2	ISR2	8	H'FA0A	USB	16
EP0i data register	EPDR0i	8	H'FA0C	USB	16
EP0o data register	EPDR0o	8	H'FA0D	USB	16
EP0s data register	EPDR0s	8	H'FA0E	USB	16
EP1 data register	EPDR1	8	H'FA10	USB	16
EP2 data register	EPDR2	8	H'FA14	USB	16
EP3 data register	EPDR3	8	H'FA18	USB	16
EP0o receive data size register	EPSZ0o	8	H'FA24	USB	16
EP1 receive data size register	EPSZ1	8	H'FA25	USB	16
Data status register	DASTS	8	H'FA27	USB	16
FIFO clear register	FCLR	8	H'FA28	USB	16
Endpoint stall register	EPSTL	8	H'FA2A	USB	16
Trigger register	TRG	8	H'FA2C	USB	16
DTC transfer setting register	DMA	8	H'FA2D	USB	16
Configuration value register	CVR	8	H'FA2E	USB	16
Control register	CTLR	8	H'FA2F	USB	16
Endpoint information register	EPIR	8	H'FA32	USB	16
Transceiver test register 0	TRNTREG0	8	H'FA44	USB	16
Transceiver test register 1	TRNTREG1	8	H'FA45	USB	16
Receive buffer register	FRBR	8	H'FC80	SCIF	16
Transmitter holding register	FTHR	8	H'FC80	SCIF	16

Line status register	FLSR	8	H'FC85	SCIF	16
Modem status register	FMSR	8	H'FC86	SCIF	16
Scratch pad register	FSCR	8	H'FC87	SCIF	16
SCIF control register	SCIFCR	8	H'FC88	SCIF	16
SS control register H	SSCRH	8	H'FCC0	SSU	16
SS control register L	SSCRL	8	H'FCC1	SSU	16
SS mode register	SSMR	8	H'FCC2	SSU	16
SS enable register	SSER	8	H'FCC3	SSU	16
SS status register	SSSR	8	H'FCC4	SSU	16
SS control register 2	SSCR2	8	H'FCC5	SSU	16
SS transmit data register 0	SSTDR0	8	H'FCC6	SSU	16
SS transmit data register 1	SSTDR1	8	H'FCC7	SSU	16
SS transmit data register 2	SSTDR2	8	H'FCC8	SSU	16
SS transmit data register 3	SSTDR3	8	H'FCC9	SSU	16
SS receive data register 0	SSRDR0	8	H'FCCA	SSU	16
SS receive data register 1	SSRDR1	8	H'FCCB	SSU	16
SS receive data register 2	SSRDR2	8	H'FCCC	SSU	16
SS receive data register 3	SSRDR3	8	H'FCCD	SSU	16
SS shift register	SSTRSR	8	H'FCCE	SSU	16
Host interface control register 4	HICR4	8	H'FD00	LPC	16
BT status register 0	BTSR0	8	H'FD02	LPC	16
BT status register 1	BTSR1	8	H'FD03	LPC	16

SMIC data register	SMICDTR	8	H'FD0B	LPC	16
SMIC interrupt register 0	SMICIR0	8	H'FD0C	LPC	16
SMIC interrupt register 1	SMICIR1	8	H'FD0E	LPC	16
SERIRQ control register3	SIRQCR3	8	H'FD0F	LPC	16
Bidirectional data register 0MW	TWR0MW	8	H'FD10	LPC	16
Bidirectional data register 0SW	TWR0SW	8	H'FD10	LPC	16
Bidirectional data register 1	TWR1	8	H'FD11	LPC	16
Bidirectional data register 2	TWR2	8	H'FD12	LPC	16
Bidirectional data register 3	TWR3	8	H'FD13	LPC	16
Bidirectional data register 4	TWR4	8	H'FD14	LPC	16
Bidirectional data register 5	TWR5	8	H'FD15	LPC	16
Bidirectional data register 6	TWR6	8	H'FD16	LPC	16
Bidirectional data register 7	TWR7	8	H'FD17	LPC	16
Bidirectional data register 8	TWR8	8	H'FD18	LPC	16
Bidirectional data register 9	TWR9	8	H'FD19	LPC	16
Bidirectional data register 10	TWR10	8	H'FD1A	LPC	16
Bidirectional data register 11	TWR11	8	H'FD1B	LPC	16
Bidirectional data register 12	TWR12	8	H'FD1C	LPC	16
Bidirectional data register 13	TWR13	8	H'FD1D	LPC	16
Bidirectional data register 14	TWR14	8	H'FD1E	LPC	16
Bidirectional data register 15	TWR15	8	H'FD1F	LPC	16
Input data register 3	IDR3	8	H'FD20	LPC	16

Input data register 1	IDR1	8	H'FD28	LPC	16
Output data register 1	ODR1	8	H'FD29	LPC	16
Status register 1	STR1	8	H'FD2A	LPC	16
SERIRQ control register 5	SIRQCR5	8	H'FD2B	LPC	16
Input data register 2	IDR2	8	H'FD2C	LPC	16
Output data register 2	ODR2	8	H'FD2D	LPC	16
Status register 2	STR2	8	H'FD2E	LPC	16
Host interface select register	HISEL	8	H'FD2F	LPC	16
Host interface control register 0	HICR0	8	H'FD30	LPC	16
Host interface control register 1	HICR1	8	H'FD31	LPC	16
Host interface control register 2	HICR2	8	H'FD32	LPC	16
Host interface control register 3	HICR3	8	H'FD33	LPC	16
SERIRQ control register2	SIRQCR2	8	H'FD34	LPC	16
BT data buffer	BTDTR	8	H'FD35	LPC	16
BT FIFO valid size register 0	BTFVSR0	8	H'FD36	LPC	16
BT FIFO valid size register 1	BTFVSR1	8	H'FD37	LPC	16
LPC channel 1, 2 address register H	LADR12H	8	H'FD38	LPC	16
LPC channel 1, 2 address register L	LADR12L	8	H'FD39	LPC	16
SCIF address register H	SCIFADRH	8	H'FD3A	LPC	16
SCIF address register L	SCIFADRL	8	H'FD3B	LPC	16
Sub-chip module stop control register BH	SUBMSTPBH	8	H'FE3E	SYSTEM	8
Sub-chip module stop control register BL	SUBMSTPBL	8	H'FE3F	SYSTEM	8

Port F output data register	PFODR	8	H'FE49	PORT	8
Port E input data register	PEPIN	8	H'FE4A	PORT	8
Port E data direction register	PEDDR	8	H'FE4A	PORT	8
Port F input data register	PFPIN	8	H'FE4B	PORT	8
Port F data direction register	PFDDR	8	H'FE4B	PORT	8
Port C output data register	PCODR	8	H'FE4C	PORT	8
Port D output data register	PDODR	8	H'FE4D	PORT	8
Port C input data register	PCPIN	8	H'FE4E	PORT	8
Port C data direction register	PCDDR	8	H'FE4E	PORT	8
Port D input data register	PDPIN	8	H'FE4F	PORT	8
Port D data direction register	PDDDR	8	H'FE4F	PORT	8
Flash code control/status register	FCCS	8	H'FE88	FLASH	8
Flash program code select register	FPCS	8	H'FE89	FLASH	8
Flash erase code select register	FECS	8	H'FE8A	FLASH	8
Flash key code register	FKEY	8	H'FE8C	FLASH	8
Flash MAT select register	FMATS	8	H'FE8D	FLASH	8
Flash transfer destination address register	FTDAR	8	H'FE8E	FLASH	8
I^2C bus control register_4	ICCR_4	8	H'FE90	IIC_4	8
I^2C bus status register_4	ICSR_4	8	H'FE91	IIC_4	8
I^2C bus data register_4	ICDR_4	8	H'FE92	IIC_4	8
Second slave address register_4	SARX_4	8	H'FE92	IIC_4	8

Slave address register_5	SAR_5	8	H'FE97	IIC_5	8
Serial mode register_1	SMR_1	8	H'FE98	SCI_1	8
Bit rate register_1	BRR_1	8	H'FE99	SCI_1	8
Serial control register_1	SCR_1	8	H'FE9A	SCI_1	8
Transmit data register_1	TDR_1	8	H'FE9B	SCI_1	8
Serial status register_1	SSR_1	8	H'FE9C	SCI_1	8
Receive data register_1	RDR_1	8	H'FE9D	SCI_1	8
Smart card mode register_1	SCMR_1	8	H'FE9E	SCI_1	8
A/D data register A	ADDRA	16	H'FEA0	ADC	16
A/D data register B	ADDRB	16	H'FEA2	ADC	16
A/D data register C	ADDRC	16	H'FEA4	ADC	16
A/D data register D	ADDRD	16	H'FEA6	ADC	16
A/D data register E	ADDRE	16	H'FEA8	ADC	16
A/D data register F	ADDRF	16	H'FEAA	ADC	16
A/D data register G	ADDRG	16	H'FEAC	ADC	16
A/D data register H	ADDRH	16	H'FEAE	ADC	16
A/D control/status register	ADCSR	8	H'FEB0	ADC	16
A/D control register	ADCR	8	H'FEB1	ADC	16
Serial multiplexed mode register 0	SMR0	8	H'FEB8	SMX	8
Serial multiplexed mode register 1	SMR1	8	H'FEB9	SMX	8
Noise canceler enable register	P4BNCE	8	H'FEBA	PORT	8
Noise canceler mode control register	P4BNCMC	8	H'FEBB	PORT	8

I²C bus mode register_3	ICMR_3	8	H'FEC3	IIC_3	8
Slave address register_3	SAR_3	8	H'FEC3	IIC_3	8
I²C bus control register_2	ICCR_2	8	H'FEC8	IIC_2	8
I²C bus status register_2	ICSR_2	8	H'FEC9	IIC_2	8
I²C bus data register_2	ICDR_2	8	H'FECA	IIC_2	8
Second slave address register_2	SARX_2	8	H'FECA	IIC_2	8
I²C bus mode register_2	ICMR_2	8	H'FECB	IIC_2	8
Slave address register_2	SAR_2	8	H'FECB	IIC_2	8
PWMX (D/A) data register A_1	DADRA_1	16	H'FECC	PWMX_1	8
PWMX (D/A) control register_1	DACR_1	8	H'FECC	PWMX_1	8
PWMX (D/A) data register B_1	DADRB_1	16	H'FECE	PWMX_1	8
PWMX (D/A) counter_1	DACNT_1	16	H'FECE	PWMX_1	8
CRC control register	CRCCR	8	H'FED4	CRC	16
CRC data input register	CRCDIR	8	H'FED5	CRC	16
CRC data output register	CRCDOR	16	H'FED6	CRC	16
I²C bus extended control register_0	ICXR_0	8	H'FED8	IIC_0	8
I²C bus extended control register_1	ICXR_1	8	H'FED9	IIC_1	8
I²C SMBus control register	ICSMBCR	8	H'FEDB	IIC	8
I²C bus extended control register_2	ICXR_2	8	H'FEDC	IIC_2	8
I²C bus extended control register_3	ICXR_3	8	H'FEDD	IIC_3	8
I²C bus transfer select register	IICX3	8	H'FEDF	IIC	8
I²C bus extended control register_4	ICXR_4	8	H'FEE0	IIC_4	8
I²C bus extended control register_5	ICXR_5	8	H'FEE1	IIC_5	8

IRQ sense control register H	ISCRH	8	H'FEEC	INT	8
IRQ sense control register L	ISCRL	8	H'FEED	INT	8
DTC enable register A	DTCERA	8	H'FEEE	DTC	8
DTC enable register B	DTCERB	8	H'FEEF	DTC	8
DTC enable register C	DTCERC	8	H'FEF0	DTC	8
DTC enable register D	DTCERD	8	H'FEF1	DTC	8
DTC enable register E	DTCERE	8	H'FEF2	DTC	8
DTC vector register	DTVECR	8	H'FEF3	DTC	8
Address break control register	ABRKCR	8	H'FEF4	INT	8
Break address register A	BARA	8	H'FEF5	INT	8
Break address register B	BARB	8	H'FEF6	INT	8
Break address register C	BARC	8	H'FEF7	INT	8
IRQ enable register 16	IER16	8	H'FEF8	INT	8
IRQ status register 16	ISR16	8	H'FEF9	INT	8
IRQ sense control register 16H	ISCR16H	8	H'FEFA	INT	8
IRQ sense control register 16L	ISCR16L	8	H'FEFB	INT	8
IRQ sense port select register 16	ISSR16	8	H'FEFC	PORT	8
IRQ sense port select register	ISSR	8	H'FEFD	PORT	8
Port control register 0	PTCNT0	8	H'FEFE	PORT	8
Bus control register 2	BCR2	8	H'FF80	BSC	8
Wait state control register 2	WSCR2	8	H'FF81	BSC	8
Peripheral clock select register	PCSR	8	H'FF82	PWMX	8

I^2C bus data register _1	ICDR_1	8	H'FF8E	IIC_1	8
Second slave address register _1	SARX_1	8	H'FF8E	IIC_1	8
I^2C bus mode register_1	ICMR_1	8	H'FF8F	IIC_1	8
Slave address register _1	SAR_1	8	H'FF8F	IIC_1	8
Timer interrupt enable register	TIER	8	H'FF90	FRT	8
Timer control/status register	TCSR	8	H'FF91	FRT	8
Free-running counter	FRC	16	H'FF92	FRT	16
Output compare register A	OCRA	16	H'FF94	FRT	16
Output compare register B	OCRB	16	H'FF94	FRT	16
Timer control register	TCR	8	H'FF96	FRT	16
Timer output compare control register	TOCR	8	H'FF97	FRT	16
Output compare register AR	OCRAR	16	H'FF98	FRT	16
Output compare register AF	OCRAF	16	H'FF9A	FRT	16
PWMX (D/A) data register A_0	DADRA_0	16	H'FFA0	PWMX_0	8
PWMX (D/A) control register_0	DACR_0	8	H'FFA0	PWMX_0	8
PWMX (D/A) data register B_0	DADRB_0	16	H'FFA6	PWMX_0	8
PWMX (D/A) counter_0	DACNT_0	16	H'FFA6	PWMX_0	8
Timer control/status register _0 (read)	TCSR_0	8	H'FFA8	WDT_0	16
Timer control/status register _0 (write)	TCSR_0	16	H'FFA8	WDT_0	16
Timer counter_0 (read)	TCNT_0	8	H'FFA9	WDT_0	16
Timer counter_0 (write)	TCNT_0	16	H'FFA8	WDT_0	16

Port 2 data direction register	P2DDR	8	H'FFB1	PORT	8
Port 1 data register	P1DR	8	H'FFB2	PORT	8
Port 2 data register	P2DR	8	H'FFB3	PORT	8
Port 3 data direction register	P3DDR	8	H'FFB4	PORT	8
Port 4 data direction register	P4DDR	8	H'FFB5	PORT	8
Port 3 data register	P3DR	8	H'FFB6	PORT	8
Port 4 data register	P4DR	8	H'FFB7	PORT	8
Port 5 data direction register	P5DDR	8	H'FFB8	PORT	8
Port 6 data direction register	P6DDR	8	H'FFB9	PORT	8
Port 5 data register	P5DR	8	H'FFBA	PORT	8
Port 6 data register	P6DR	8	H'FFBB	PORT	8
Port B output data register	PBODR	8	H'FFBC	PORT	8
Port B input data register	PBPIN	8	H'FFBD	PORT	8
Port 8 data direction register	P8DDR	8	H'FFBD	PORT	8
Port 7 input data register	P7PIN	8	H'FFBE	PORT	8
Port B data direction register	PBDDR	8	H'FFBE	PORT	8
Port 8 data register	P8DR	8	H'FFBF	PORT	8
Port 9 data direction register	P9DDR	8	H'FFC0	PORT	8
Port 9 data register	P9DR	8	H'FFC1	PORT	8
Interrupt enable register	IER	8	H'FFC2	INT	8
Serial timer control register	STCR	8	H'FFC3	SYSTEM	8
System control register	SYSCR	8	H'FFC4	SYSTEM	8
Mode control register	MDCR	8	H'FFC5	SYSTEM	8

Time constant register A_1	TCORA_1	8	H'FFCD	TMR_1	8
Time constant register B_0	TCORB_0	8	H'FFCE	TMR_0	8
Time constant register B_1	TCORB_1	8	H'FFCF	TMR_1	8
Timer counter_0	TCNT_0	8	H'FFD0	TMR_0	8
Timer counter_1	TCNT_1	8	H'FFD1	TMR_1	8
I^2C bus control register_0	ICCR_0	8	H'FFD8	IIC_0	8
I^2C bus status register_0	ICSR_0	8	H'FFD9	IIC_0	8
I^2C bus data register_0	ICDR_0	8	H'FFDE	IIC_0	8
Second slave address register_0	SARX_0	8	H'FFDE	IIC_0	8
I^2C bus mode register_0	ICMR_0	8	H'FFDF	IIC_0	8
Slave address register_0	SAR_0	8	H'FFDF	IIC_0	8
Serial mode register_3	SMR_3	8	H'FFE0	SCI_3	8
Bit rate register_3	BRR_3	8	H'FFE1	SCI_3	8
Serial control register_3	SCR_3	8	H'FFE2	SCI_3	8
Transmit data register_3	TDR_3	8	H'FFE3	SCI_3	8
Serial status register_3	SSR_3	8	H'FFE4	SCI_3	8
Receive data register_3	RDR_3	8	H'FFE5	SCI_3	8
Smart card mode register_3	SCMR_3	8	H'FFE6	SCI_3	8
Timer control/ status register_1 (read)	TCSR_1	8	H'FFEA	WDT_1	16
Timer control/ status register_1 (write)	TCSR_1	16	H'FFEA	WDT_1	16
Timer counter_1 (read)	TCNT_1	8	H'FFEB	WDT_1	16
Timer counter_1 (write)	TCNT_1	16	H'FFEA	WDT_1	16

Time constant register A_Y	TCORA_Y	8	H'FFF2	TMR_Y	8
Time constant register B_Y	TCORB_Y	8	H'FFF3	TMR_Y	8
Timer counter_Y	TCNT_Y	8	H'FFF4	TMR_Y	8
Timer connection register S	TCONRS	8	H'FFFE	TMR	8

Notes: 1. The registers related to USB are supported only by the H8S/2472 Group.
2. The registers related to PECI are supported only by the H8S/2472 Group and H8S/2462 Group.

	—	—	—	PRCEF	—	—	MPDE	—
	—	RE	TE	—	ILB	ELB	DM	PRM
ECSR	—	—	—	—	—	—	—	—
	—	—	—	—	—	—	—	—
	—	—	—	—	—	—	—	—
	—	—	—	PSRTO	—	LCHNG	MPD	ICD
ECSIPR	—	—	—	—	—	—	—	—
	—	—	—	—	—	—	—	—
	—	—	—	—	—	—	—	—
	—	—	—	PSRTOIP	—	LCHNGIP	MPDIP	ICDIP
PIR	—	—	—	—	—	—	—	—
	—	—	—	—	—	—	—	—
	—	—	—	—	—	—	—	—
	—	—	—	—	MDI	MDO	MMD	MDC
MAHR	MA47	MA46	MA45	MA44	MA43	MA42	MA41	MA40
	MA39	MA38	MA37	MA36	MA35	MA34	MA33	MA32
	MA31	MA30	MA29	MA28	MA27	MA26	MA25	MA24
	MA23	MA22	MA21	MA20	MA19	MA18	MA17	MA16
MALR	—	—	—	—	—	—	—	—
	—	—	—	—	—	—	—	—
	MA15	MA14	MA13	MA12	MA11	MA10	MA9	MA8
	MA7	MA6	MA5	MA4	MA3	MA2	MA1	MA0

	—	—	—	—	—	—	—	LMON
TROCR	TROC31	TROC30	TROC29	TROC28	TROC27	TROC26	TROC25	TROC2
	TROC23	TROC22	TROC21	TROC20	TROC19	TROC18	TROC17	TROC1
	TROC15	TROC14	TROC13	TROC12	TROC11	TROC10	TROC9	TROC8
	TROC7	TROC6	TROC5	TROC4	TROC3	TROC2	TROC1	TROC0
CDCR	COSDC31	COSDC30	COSDC29	COSDC28	COSDC27	COSDC26	COSDC25	COSDC
	COSDC23	COSDC22	COSDC21	COSDC20	COSDC19	COSDC18	COSDC17	COSDC
	COSDC15	COSDC14	COSDC13	COSDC12	COSDC11	COSDC10	COSDC9	COSDC
	COSDC7	COSDC6	COSDC5	COSDC4	COSDC3	COSDC2	COSDC1	COSDC
LCCR	LCC31	LCC30	LCC29	LCC28	LCC27	LCC26	LCC25	LCC24
	LCC23	LCC22	LCC21	LCC20	LCC19	LCC18	LCC17	LCC16
	LCC15	LCC14	LCC13	LCC12	LCC11	LCC10	LCC9	LCC8
	LCC7	LCC6	LCC5	LCC4	LCC3	LCC2	LCC1	LCC0
CNDCR	CNDC31	CNDC30	CNDC29	CNDC28	CNDC27	CNDC26	CNDC25	CNDC2
	CNDC23	CNDC22	CNDC21	CNDC20	CNDC19	CNDC18	CNDC17	CNDC1
	CNDC15	CNDC14	CNDC13	CNDC12	CNDC11	CNDC10	CNDC9	CNDC8
	CNDC7	CNDC6	CNDC5	CNDC4	CNDC3	CNDC2	CNDC1	CNDC0
CEFCR	CEFC31	CEFC30	CEFC29	CEFC28	CEFC27	CEFC26	CEFC25	CEFC2
	CEFC23	CEFC22	CEFC21	CEFC20	CEFC19	CEFC18	CEFC17	CEFC1
	CEFC15	CEFC14	CEFC13	CEFC12	CEFC11	CEFC10	CEFC9	CEFC8
	CEFC7	CEFC6	CEFC5	CEFC4	CEFC3	CEFC2	CEFC1	CEFC0
FRECR	FREC31	FREC30	FREC29	FREC28	FREC27	FREC26	FREC25	FREC2
	FREC23	FREC22	FREC21	FREC20	FREC19	FREC18	FREC17	FREC1
	FREC15	FREC14	FREC13	FREC12	FREC11	FREC10	FREC9	FREC8
	FREC7	FREC6	FREC5	FREC4	FREC3	FREC2	FREC1	FREC0

	TLFC7	TLFC6	TLFC5	TLFC4	TLFC3	TLFC2	TLFC1	TLFC0
RFCR	RFC31	RFC30	RFC29	RFC28	RFC27	RFC26	RFC25	RFC24
	RFC23	RFC22	RFC21	RFC20	RFC19	RFC18	RFC17	RFC16
	RFC15	RFC14	RFC13	RFC12	RFC11	RFC10	RFC9	RFC8
	RFC7	RFC6	RFC5	RFC4	RFC3	RFC2	RFC1	RFC0
MAFCR	MAFC31	MAFC30	MAFC29	MAFC28	MAFC27	MAFC26	MAFC25	MAFC24
	MAFC23	MAFC22	MAFC21	MAFC20	MAFC19	MAFC18	MAFC17	MAFC16
	MAFC15	MAFC14	MAFC13	MAFC12	MAFC11	MAFC10	MAFC9	MAFC8
	MAFC7	MAFC6	MAFC5	MAFC4	MAFC3	MAFC2	MAFC1	MAFC0
IPGR	—	—	—	—	—	—	—	—
	—	—	—	—	—	—	—	—
	—	—	—	—	—	—	—	—
	—	—	—	IPG4	IPG3	IPG2	IPG1	IPG0
APR	—	—	—	—	—	—	—	—
	—	—	—	—	—	—	—	—
	AP15	AP14	AP13	AP12	AP11	AP10	AP9	AP8
	AP7	AP6	AP5	AP4	AP3	AP2	AP1	AP0
MPR	—	—	—	—	—	—	—	—
	—	—	—	—	—	—	—	—
	MP15	MP14	MP13	MP12	MP11	MP10	MP9	MP8
	MP7	MP6	MP5	MP4	MP3	MP2	MP1	MP0
TPAUSER	—	—	—	—	—	—	—	—
	—	—	—	—	—	—	—	—
	TPAUSE15	TPAUSE14	TPAUSE13	TPAUSE12	TPAUSE11	TPAUSE10	TPAUSE9	TPAUSE8
	TPAUSE7	TPAUSE6	TPAUSE5	TPAUSE4	TPAUSE3	TPAUSE2	TPAUSE1	TPAUSE0

	—	—	—	—	—	—	—	TR
EDRRR	—	—	—	—	—	—	—	—
	—	—	—	—	—	—	—	—
	—	—	—	—	—	—	—	—
	—	—	—	—	—	—	—	RR
TDLAR	TDLA31	TDLA30	TDLA29	TDLA28	TDLA27	TDLA26	TDLA25	TDLA24
	TDLA23	TDLA22	TDLA21	TDLA20	TDLA19	TDLA18	TDLA17	TDLA16
	TDLA15	TDLA14	TDLA13	TDLA12	TDLA11	TDLA10	TDLA9	TDLA8
	TDLA7	TDLA6	TDLA5	TDLA4	TDLA3	TDLA2	TDLA1	TDLA0
RDLAR	RDLA31	RDLA30	RDLA29	RDLA28	RDLA27	RDLA26	RDLA25	RDLA24
	RDLA23	RDLA22	RDLA21	RDLA20	RDLA19	RDLA18	RDLA17	RDLA16
	RDLA15	RDLA14	RDLA13	RDLA12	RDLA11	RDLA10	RDLA9	RDLA8
	RDLA7	RDLA6	RDLA5	RDLA4	RDLA3	RDLA2	RDLA1	RDLA0
EESR	—	TWB	—	—	—	TABT	RABT	RFCOF
	ADE	ECI	TC	TDE	TFUF	FR	RDE	RFOF
	—	—	—	—	CND	DLC	CD	TRO
	RMAF	—	—	RRF	RTLF	RTSF	PRE	CERF
EESIPR	—	TWBIP	—	—	—	TABTIP	RABTIP	RFCOF
	ADEIP	ECIIP	TCIP	TDEIP	TFUFIP	FRIP	RDEIP	RFOFIP
	—	—	—	—	CNDIP	DLCIP	CDIP	TROIP
	RMAFIP	—	—	RRFIP	RTLFIP	RTSFIP	PREIP	CERFIP
TRSCER	—	—	—	—	—	—	—	—
	—	—	—	—	—	—	—	—
	—	—	—	—	—	—	—	—
	RMAFCE	—	—	RRFCE	—	—	—	—

FDR	—	—	—	—	—	—	—	—
	—	—	—	—	—	—	—	—
	—	—	—	—	—	TFD2	TFD1	TFD0
	—	—	—	—	—	RFD2	RFD1	RFD0
RMCR	—	—	—	—	—	—	—	—
	—	—	—	—	—	—	—	—
	—	—	—	—	—	—	—	—
	—	—	—	—	—	—	—	RNC
FCFTR	—	—	—	—	—	—	—	—
	—	—	—	—	—	RFF2	RFF1	RFF0
	—	—	—	—	—	—	—	—
	—	—	—	—	—	RFD2	RFD1	RFD0
TRIMD	—	—	—	—	—	—	—	—
	—	—	—	—	—	—	—	—
	—	—	—	—	—	—	—	—
	—	—	—	—	—	—	—	TIS
RBWAR	RBWA31	RBWA30	RBWA29	RBWA28	RBWA27	RBWA26	RBWA25	RBWA24
	RBWA23	RBWA22	RBWA21	RBWA20	RBWA19	RBWA18	RBWA17	RBWA16
	RBWA15	RBWA14	RBWA13	RBWA12	RBWA11	RBWA10	RBWA9	RBWA8
	RBWA7	RBWA6	RBWA5	RBWA4	RBWA3	RBWA2	RBWA1	RBWA0
RDFAR	RDFA31	RDFA30	RDFA29	RDFA28	RDFA27	RDFA26	RDFA25	RDFA24
	RDFA23	RDFA22	RDFA21	RDFA20	RDFA19	RDFA18	RDFA17	RDFA16
	RDFA15	RDFA14	RDFA13	RDFA12	RDFA11	RDFA10	RDFA9	RDFA8
	RDFA7	RDFA6	RDFA5	RDFA4	RDFA3	RDFA2	RDFA1	RDFA0

ECBRR	—	—	—	—	—	—	—	RTM
IFR0	BRST	EP1FULL	EP2TR	EP2EMPTY	SETUPTS	EP0oTS	EP0iTR	EP0iTS
IFR1	—	—	—	—	VBUSMN	EP3TR	EP3TS	VBUSF
IFR2	—	—	SURSS	SURSF	CFDN	—	SETC	SETI
IER0	BRST	EP1FULL	EP2TR	EP2EMPTY	SETUPTS	EP0oTS	EP0iTR	EP0iTS
IER1	—	—	—	—	—	EP3TR	EP3TS	VBUSF
IER2	—	—	—	SURSE	CFDN	—	SETCE	SETIE
ISR0	BRST	EP1FULL	EP2TR	EP2EMPTY	SETUPTS	EP0oTS	EP0iTR	EP0iTS
ISR1	—	—	—	—	—	EP3TR	EP3TS	VBUSF
ISR2	SSRSME	—	—	SURSE	CFDN	—	SETCE	SETIE
EPDR0i	D7	D6	D5	D4	D3	D2	D1	D0
EPDR0o	D7	D6	D5	D4	D3	D2	D1	D0
EPDR0s	D7	D6	D5	D4	D3	D2	D1	D0
EPDR1	D7	D6	D5	D4	D3	D2	D1	D0
EPDR2	D7	D6	D5	D4	D3	D2	D1	D0
EPDR3	D7	D6	D5	D4	D3	D2	D1	D0
EPSZ0o	—	—	—	—	—	—	—	—
EPSZ1	—	—	—	—	—	—	—	—
DASTS	—	—	EP3DE	EP2DE	—	—	—	EP0iDE
FCLR	—	EP3CLR	EP1CLR	EP2CLR	—	—	—	EP0iCL
EPSTL	—	—	—	—	EP3STL	EP2STL	EP1STL	EP0STL
TRG	—	EP3KTE	EP1RDFN	EP2PKTE	—	EP0sRDFN	EP0oRDFN	EP0iPK
DMA	—	—	—	—	—	PULLUP_E	EP2DMAE	EP1DM
CVR	CNFV1	CNFV0	INTV1	INTV0	—	ALTV2	ALTV1	ALTV0

FIER	—	—	—	—	EDSSI	ELSI	ETBEI	ERBFI
FDLH	bit 7	bit 6	bit 5	bit 4	bit 3	bit 2	bit 1	bit 0
FIIR	FIFOE1	FIFOE0	—	—	INTID2	INTID1	INTID0	INTPEND
FFCR	RCVRTRIG1	RCVRTRIG0	—	—	DMAMODE	XMITFRST	RCVRFRST	FIFOE
FLCR	DLAB	BREAK	STICKPARITY	EPS	PEN	STOP	CLS1	CLS0
FMCR	—	—	—	LOOPBACK	OUT2	OUT1	RTS	DTR
FLSR	RXFIFOERR	TEMT	THRE	BI	FE	PE	OE	DR
FMSR	DCD	RI	DSR	CTS	DDCD	TERI	DDSR	DCTS
FSCR	bit 7	bit 6	bit 5	bit 4	bit 3	bit 2	bit 1	bit 0
SCIFCR	SCIFOE1	SCIFOE0	—	OUT2LOOP	CKSEL1	CKSEL0	SCIFRST	REGRST
SSCRH	MSS	BIDE	—	SOL	SOLP	SCKS	CSS1	CSS0
SSCRL	—	SSUMS	SRES	—	—	—	DATS1	DATS0
SSMR	MLS	CPOS	CPHS	—	—	CKS2	CKS1	CKS0
SSER	TE	RE	—	—	TEIE	TIE	RIE	CEIE
SSSR	—	ORER	—	—	TEND	TDRE	RDRF	CE
SSCR2	SDOS	SSCKOS	SCSOS	TENDSTS	SCSATS	SSODTS	—	—
SSTDR0	bit7	bit6	bit5	bit4	bit3	bit2	bit1	bit0
SSTDR1	bit7	bit6	bit5	bit4	bit3	bit2	bit1	bit0
SSTDR2	bit7	bit6	bit5	bit4	bit3	bit2	bit1	bit0
SSTDR3	bit7	bit6	bit5	bit4	bit3	bit2	bit1	bit0
SSRDR0	bit7	bit6	bit5	bit4	bit3	bit2	bit1	bit0

BTCSR1	RSTRENBL	HRSTIE	IRQCRIE	BEVTIE	B2HIE	H2BIE	CRRPIE	CRWPI
BTCR	B_BUSY	H_BUSY	OEM0	BEVT_ATN	B2H_ATN	H2B_ATN	CLR_RD_ PTR	CLR_W PTR
BTIMSR	BMC_ HWRST	—	—	OEM3	OEM2	OEM1	B2H_IRQ	B2H_IR EN
SMICFLG	RX_DATA_ RDY	TX_DATA_ RDY	—	SMI	SEVT_ATN	SMS_ATN	—	BUSY
HICR5	—	—	—	—	—	—	SCIFE	—
SMICCSR	bit 7	bit 6	bit 5	bit 4	bit 3	bit 2	bit 1	bit 0
SMICDTR	bit 7	bit 6	bit 5	bit 4	bit 3	bit 2	bit 1	bit 0
SMICIR0	—	—	—	HDTWI	HDTRI	STARI	CTLWI	BUSYI
SMICIR1	—	—	—	HDTWIE	HDTRIE	STARIE	CTLWIE	BUSYIE
SIRQCR3	—	—	—	—	SCSIRQ3	SCSIRQ2	SCSIRQ1	SCSIRQ
TWR0MW	bit 7	bit 6	bit 5	bit 4	bit 3	bit 2	bit 1	bit 0
TWR0SW	bit 7	bit 6	bit 5	bit 4	bit 3	bit 2	bit 1	bit 0
TWR1	bit 7	bit 6	bit 5	bit 4	bit 3	bit 2	bit 1	bit 0
TWR2	bit 7	bit 6	bit 5	bit 4	bit 3	bit 2	bit 1	bit 0
TWR3	bit 7	bit 6	bit 5	bit 4	bit 3	bit 2	bit 1	bit 0
TWR4	bit 7	bit 6	bit 5	bit 4	bit 3	bit 2	bit 1	bit 0
TWR5	bit 7	bit 6	bit 5	bit 4	bit 3	bit 2	bit 1	bit 0
TWR6	bit 7	bit 6	bit 5	bit 4	bit 3	bit 2	bit 1	bit 0
TWR7	bit 7	bit 6	bit 5	bit 4	bit 3	bit 2	bit 1	bit 0

TWR15	bit 7	bit 6	bit 5	bit 4	bit 3	bit 2	bit 1	bit 0
IDR3	bit 7	bit 6	bit 5	bit 4	bit 3	bit 2	bit 1	bit 0
ODR3	bit 7	bit 6	bit 5	bit 4	bit 3	bit 2	bit 1	bit 0
STR3*[1]	IBF3B	OBF3B	MWMF	SWMF	C/$\overline{D}$3	DBU32	IBF3A	OBF3A
STR3*[2]	DBU37	DBU36	DBU35	DBU34	C/$\overline{D}$3	DBU32	IBF3A	OBF3A
SIRQCR4	IRQ15E	IRQ14E	IRQ13E	IRQ8E	IRQ7E	IRQ5E	IRQ4E	IRQ3E
LADR3H	bit 15	bit 14	bit 13	bit 12	bit 11	bit 10	bit 9	bit 8
LADR3L	bit 7	bit 6	bit 5	bit 4	bit 3	—	bit 1	TWRE
SIRQCR0	Q/$\overline{C}$	SELREQ	IEDIR2	SMIE3B	SMIE3A	SMIE2	IRQ12E1	IRQ1E1
SIRQCR1	IRQ11E3	IRQ10E3	IRQ9E3	IRQ6E3	IRQ11E2	IRQ10E2	IRQ9E2	IRQ6E2
IDR1	bit 7	bit 6	bit 5	bit 4	bit 3	bit 2	bit 1	bit 0
ODR1	bit 7	bit 6	bit 5	bit 4	bit 3	bit 2	bit 1	bit 0
STR1	DBU17	DBU16	DBU15	DBU14	C/$\overline{D}$1	DBU12	IBF1	OBF1
SIRQCR5	SELIRQ15	SELIRQ14	SELIRQ13	SELIRQ8	SELIRQ7	SELIRQ5	SELIRQ4	SELIRQ3
IDR2	bit 7	bit 6	bit 5	bit 4	bit 3	bit 2	bit 1	bit 0
ODR2	bit 7	bit 6	bit 5	bit 4	bit 3	bit 2	bit 1	bit 0
STR2	DBU27	DBU26	DBU25	DBU24	C/$\overline{D}$2	DBU22	IBF2	OBF2
HISEL	SELSTR3	SELIRQ11	SELIRQ10	SELIRQ9	SELIRQ6	SELSMI	SELIRQ12	SELIRQ1
HICR0	LPC3E	LPC2E	LPC1E	FGA20E	SDWNE	PMEE	LSMIE	LSCIE
HICR1	LPCBSY	CLKREQ	IRQBSY	LRSTB	SDWNB	PMEB	LSMIB	LSCIB
HICR2	GA20	LRST	SDWN	ABRT	IBFIE3	IBFIE2	IBFIE1	ERRIE
HICR3	LFRAME	CLKRUN	SERIRQ	LRESET	LPCPD	PME	LSMI	LSCI

SCIF ADRL	bit 7	bit 6	bit 5	bit 4	bit 3	—	—	—
SUBMSTPBH	SMSTPB15	SMSTPB14	SMSTPB13	SMSTPB12	SMSTPB11	SMSTPB10	SMSTPB9	SMSTP
SUBMSTPBL	SMSTPB7	SMSTPB6	SMSTPB5	PECI	SCIF	SMSTPB2	LPC	SMSTP
ECS	E15	E14	E13	E12	E11	E10	E9	E8
	E7	E6	E5	E4	E3	E2	E1	E0
ECCR	EDSB	—	—	—	ECSB3	ECSB2	ECSB1	ECSB0
MSTPCRA	MSTPA7	MSTPA6	MSTPA5	MSTPA4	MSTPA3	MSTPA2	MSTPA1	MSTPA
P3NCE	P37NCE	P36NCE	P35NCE	P34NCE	P33NCE	P32NCE	P31NCE	P30NCE
P3NCMC	P37NCMC	P36NCMC	P35NCMC	P34NCMC	P33NCMC	P32NCMC	P31NCMC	P30NCM
NCCS	—	—	—	—	—	NCCK2	NCCK1	NCCK0
PEODR	PE7ODR	PE6ODR	PE5ODR	PE4ODR	PE3ODR	PE2ODR	PE1ODR	PE0OD
PFODR	—	PF6ODR	PF5ODR	PF4ODR	PF3ODR	PF2ODR	PF1ODR	PF0ODI
PEPIN	PE7PIN	PE6PIN	PE5PIN	PE4PIN	PE3PIN	PE2PIN	PE1PIN	PE0PIN
PEDDR	PE7DDR	PE6DDR	PE5DDR	PE4DDR	PE3DDR	PE2DDR	PE1DDR	PE0DDI
PFPIN	—	PF6PIN	PF5PIN	PF4PIN	PF3PIN	PF2PIN	PF1PIN	PF0PIN
PFDDR	—	PF6DDR	PF5DDR	PF4DDR	PF3DDR	PF2DDR	PF1DDR	PF0DDI
PCODR	PC7ODR	PC6ODR	PC5ODR	PC4ODR	PC3ODR	PC2ODR	PC1ODR	PC0OD
PDODR	PD7ODR	PD6ODR	PD5ODR	PD4ODR	PD3ODR	PD2ODR	PD1ODR	PD0OD
PCPIN	PC7PIN	PC6PIN	PC5PIN	PC4PIN	PC3PIN	PC2PIN	PC1PIN	PC0PIN
PCDDR	PC7DDR	PC6DDR	PC5DDR	PC4DDR	PC3DDR	PC2DDR	PC1DDR	PC0DD
PDPIN	PD7PIN	PD6PIN	PD5PIN	PD4PIN	PD3PIN	PD2PIN	PD1PIN	PD0PIN
PDDDR	PD7DDR	PD6DDR	PD5DDR	PD4DDR	PD3DDR	PD2DDR	PD1DDR	PD0DD

ICSR_4	ESTP	STOP	IRTR	AASX	AL	AAS	ADZ	ACKB
ICDR_4	bit 7	bit 6	bit 5	bit 4	bit 3	bit 2	bit 1	bit 0
SARX_4	SVAX6	SVAX5	SVAX4	SVAX3	SVAX2	SVAX1	SVAX0	FSX
ICMR_4	MLS	WAIT	CKS2	CKS1	CKS0	BC2	BC1	BC0
SAR_4	SVA6	SVA5	SVA4	SVA3	SVA2	SVA1	SVA0	FS
ICCR_5	ICE	IEIC	MST	TRS	ACKE	BBSY	IRIC	SCP
ICSR_5	ESTP	STOP	IRTR	AASX	AL	AAS	ADZ	ACKB
ICDR_5	bit 7	bit 6	bit 5	bit 4	bit 3	bit 2	bit 1	bit 0
SARX_5	SVAX6	SVAX5	SVAX4	SVAX3	SVAX2	SVAX1	SVAX0	FSX
ICMR_5	MLS	WAIT	CKS2	CKS1	CKS0	BC2	BC1	BC0
SAR_5	SVA6	SVA5	SVA4	SVA3	SVA2	SVA1	SVA0	FS
SMR_1※3	C/$\overline{A}$ (GM)	CHR (BLK)	PE (PE)	O/$\overline{E}$ (O/$\overline{E}$)	STOP (BCP1)	MP (BCP0)	CKS1 (CKS1)	CKS0 (CKS0)
BRR_1	bit 7	bit 6	bit 5	bit 4	bit 3	bit 2	bit 1	bit 0
SCR_1	TIE	RIE	TE	RE	MPIE	TEIE	CKE1	CKE0
TDR_1	bit 7	bit 6	bit 5	bit 4	bit 3	bit 2	bit 1	bit 0
SSR_1※3	TDRE (TDRE)	RDRF (RDRF)	ORER (ORER)	FER (ERS)	PER (PER)	TEND (TEND)	MPB (MPB)	MPBT (MPBT)
RDR_1	bit 7	bit 6	bit 5	bit 4	bit 3	bit 2	bit 1	bit 0
SCMR_1	—	—	—	—	SDIR	SINV	—	SMIF

	AD1	AD0	—	—	—	—	—	—
ADDRE	AD9	AD8	AD7	AD6	AD5	AD4	AD3	AD2
	AD1	AD0	—	—	—	—	—	—
ADDRF	AD9	AD8	AD7	AD6	AD5	AD4	AD3	AD2
	AD1	AD0	—	—	—	—	—	—
ADDRG	AD9	AD8	AD7	AD6	AD5	AD4	AD3	AD2
	AD1	AD0	—	—	—	—	—	—
ADDRH	AD9	AD8	AD7	AD6	AD5	AD4	AD3	AD2
	AD1	AD0	—	—	—	—	—	—
ADCSR	ADF	ADIE	ADST	—	—	CH2	CH1	CH0
ADCR	TRGS1	TRGS0	SCANE	SCANS	CKS1	CKS0	ADSTCLR	EXTRG
SMR0	DCD1	RI1	DSR1	SME	—	SM2	SM1	SM0
SMR1	CTS1	DTR1	RTS1	CTS3	—	RTS3	—	—
P4BNCE	P47NCE	P46NCE	P45NCE	P44NCE	PB3NCE	PB2NCE	PB1NCE	PB0NCI
P4BNCMC	P47NCMC	P46NCMC	P45NCMC	P44NCMC	PB3NCMC	PB2NCMC	PB1NCMC	PB0NCI
P6PCR	P67PCR	P66PCR	P65PCR	P64PCR	P63PCR	P62PCR	P61PCR	P60PCI
PINFNCR	—	—	—	—	—	SERIRQ OFF	LPCPD OFF	CLKRU OFF
P4PCR	P47PCR	P46PCR	P45PCR	P44PCR	P43PCR	P42PCR	P41PCR	P40PCI
ICCR_3	ICE	IEIC	MST	TRS	ACKE	BBSY	IRIC	SCP
ICSR_3	ESTP	STOP	IRTR	AASX	AL	AAS	ADZ	ACKB
ICDR_3	bit 7	bit 6	bit 5	bit 4	bit 3	bit 2	bit 1	bit 0
SARX_3	SVAX6	SVAX5	SVAX4	SVAX3	SVAX2	SVAX1	SVAX0	FSX

SAR_2	SVA6	SVA5	SVA4	SVA3	SVA2	SVA1	SVA0	FS
DADRA_1	DA13	DA12	DA11	DA10	DA9	DA8	DA7	DA6
	DA5	DA4	DA3	DA2	DA1	DA0	CFS	—
DACR_1	—	PWME	—	—	OEB	OEA	OS	CKS
DADRB_1	DA13	DA12	DA11	DA10	DA9	DA8	DA7	DA6
	DA5	DA4	DA3	DA2	DA1	DA0	CFS	REGS
DACNT_1	UC7	UC6	UC5	UC4	UC3	UC2	UC1	UC0
	UC8	UC9	UC10	UC11	UC12	UC13	—	REGS
CRCCR	DORCLR	—	—	—	—	LMS	G1	G0
CRCDIR	bit 7	bit 6	bit 5	bit 4	bit 3	bit 2	bit 1	bit 0
CRCDOR	bit 15	bit 14	bit 13	bit 12	bit 11	bit 10	bit 9	bit 8
	bit 7	bit 6	bit 5	bit 4	bit 3	bit 2	bit 1	bit 0
ICXR_0	STOPIM	HNDS	ICDRF	ICDRE	ALIE	ALSL	FNC1	FNC0
ICXR_1	STOPIM	HNDS	ICDRF	ICDRE	ALIE	ALSL	FNC1	FNC0
ICSMBCR	SMB5E	SMB4E	SMB3E	SMB2E	SMB1E	SMB0E	FSEL1	FSEL0
ICXR_2	STOPIM	HNDS	ICDRF	ICDRE	ALIE	ALSL	FNC1	FNC0
ICXR_3	STOPIM	HNDS	ICDRF	ICDRE	ALIE	ALSL	FNC1	FNC0
IICX3	—	—	—	—	TCSS	IICX5	IICX4	IICX3
ICXR_4	STOPIM	HNDS	ICDRF	ICDRE	ALIE	ALSL	FNC1	FNC0
ICXR_5	STOPIM	HNDS	ICDRF	ICDRE	ALIE	ALSL	FNC1	FNC0
KBCOMP	EVENTE	—	—	—	—	—	—	—
DTCERF	DTCEF7	DTCEF6	—	—	—	—	—	—

DTCERA	DTCEA7	DTCEA6	DTCEA5	DTCEA4	DTCEA3	—	—	—
DTCERB	—	DTCEB6	DTCEB5	—	—	—	—	—
DTCERC	—	—	—	DTCEC4	—	DTCEC2	DTCEC1	DTCEC
DTCERD	DTCED7	—	—	DTCED4	DTCED3	—	—	—
DTCERE	—	—	—	—	DTCEE3	DTCEE2	DTCEE1	DTCEE
DTVECR	SWDTE	DTVEC6	DTVEC5	DTVEC4	DTVEC3	DTVEC2	DTVEC1	DTVEC
ABRKCR	CMF	—	—	—	—	—	—	BIE
BARA	A23	A22	A21	A20	A19	A18	A17	A16
BARB	A15	A14	A13	A12	A11	A10	A9	A8
BARC	A7	A6	A5	A4	A3	A2	A1	—
IER16	IRQ15E	IRQ14E	IRQ13E	IRQ12E	IRQ11E	IRQ10E	IRQ9E	IRQ8E
ISR16	IRQ15F	IRQ14F	IRQ13F	IRQ12F	IRQ11F	IRQ10F	IRQ9F	IRQ8F
ISCR16H	IRQ15SCB	IRQ15SCA	IRQ14SCB	IRQ14SCA	IRQ13SCB	IRQ13SCA	IRQ12SCB	IRQ12S
ISCR16L	IRQ11SCB	IRQ11SCA	IRQ10SCB	IRQ10SCA	IRQ9SCB	IRQ9SCA	IRQ8SCB	IRQ8SC
ISSR16	ISS15	ISS14	ISS13	ISS12	ISS11	ISS10	ISS9	ISS8
ISSR	ISS7	ISS6	ISS5	ISS4	ISS3	ISS2	ISS1	ISS0
PTCNT0	SCPFSEL1	SCPFSEL3	—	—	PWMXS	—	OBE	—
BCR2	—	—	—	—	ADFULLE	EXCKS	—	—
WSCR2	WM10	WC11	WC10	—	—	—	—	—
PCSR	PWCKX1B	PWCKX1A	PWCKX0B	PWCKX0A	PWCKX1C	—	—	PWCKX
SYSCR2	—	—	—	—	ADMXE	—	—	—
SBYCR	SSBY	STS2	STS1	STS0	DTSPEED	SCK2	SCK1	SCK0
LPWRCR	—	—	NESEL	EXCLE	—	PNCCS	PNCAH	—
MSTPCRH	MSTP15	MSTP14	MSTP13	MSTP12	MSTP11	MSTP10	MSTP9	MSTP8

TIER	—	—	—	—	OCIAE	OCIBE	OVIE	—
TCSR	—	—	—	—	OCFA	OCFB	OVF	CCLRA
FRC	bit 15	bit 14	bit 13	bit 12	bit 11	bit 10	bit 9	bit 8
	bit 7	bit 6	bit 5	bit 4	bit 3	bit 2	bit 1	bit 0
OCRA	bit 15	bit 14	bit 13	bit 12	bit 11	bit 10	bit 9	bit 8
	bit 7	bit 6	bit 5	bit 4	bit 3	bit 2	bit 1	bit 0
OCRB	bit 15	bit 14	bit 13	bit 12	bit 11	bit 10	bit 9	bit 8
	bit 7	bit 6	bit 5	bit 4	bit 3	bit 2	bit 1	bit 0
TCR	—	—	—	—	—	—	CKS1	CKS0
TOCR	—	OCRAMS	ICRS	OCRS	—	—	—	—
OCRAR	bit 15	bit 14	bit 13	bit 12	bit 11	bit 10	bit 9	bit 8
	bit 7	bit 6	bit 5	bit 4	bit 3	bit 2	bit 1	bit 0
OCRAF	bit 15	bit 14	bit 13	bit 12	bit 11	bit 10	bit 9	bit 8
	bit 7	bit 6	bit 5	bit 4	bit 3	bit 2	bit 1	bit 0
DADRA_0	DA13	DA12	DA11	DA10	DA9	DA8	DA7	DA6
	DA5	DA4	DA3	DA2	DA1	DA0	CFS	—
DACR_0	—	PWME	—	—	OEB	OEA	OS	CKS
DADRB_0	DA13	DA12	DA11	DA10	DA9	DA8	DA7	DA6
	DA5	DA4	DA3	DA2	DA1	DA0	CFS	REGS
DACNT_0	UC7	UC6	UC5	UC4	UC3	UC2	UC1	UC0
	UC8	UC9	UC10	UC11	UC12	UC13	—	REGS
TCSR_0	OVF	WT/$\overline{IT}$	TME	—	RST/$\overline{NMI}$	CKS2	CKS1	CKS0
TCNT_0	bit 7	bit 6	bit 5	bit 4	bit 3	bit 2	bit 1	bit 0

P2DDR	P27DDR	P26DDR	P25DDR	P24DDR	P23DDR	P22DDR	P21DDR	P20DDR
P1DR	P17DR	P16DR	P15DR	P14DR	P13DR	P12DR	P11DR	P10DR
P2DR	P27DR	P26DR	P25DR	P24DR	P23DR	P22DR	P21DR	P20DR
P3DDR	P37DDR	P36DDR	P35DDR	P34DDR	P33DDR	P32DDR	P31DDR	P30DDR
P4DDR	P47DDR	P46DDR	P45DDR	P44DDR	P43DDR	P42DDR	P41DDR	P40DDR
P3DR	P37DR	P36DR	P35DR	P34DR	P33DR	P32DR	P31DR	P30DR
P4DR	P47DR	P46DR	P45DR	P44DR	P43DR	P42DR	P41DR	P40DR
P5DDR	P57DDR	P56DDR	P55DDR	P54DDR	P53DDR	P52DDR	P51DDR	P50DDR
P6DDR	P67DDR	P66DDR	P65DDR	P64DDR	P63DDR	P62DDR	P61DDR	P60DDR
P5DR	P57DR	P56DR	P55DR	P54DR	P53DR	P52DR	P51DR	P50DR
P6DR	P67DR	P66DR	P65DR	P64DR	P63DR	P62DR	P61DR	P60DR
PBODR	PB7ODR	PB6ODR	PB5ODR	PB4ODR	PB3ODR	PB2ODR	PB1ODR	PB0ODR
PBPIN	PB7PIN	PB6PIN	PB5PIN	PB4PIN	PB3PIN	PB2PIN	PB1PIN	PB0PIN
P8DDR	P87DDR	P86DDR	P85DDR	P84DDR	P83DDR	P82DDR	P81DDR	P80DDR
P7PIN	P77PIN	P76PIN	P75PIN	P74PIN	P73PIN	P72PIN	P71PIN	P70PIN
PBDDR	PB7DDR	PB6DDR	PB5DDR	PB4DDR	PB3DDR	PB2DDR	PB1DDR	PB0DDR
P8DR	P87DR	P86DR	P85DR	P84DR	P83DR	P82DR	P81DR	P80DR
P9DDR	P97DDR	P96DDR	P95DDR	P94DDR	P93DDR	P92DDR	P91DDR	P90DDR
P9DR	P97DR	P96DR	P95DR	P94DR	P93DR	P92DR	P91DR	P90DR
IER	IRQ7E	IRQ6E	IRQ5E	IRQ4E	IRQ3E	IRQ2E	IRQ1E	IRQ0E
STCR	IICX2	IICX1	IICX0	—	FLSHE	—	ICKS1	ICKS0
SYSCR	CS256E	IOSE	INTM1	INTM0	XRST	NMIEG	—	RAME
MDCR	EXPE	—	—	—	—	MDS2	MDS1	—

TCORA_1	bit 7	bit 6	bit 5	bit 4	bit 3	bit 2	bit 1	bit 0
TCORB_0	bit 7	bit 6	bit 5	bit 4	bit 3	bit 2	bit 1	bit 0
TCORB_1	bit 7	bit 6	bit 5	bit 4	bit 3	bit 2	bit 1	bit 0
TCNT_0	Bit7	bit 6	bit 5	bit 4	bit 3	bit 2	bit 1	bit 0
TCNT_1	bit 7	bit 6	bit 5	bit 4	bit 3	bit 2	bit 1	bit 0
ICCR_0	ICE	IEIC	MST	TRS	ACKE	BBSY	IRIC	SCP
ICSR_0	ESTP	STOP	IRTR	AASX	AL	AAS	ADZ	ACKB
ICDR_0	bit 7	bit 6	bit 5	bit 4	bit 3	bit 2	bit 1	bit 0
SARX_0	SVAX6	SVAX5	SVAX4	SVAX3	SVAX2	SVAX1	SVAX0	FSX
ICMR_0	MLS	WAIT	CKS2	CKS1	CKS0	BC2	BC1	BC0
SAR_0	SVA6	SVA5	SVA4	SVA3	SVA2	SVA1	SVA0	FS
SMR_3*3	C/$\overline{A}$ (GM)	CHR (BLK)	PE (PE)	O/$\overline{E}$ (O/$\overline{E}$)	STOP (BCP1)	MP (BCP0)	CKS1 (CKS1)	CKS0 (CKS0)
BRR_3	bit 7	bit 6	bit 5	bit 4	bit 3	bit 2	bit 1	bit 0
SCR_3	TIE	RIE	TE	RE	MPIE	TIE	CKE1	CKE0
TDR_3	bit 7	bit 6	bit 5	bit 4	bit 3	bit 2	bit 1	bit 0
SSR_3*3	TDRE (TDRE)	RDRF (RDRF)	ORER (ORER)	FER (ERS)	PER (PER)	TEND (TEND)	MPB (MPB)	MPBT (MPBT)
RDR_3	bit 7	bit 6	bit 5	bit 4	bit 3	bit 2	bit 1	bit 0
SCMR_3	—	—	—	—	SDIR	SINV	—	SMIF
TCSR_1	OVF	WT/$\overline{IT}$	TME	PSS	RST/$\overline{NMI}$	CKS2	CKS1	CKS0
TCNT_1	bit 7	bit 6	bit 5	bit 4	bit 3	bit 2	bit 1	bit 0
TCR_X	CMIEB	CMIEA	OVIE	—	—	CKS2	CKS1	CKS0
TCSR_X	CMFB	CMFA	OVF	—	—	—	—	—

TCNT_Y	bit 7	bit 6	bit 5	bit 4	bit 3	bit 2	bit 1	bit 0
TCONRS	TMRX/Y	—	—	—	—	—	—	—

Notes: The registers related to USB are supported only by the H8S/2472 Group.

The registers related to PECI are supported only by the H8S/2472 Group and the Group

1. When TWRE = 1 or SELSTR3 = 0
2. When TWRE = 0 and SELSTR3 = 1
3. Some Bits have different names in normal mode and smart card interface mo Bit name in smart card interface mode is enclosed in parentheses.

RFLR	Initialized	Initialized	—	—	—	—	Initialized
PSR	Initialized	Initialized	—	—	—	—	Initialized
TROCR	Initialized	Initialized	—	—	—	—	Initialized
CDCR	Initialized	Initialized	—	—	—	—	Initialized
LCCR	Initialized	Initialized	—	—	—	—	Initialized
CNDCR	Initialized	Initialized	—	—	—	—	Initialized
CEFCR	Initialized	Initialized	—	—	—	—	Initialized
FRECR	Initialized	Initialized	—	—	—	—	Initialized
TSFRCR	Initialized	Initialized	—	—	—	—	Initialized
TLFRCR	Initialized	Initialized	—	—	—	—	Initialized
RFCR	Initialized	Initialized	—	—	—	—	Initialized
MAFCR	Initialized	Initialized	—	—	—	—	Initialized
IPGR	Initialized	Initialized	—	—	—	—	Initialized
APR	Initialized	Initialized	—	—	—	—	Initialized
MPR	Initialized	Initialized	—	—	—	—	Initialized
TPAUSER	Initialized	Initialized	—	—	—	—	Initialized
EDMR	Initialized	Initialized	—	—	—	—	Initialized
EDTRR	Initialized	Initialized	—	—	—	—	Initialized
EDRRR	Initialized	Initialized	—	—	—	—	Initialized
TDLAR	Initialized	Initialized	—	—	—	—	Initialized
RDLAR	Initialized	Initialized	—	—	—	—	Initialized

TRIMD	Initialized	Initialized	—	—	—	—	Initialized
RBWAR	Initialized	Initialized	—	—	—	—	Initialized
RDFAR	Initialized	Initialized	—	—	—	—	Initialized
TBRAR	Initialized	Initialized	—	—	—	—	Initialized
TDFAR	Initialized	Initialized	—	—	—	—	Initialized
ECBRR	Initialized	Initialized	—	—	—	—	Initialized
IFR0	Initialized	Initialized	—	—	—	—	Initialized
IFR1	Initialized	Initialized	—	—	—	—	Initialized
IFR2	Initialized	Initialized	—	—	—	—	Initialized
IER0	Initialized	Initialized	—	—	—	—	Initialized
IER1	Initialized	Initialized	—	—	—	—	Initialized
IER2	Initialized	Initialized	—	—	—	—	Initialized
ISR0	Initialized	Initialized	—	—	—	—	Initialized
ISR1	Initialized	Initialized	—	—	—	—	Initialized
ISR2	Initialized	Initialized	—	—	—	—	Initialized
EPDR0i	Initialized	Initialized	—	—	—	—	Initialized
EPDR0o	Initialized	Initialized	—	—	—	—	Initialized
EPDR0s	Initialized	Initialized	—	—	—	—	Initialized
EPDR1	Initialized	Initialized	—	—	—	—	Initialized
EPDR2	Initialized	Initialized	—	—	—	—	Initialized
EPDR3	Initialized	Initialized	—	—	—	—	Initialized

CTLR	Initialized	Initialized	—	—	—	—	Initialized
EPIR	Initialized	Initialized	—	—	—	—	Initialized
TRNTREG0	Initialized	Initialized	—	—	—	—	Initialized
TRNTREG1	Initialized	Initialized	—	—	—	—	Initialized
FRBR	Initialized	Initialized	—	—	—	—	Initialized
FTHR	Initialized	Initialized	—	—	—	—	Initialized
FDLL	Initialized	Initialized	—	—	—	—	Initialized
FIER	Initialized	Initialized	—	—	—	—	Initialized
FDLH	Initialized	Initialized	—	—	—	—	Initialized
FIIR	Initialized	Initialized	—	—	—	—	Initialized
FFCR	Initialized	Initialized	—	—	—	—	Initialized
FLCR	Initialized	Initialized	—	—	—	—	Initialized
FMCR	Initialized	Initialized	—	—	—	—	Initialized
FLSR	Initialized	Initialized	—	—	—	—	Initialized
FMSR	Initialized	Initialized	—	—	—	—	Initialized
FSCR	Initialized	Initialized	—	—	—	—	Initialized
SCIFCR	Initialized	Initialized	—	—	—	—	Initialized
SSCRH	Initialized	Initialized	—	—	—	—	Initialized
SSCRL	Initialized	Initialized	—	—	—	—	Initialized
SSMR	Initialized	Initialized	—	—	—	—	Initialized
SSER	Initialized	Initialized	—	—	—	—	Initialized

SSRDR2	Initialized	Initialized	—	—	—	—	Initialized
SSRDR3	Initialized	Initialized	—	—	—	—	Initialized
SSTRSR	Initialized	Initialized	—	—	—	—	Initialized
HICR4	Initialized	Initialized	—	—	—	—	Initialized
BTSR0	Initialized	Initialized	—	—	—	—	Initialized
BTSR1	Initialized	Initialized	—	—	—	—	Initialized
BTCSR0	Initialized	Initialized	—	—	—	—	Initialized
BTCSR1	Initialized	Initialized	—	—	—	—	Initialized
BTCR	Initialized	Initialized	—	—	—	—	Initialized
BTIMSR	Initialized	Initialized	—	—	—	—	Initialized
SMICFLG	Initialized	Initialized	—	—	—	—	Initialized
HICR5	Initialized	Initialized	—	—	—	—	Initialized
SMICCSR	—	—	—	—	—	—	—
SMICDTR	—	—	—	—	—	—	—
SMICIR0	Initialized	Initialized	—	—	—	—	Initialized
SMICIR1	Initialized	Initialized	—	—	—	—	Initialized
SIRQCR3	Initialized	Initialized	—	—	—	—	Initialized
TWR0MW	—	—	—	—	—	—	—
TWR0SW	—	—	—	—	—	—	—
TWR1	—	—	—	—	—	—	—
TWR2	—	—	—	—	—	—	—

TWR11	—	—	—	—	—	—	—
TWR12	—	—	—	—	—	—	—
TWR13	—	—	—	—	—	—	—
TWR14	—	—	—	—	—	—	—
TWR15	—	—	—	—	—	—	—
IDR3	—	—	—	—	—	—	—
ODR3	—	—	—	—	—	—	—
STR3	Initialized	Initialized	—	—	—	—	Initialized
SIRQCR4	Initialized	Initialized	—	—	—	—	Initialized
LADR3H	Initialized	Initialized	—	—	—	—	Initialized
LADR3L	Initialized	Initialized	—	—	—	—	Initialized
SIRQCR0	Initialized	Initialized	—	—	—	—	Initialized
SIRQCR1	Initialized	Initialized	—	—	—	—	Initialized
IDR1	—	—	—	—	—	—	—
ODR1	—	—	—	—	—	—	—
STR1	Initialized	Initialized	—	—	—	—	Initialized
SIRQCR5	Initialized	Initialized	—	—	—	—	Initialized
IDR2	—	—	—	—	—	—	—
ODR2	—	—	—	—	—	—	—
STR2	Initialized	Initialized	—	—	—	—	Initialized
HISEL	Initialized	Initialized	—	—	—	—	Initialized

LADR12H	Initialized	Initialized	—	—	—	—	Initialized
LADR12L	Initialized	Initialized	—	—	—	—	Initialized
SCIFADRH	Initialized	Initialized	—	—	—	—	Initialized
SCIFADRL	Initialized	Initialized	—	—	—	—	Initialized
SUBMSTPBH	Initialized	Initialized	—	—	—	—	Initialized
SUBMSTPBL	Initialized	Initialized	—	—	—	—	Initialized
ECS	Initialized	Initialized	—	—	—	—	Initialized
ECCR	Initialized	Initialized	—	—	—	—	Initialized
MSTPCRA	Initialized	Initialized	—	—	—	—	Initialized
P3NCE	Initialized	Initialized	—	—	—	—	Initialized
P3NCMC	Initialized	Initialized	—	—	—	—	Initialized
NCCS	Initialized	Initialized	—	—	—	—	Initialized
PEODR	Initialized	Initialized	—	—	—	—	Initialized
PFODR	Initialized	—	—	—	—	—	Initialized
PEPIN	—	—	—	—	—	—	—
PEDDR	Initialized	Initialized	—	—	—	—	Initialized
PFPIN	—	—	—	—	—	—	—
PFDDR	Initialized	—	—	—	—	—	Initialized
PCODR	Initialized	Initialized	—	—	—	—	Initialized
PDODR	Initialized	Initialized	—	—	—	—	Initialized

FMATS	Initialized	Initialized	—	—	—	—	Initialized
FTDAR	Initialized	Initialized	—	—	—	—	Initialized
ICCR_4	Initialized	Initialized	—	—	—	—	Initialized
ICSR_4	Initialized	Initialized	—	—	—	—	Initialized
ICDR_4	—	—	—	—	—	—	—
SARX_4	Initialized	Initialized	—	—	—	—	Initialized
ICMR_4	Initialized	Initialized	—	—	—	—	Initialized
SAR_4	Initialized	Initialized	—	—	—	—	Initialized
ICCR_5	Initialized	Initialized	—	—	—	—	Initialized
ICSR_5	Initialized	Initialized	—	—	—	—	Initialized
ICDR_5	—	—	—	—	—	—	—
SARX_5	Initialized	Initialized	—	—	—	—	Initialized
ICMR_5	Initialized	Initialized	—	—	—	—	Initialized
SAR_5	Initialized	Initialized	—	—	—	—	Initialized
SMR_1	Initialized	Initialized	—	—	—	—	Initialized
BRR_1	Initialized	Initialized	—	—	—	—	Initialized
SCR_1	Initialized	Initialized	—	—	—	—	Initialized
TDR_1	Initialized	Initialized	—	—	Initialized	Initialized	Initialized
SSR_1	Initialized	Initialized	—	—	Initialized	Initialized	Initialized
RDR_1	Initialized	Initialized	—	—	Initialized	Initialized	Initialized
SCMR_1	Initialized	Initialized	—	—	—	—	Initialized

ADCSR	Initialized	Initialized	—	—	Initialized	Initialized	Initialized
ADCR	Initialized	Initialized	—	—	Initialized	Initialized	Initialized
SMR0	Initialized	Initialized	—	—	—	—	Initialized
SMR1	Initialized	Initialized	—	—	—	—	Initialized
P4BNCE	Initialized	Initialized	—	—	—	—	Initialized
P4BNCMC	Initialized	Initialized	—	—	—	—	Initialized
P6PCR	Initialized	Initialized	—	—	Initialized	Initialized	Initialized
PINFNCR	Initialized	Initialized	—	—	Initialized	Initialized	Initialized
P4PCR	Initialized	Initialized	—	—	Initialized	Initialized	Initialized
ICCR_3	Initialized	Initialized	—	—	—	—	Initialized
ICSR_3	Initialized	Initialized	—	—	—	—	Initialized
ICDR_3	—	—	—	—	—	—	—
SARX_3	Initialized	Initialized	—	—	—	—	Initialized
ICMR_3	Initialized	Initialized	—	—	—	—	Initialized
SAR_3	Initialized	Initialized	—	—	—	—	Initialized
ICCR_2	Initialized	Initialized	—	—	—	—	Initialized
ICSR_2	Initialized	Initialized	—	—	—	—	Initialized
ICDR_2	—	—	—	—	—	—	—
SARX_2	Initialized	Initialized	—	—	—	—	Initialized
ICMR_2	Initialized	Initialized	—	—	—	—	Initialized
SAR_2	Initialized	Initialized	—	—	—	—	Initialized

ICXR_0	Initialized	Initialized	—	—	—	—	Initialized
ICXR_1	Initialized	Initialized	—	—	—	—	Initialized
ICSMBCR	Initialized	Initialized	—	—	—	—	Initialized
ICXR_2	Initialized	Initialized	—	—	—	—	Initialized
ICXR_3	Initialized	Initialized	—	—	—	—	Initialized
IICX3	Initialized	Initialized	—	—	—	—	Initialized
ICXR_4	Initialized	Initialized	—	—	—	—	Initialized
ICXR_5	Initialized	Initialized	—	—	—	—	Initialized
KBCOMP	Initialized	Initialized	—	—	—	—	Initialized
DTCERF	Initialized	Initialized	—	—	—	—	Initialized
ICRD	Initialized	Initialized	—	—	—	—	Initialized
ICRA	Initialized	Initialized	—	—	—	—	Initialized
ICRB	Initialized	Initialized	—	—	—	—	Initialized
ICRC	Initialized	Initialized	—	—	—	—	Initialized
ISR	Initialized	Initialized	—	—	—	—	Initialized
ISCRH	Initialized	Initialized	—	—	—	—	Initialized
ISCRL	Initialized	Initialized	—	—	—	—	Initialized
DTCERA	Initialized	Initialized	—	—	—	—	Initialized
DTCERB	Initialized	Initialized	—	—	—	—	Initialized
DTCERC	Initialized	Initialized	—	—	—	—	Initialized
DTCERD	Initialized	Initialized	—	—	—	—	Initialized
DTCERE	Initialized	Initialized	—	—	—	—	Initialized
DTVECR	Initialized	Initialized	—	—	—	—	Initialized

ISSR16	Initialized	Initialized	—	—	—	—	Initialized
ISSR	Initialized	Initialized	—	—	—	—	Initialized
PTCNT0	Initialized	Initialized	—	—	—	—	Initialized
BCR2	Initialized	Initialized	—	—	—	—	Initialized
WSCR2	Initialized	Initialized	—	—	—	—	Initialized
PCSR	Initialized	Initialized	—	—	—	—	Initialized
SYSCR2	Initialized	Initialized	—	—	—	—	Initialized
SBYCR	Initialized	Initialized	—	—	—	—	Initialized
LPWRCR	Initialized	Initialized	—	—	—	—	Initialized
MSTPCRH	Initialized	Initialized	—	—	—	—	Initialized
MSTPCRL	Initialized	Initialized	—	—	—	—	Initialized
ICCR_1	Initialized	Initialized	—	—	—	—	Initialized
ICSR_1	Initialized	Initialized	—	—	—	—	Initialized
ICDR_1	—	—	—	—	—	—	—
SARX_1	Initialized	Initialized	—	—	—	—	Initialized
ICMR_1	Initialized	Initialized	—	—	—	—	Initialized
SAR_1	Initialized	Initialized	—	—	—	—	Initialized
TIER	Initialized	Initialized	—	—	—	—	Initialized
TCSR	Initialized	Initialized	—	—	—	—	Initialized
FRC	Initialized	Initialized	—	—	—	—	Initialized
OCRA	Initialized	Initialized	—	—	—	—	Initialized
OCRB	Initialized	Initialized	—	—	—	—	Initialized

DACNT_0	Initialized	Initialized	—	—	Initialized	Initialized	Initialized
TCSR_0	Initialized	Initialized	—	—	—	—	Initialized
TCNT_0	Initialized	Initialized	—	—	—	—	Initialized
PAODR	Initialized	Initialized	—	—	—	—	Initialized
PAPIN	—	—	—	—	—	—	—
PADDR	Initialized	Initialized	—	—	—	—	Initialized
P1PCR	Initialized	Initialized	—	—	—	—	Initialized
P2PCR	Initialized	Initialized	—	—	—	—	Initialized
P3PCR	Initialized	Initialized	—	—	—	—	Initialized
P1DDR	Initialized	Initialized	—	—	—	—	Initialized
P2DDR	Initialized	Initialized	—	—	—	—	Initialized
P1DR	Initialized	Initialized	—	—	—	—	Initialized
P2DR	Initialized	Initialized	—	—	—	—	Initialized
P3DDR	Initialized	Initialized	—	—	—	—	Initialized
P4DDR	Initialized	—	—	—	—	—	Initialized
P3DR	Initialized	Initialized	—	—	—	—	Initialized
P4DR	Initialized	—	—	—	—	—	Initialized
P5DDR	Initialized	Initialized	—	—	—	—	Initialized
P6DDR	Initialized	Initialized	—	—	—	—	Initialized
P5DR	Initialized	Initialized	—	—	—	—	Initialized
P6DR	Initialized	Initialized	—	—	—	—	Initialized
PBODR	Initialized	Initialized	—	—	—	—	Initialized

IER	Initialized	Initialized	—	—	—	—	Initialized
STCR	Initialized	Initialized	—	—	—	—	Initialized
SYSCR	Initialized	Initialized	—	—	—	—	Initialized
MDCR	Initialized	Initialized	—	—	—	—	Initialized
BCR	Initialized	Initialized	—	—	—	—	Initialized
WSCR	Initialized	Initialized	—	—	—	—	Initialized
TCR_0	Initialized	Initialized	—	—	—	—	Initialized
TCR_1	Initialized	Initialized	—	—	—	—	Initialized
TCSR_0	Initialized	Initialized	—	—	—	—	Initialized
TCSR_1	Initialized	Initialized	—	—	—	—	Initialized
TCORA_0	Initialized	Initialized	—	—	—	—	Initialized
TCORA_1	Initialized	Initialized	—	—	—	—	Initialized
TCORB_0	Initialized	Initialized	—	—	—	—	Initialized
TCORB_1	Initialized	Initialized	—	—	—	—	Initialized
TCNT_0	Initialized	Initialized	—	—	—	—	Initialized
TCNT_1	Initialized	Initialized	—	—	—	—	Initialized
ICCR_0	Initialized	Initialized	—	—	—	—	Initialized
ICSR_0	Initialized	Initialized	—	—	—	—	Initialized
ICDR_0	—	—	—	—	—	—	—
SARX_0	Initialized	Initialized	—	—	—	—	Initialized
ICMR_0	Initialized	Initialized	—	—	—	—	Initialized
SAR_0	Initialized	Initialized	—	—	—	—	Initialized

TCNT_1	Initialized	Initialized	—	—	—	—	Initialized
TCR_X	Initialized	Initialized	—	—	—	—	Initialized
TCSR_X	Initialized	Initialized	—	—	—	—	Initialized
TCNT_X	Initialized	Initialized	—	—	—	—	Initialized
TCORA_X	Initialized	Initialized	—	—	—	—	Initialized
TCORB_X	Initialized	Initialized	—	—	—	—	Initialized
TCR_Y	Initialized	Initialized	—	—	—	—	Initialized
TCSR_Y	Initialized	Initialized	—	—	—	—	Initialized
TCORA_Y	Initialized	Initialized	—	—	—	—	Initialized
TCORB_Y	Initialized	Initialized	—	—	—	—	Initialized
TCNT_Y	Initialized	Initialized	—	—	—	—	Initialized
TCONRS	Initialized	Initialized	—	—	—	—	Initialized

Notes: 1. The registers related to USB are supported only by the H8S/2472 Group.
2. The registers related to PECI are supported only by the H8S/2472 Group and H8S/2462 Group.

Leave the other FLCI pins open.

Input voltage (pins multiplexed with analog input)	(1)	V_{in}	–0.3 to AVCC + 0.3
Input voltage (pins multiplexed with IIC functions)	(2)	V_{in}	–0.3 to +6.5
Input voltage (pins other than (1) and (2) above)		V_{in}	–0.3 to VCC + 0.3
Reference power supply voltage		AVref	–0.3 to AVCC + 0.3
Analog power supply voltage		AVCC	–0.3 to +4.3
Analog input voltage		V_{AN}	–0.3 to AVCC + 0.3
PECI reference power supply voltage		PEVref	–0.3 to +1.5
Input voltage (PECI)		V_{in}	–0.15 to PEVref + 0.15
Operating temperature		T_{opr}	–20 to +75 (regular specifications
			–40 to +85 (wide temperature specifications)
Operating temperature (when flash memory is programmed or erased)		T_{opr}	0 to +75
Storage temperature		T_{stg}	–55 to +125

Caution: Permanent damage to this LSI may result if absolute maximum ratings are exce

Note: * Voltage applied to the VCC pin.
Make sure power is not applied to the VCL pin.

Item			Symbol	Min.	Typ.	Max.	Unit	Co
Schmitt trigger input voltage	EVENT15 to EVENT0, (Ex)DB7 to (Ex)DB0, $\overline{\text{(Ex)IRQ15}}$ to $\overline{\text{(Ex)IRQ0}}$, $\overline{\text{ETRST}}$, XTAL, EXCL, $\overline{\text{ADTRG}}$, UXTAL, UEXTAL	(1)	V_T^-	VCC × 0.2	—	—	V	
			V_T^+	—	—	VCC × 0.7		
			$V_T^+ - V_T^-$	VCC × 0.05	—	—		
	SCL5 to SCL0, SDA5 to SDA0		V_T^-	VCC × 0.3	—	—		
			V_T^+	—	—	VCC × 0.7		
			$V_T^+ - V_T^-$	VCC × 0.05	—	—		
Input high voltage	$\overline{\text{RES}}$, $\overline{\text{STBY}}$, NMI, FWE, $\overline{\text{MD2}}$, MD1	(2)	V_{IH}	VCC × 0.9	—	VCC + 0.3		
	EXTAL			VCC × 0.7	—	VCC + 0.3		
	Port 7			2.2	—	AVCC + 0.3		
	SCL5 to SCL0, SDA5 to SDA0			—	—	5.5		
	$\overline{\text{CLKRUN}}$, GA20, $\overline{\text{PME}}$, $\overline{\text{LSMI}}$, LSCI, SERIRQ, LAD3 to LAD0, $\overline{\text{LPCPD}}$, LCLK, $\overline{\text{LRESET}}$, $\overline{\text{LFRAME}}$			VCC × 0.5	—	VCC + 0.3		
	RM_REF-CLK, RM_CRS-DV, RM_RXD0, RM_RXD1, RM_RX-ER			2.0	—	VCC + 0.3		
	Input pins other than (1) and (2) above			2.2	—	VCC + 0.3		

	RM_REF-CLK, RM_CRS-DV, RM_RXD0, RM_RXD1, RM_RX-ER		–0.3	—	0.8	
	Input pins other than (1) and (3) above		–0.3	—	VCC × 0.2	
Output high voltage	SCL5 to SCL0, SDA5 to SDA0, $\overline{\text{CLKRUN}}$, GA20, $\overline{\text{PME}}$, $\overline{\text{LSMI}}$, LSCI*2 (4)	V_{OH}	—	—	—	V
	Ports 80 to 83 , C0 to C5, D6, D7*3		0.5	—	—	I
	SERIRQ, LAD3 to LAD0		VCC × 0.9	—	—	I
	RM_TX-EN, RM_TXD0, RM_TXD1		2.4	—	—	I
	Output pins other than (4) above		VCC –0.5	—	—	I
			VCC –1.0	—	—	I
Output low voltage	SCL5 to SCL0, SDA5 to SDA0 (5)	V_{OL}	—	—	0.5	I
			—	—	0.4	I
	$\overline{\text{CLKRUN}}$, GA20, $\overline{\text{PME}}$, $\overline{\text{LSMI}}$, LSCI, SERIRQ, LAD3 to LAD0		—	—	VCC × 0.1	I
	RM_TX-EN, RM_TXD0, RM_TXD1		—	—	0.4	I
	Output pins other than (5) above		—	—	0.4	I
	HC7 to HC0		—	—	1.0	I

current (off state)							
Input pull-up MOS current	Ports 1 to 4, 6, A, D5 to D0	$-I_P$	20	—	300		V_{IN} = 0 V
Supply current*4	Normal operation	I_{CC}	—	45	60	mA	f = 34 MHz, high-spe All modules operatin
	Sleep mode		—	35	45		f = 34 MHz
	Standby mode*5		—	40	100	µA	Ta ≤ 50 °C
			—	—	250		50 °C < Ta
Analog power supply current	During A/D conversion	AI_{cc}	—	1.0	2.0	mA	
	A/D conversion standby		—	2.5	5.0	µA	
Reference power supply current	During A/D conversion	AI_{ref}	—	0.1	1.0	mA	
	A/D conversion standby		—	0.5	5.0	µA	
Input capacitance	All input pin	C_{in}	—	—	10	pF	V_{in} = 0 V, f = 1 MHz, T_a = 25 °C
RAM standby voltage		V_{RAM}	3.0	—	—	V	
VCC start voltage		VCC_{START}	—	0	0.8	V	
VCC rising edge		SVCC	—	—	20	ms/V	

Notes: 1. Do not leave the AVCC, AVref, and AVSS pins open even if the A/D converter is not use
Even if the A/D converter is not used, apply a value in the range from 3.0 V to 3.6 V to th
and AVref pins by connecting them to the power supply (VCC). The relationship betwee
two pins should be AVref ≤ AVCC.

2. An external pull-up resistor is necessary to provide high-level output from SCL5 to SCL0
SDA0 (ICE bit in ICCR is 1), $\overline{\text{CLKRUN}}$, GA20, $\overline{\text{PME}}$, $\overline{\text{LSMI}}$, and $\overline{\text{LSCI}}$.

3. Ports 80 to 83, C0 to C5, D6, and D7 are NMOS push-pull outputs.
High levels on ports 80 to 83, C0 to C5, D6, and D7 are driven by NMOS. An external pu
resistor is necessary to provide high-level output from these pins when they are used as

	Other output pins		—	—	1.6
Permissible output low current (total)	Total of HC7 to HC0	ΣI_{OL}	—	—	48
	Total of all output pins, including the above		—	—	90
Permissible output high current (per pin)	All output pins	$-I_{OH}$	—	—	2
Permissible output high current (total)	Total of all output pins	$\Sigma -I_{OH}$	—	—	60

Notes: 1. To protect LSI reliability, do not exceed the output current values in table 31.3.

2. When driving a Darlington transistor or LED, always insert a current-limiting resistor in t line, as show in figures 31.1 and 31.2.

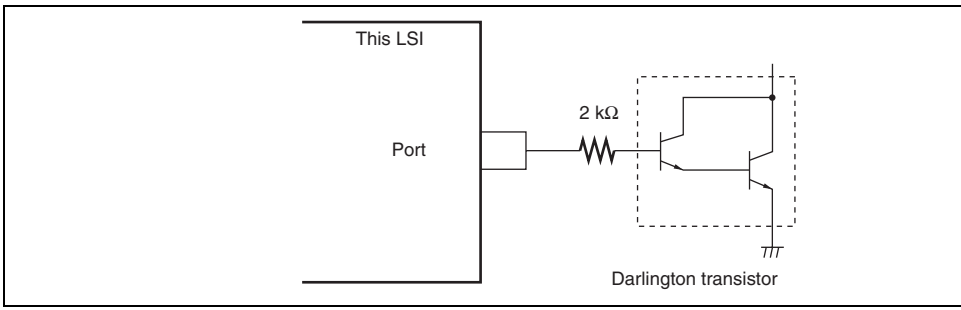


Figure 31.1 Darlington Transistor Drive Circuit (Example)

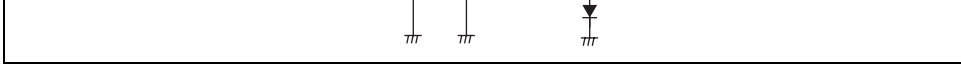


Figure 31.3 Output Load Circuit

31.3.1 Clock Timing

Table 31.4 shows the clock timing. The clock timing specified here covers clock output
clock pulse generator (crystal) and external clock input (EXTAL pin) oscillation stabiliz
times. For details of external clock input (EXTAL pin and EXCL pin) timing, see table
31.6.

Table 31.4 Clock Timing

Conditions: VCC = 3.0 V to 3.6 V, VSS = 0 V, ϕ = 20 MHz to 34 MHz

Item	Symbol	Min.	Max.	Unit	Te Co
Clock cycle time	t_{cyc}	29.4	50	ns	Fig
Clock high level pulse width	t_{CH}	9.7	—		
Clock low level pulse width	t_{CL}	9.7	—		
Clock rise time	t_{Cr}	—	5		
Clock fall time	t_{Cf}	—	5		
Reset oscillation stabilization (crystal)	t_{OSC1}	10	—	ms	Fig
Software standby oscillation stabilization time (crystal)	t_{OSC2}	8	—		Fig

External clock input falling time	t_{EXf}	—	5	ns	
Clock low level pulse width	t_{CL}	0.4	0.6	t_{cyc}	Figu
Clock high level pulse width	t_{CH}	0.4	0.6	t_{cyc}	
External clock output stabilization delay time	t_{DEXT}*	500	—	µs	Figu

Note: * t_{DEXT} includes a $\overline{RES}$ pulse width (t_{RESW}).

Table 31.6 Subclock Input Conditions

Conditions: VCC = 3.0 V to 3.6 V, VSS = 0 V, ϕ = 20 MHz to 34 MHz

Item	Symbol	Min.	Typ.	Max.	Unit	Tes Con
Subclock input low level pulse width	t_{EXCLL}	—	15.26	—	µs	Figu
Subclock input high level pulse width	t_{EXCLH}	—	15.26	—	µs	
Subclock input rising time	t_{EXCLr}	—	—	10	ns	
Subclock input falling time	t_{EXCLf}	—	—	10	ns	
Clock low level pulse width	t_{CL}	0.4	—	0.6	t_{cyc}	Figu
Clock high level pulse width	t_{CH}	0.4	—	0.6	t_{cyc}	

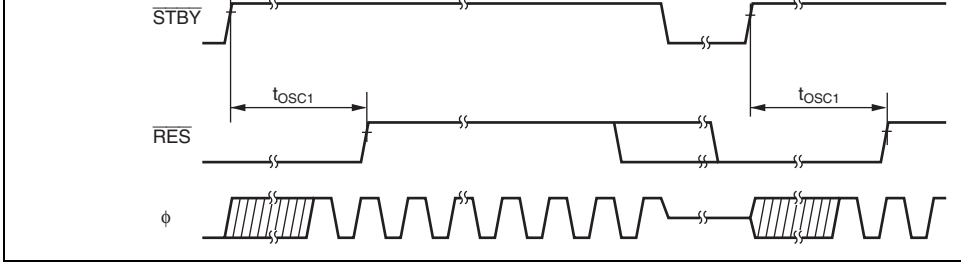


Figure 31.5 Oscillation Stabilization Timing

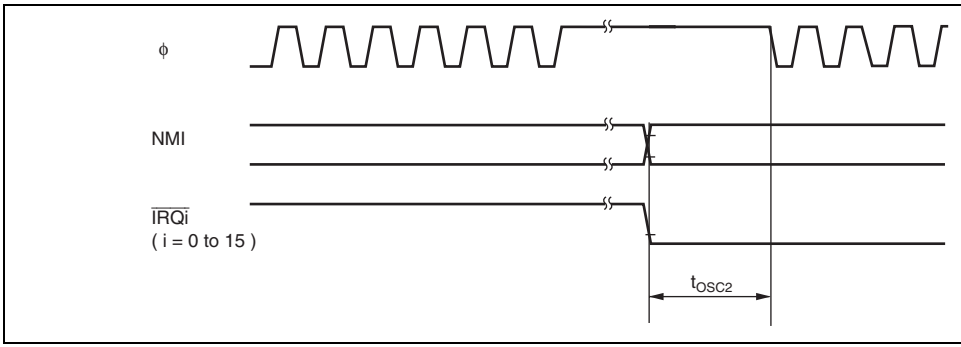
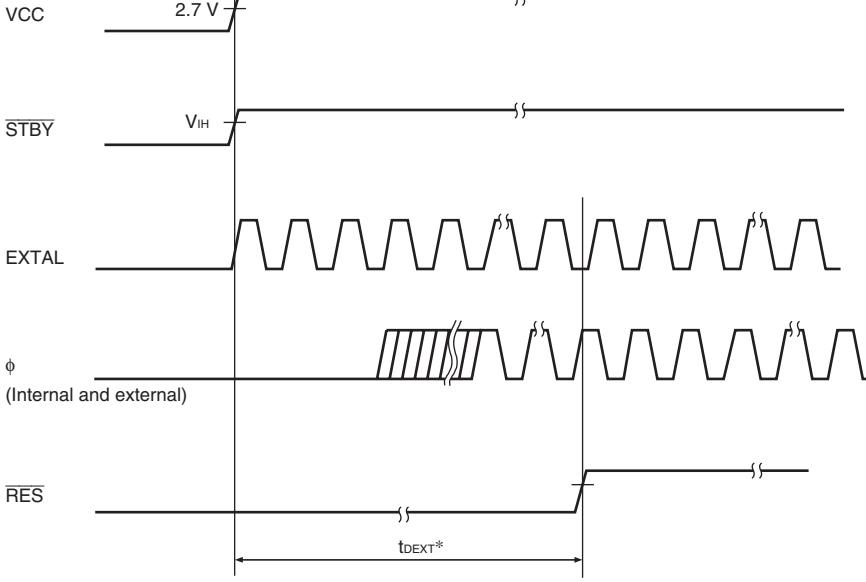


Figure 31.6 Oscillation Stabilization Timing (Exiting Software Standby Mo



Note: The external clock output stabilization delay time (t_{DEXT}) includes a $\overline{RES}$ pulse width (t_{RESW}).

Figure 31.8 Timing of External Clock Output Stabilization Delay Time

$\overline{RES}$ pulse width	t_{RESW}	20	—	t_{cyc}	
NMI setup time	t_{NMIS}	150	—	ns	Figure 31.11
NMI hold time	t_{NMIH}	10	—		
NMI pulse width (exiting software standby mode)	t_{NMIW}	200	—		
IRQ setup time ($\overline{IRQ15}$ to $\overline{IRQ0}$)	t_{IRQS}	150	—		
IRQ hold time ($\overline{IRQ15}$ to $\overline{IRQ0}$)	t_{IRQH}	10	—		
IRQ pulse width ($\overline{IRQ15}$ to $\overline{IRQ0}$) (exiting software standby mode)	t_{IRQW}	200	—		

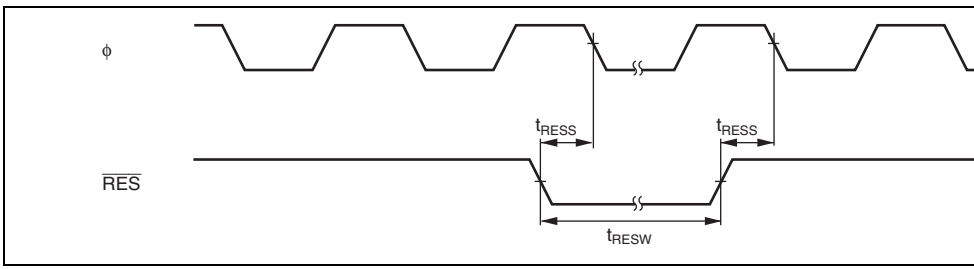


Figure 31.10 Reset Input Timing

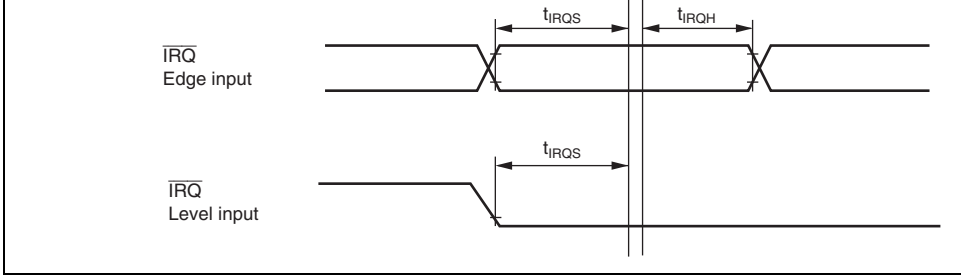
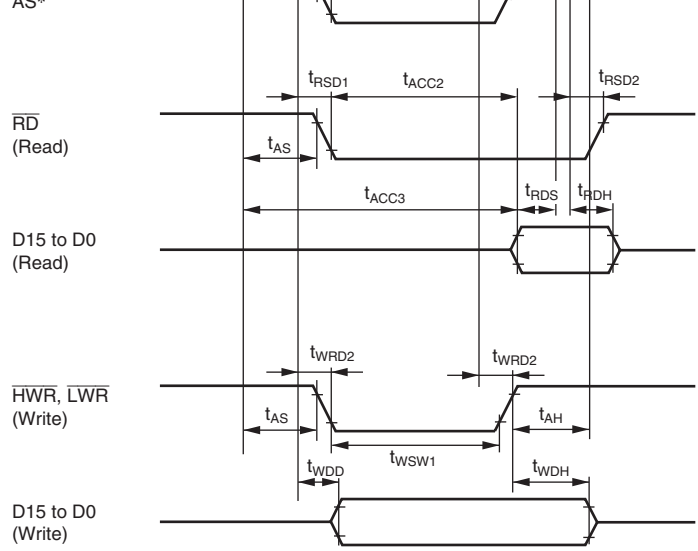


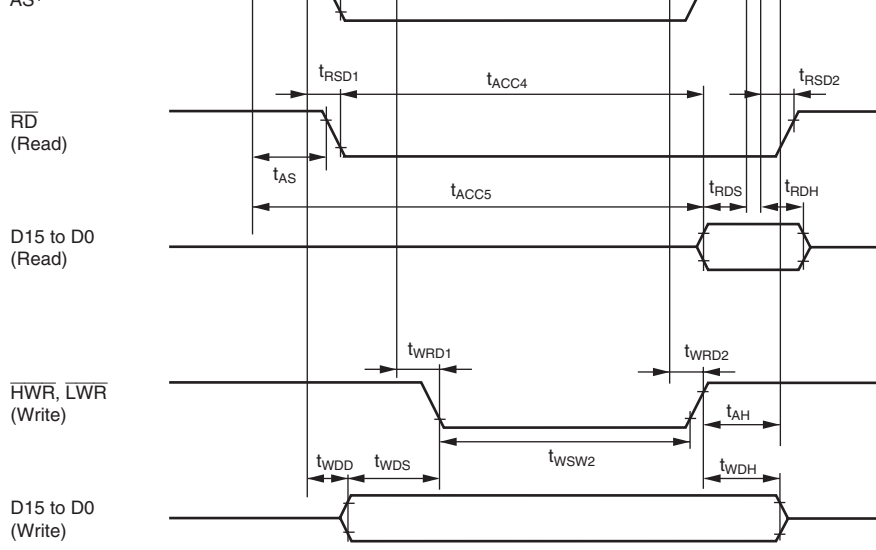
Figure 31.11 Interrupt Input Timing

Address setup time	t_{AS}	$0.5 \times t_{cyc} - 14.7$	—
Address hold time	t_{AH}	$0.5 \times t_{cyc} - 9.7$	—
$\overline{CS}$ delay time ($\overline{IOS}$, $\overline{CS256}$)	t_{CSD}	—	14.7
$\overline{AS}$ delay time	t_{ASD}	—	14.7
$\overline{HBE}$ delay time	t_{HBD}	—	$t_{AD} + 5.0$
$\overline{LBE}$ delay time	t_{LBD}	—	$t_{AD} + 5.0$
$\overline{RD}$ delay time 1	t_{RSD1}	—	14.7
$\overline{RD}$ delay time 2	t_{RSD2}	—	14.7
Read data setup time	t_{RDS}	14.7	—
Read data hold time	t_{RDH}	0	—
Read data access time 1	t_{ACC1}	—	$1.0 \times t_{cyc} - 29.4$
Read data access time 2	t_{ACC2}	—	$1.5 \times t_{cyc} - 24.7$
Read data access time 3	t_{ACC3}	—	$2.0 \times t_{cyc} - 29.4$
Read data access time 4	t_{ACC4}	—	$2.5 \times t_{cyc} - 24.7$
Read data access time 5	t_{ACC5}	—	$3.0 \times t_{cyc} - 29.4$
$\overline{WR}$ delay time 1	t_{WRD1}	—	14.7
$\overline{WR}$ delay time 2	t_{WRD2}	—	14.7
$\overline{WR}$ pulse width 1	t_{WSW1}	$1.0 \times t_{cyc} - 19.6$	—
$\overline{WR}$ pulse width 2	t_{WSW2}	$1.5 \times t_{cyc} - 19.6$	—
Write data delay time	t_{WDD}	—	24.7
Write data setup time	t_{WDS}	0	—
Write data hold time	t_{WDH}	$0.5 \times t_{cyc} - 5$	—
$\overline{WAIT}$ setup time	t_{WTS}	24.7	—
$\overline{WAIT}$ hold time	t_{WTH}	5	—



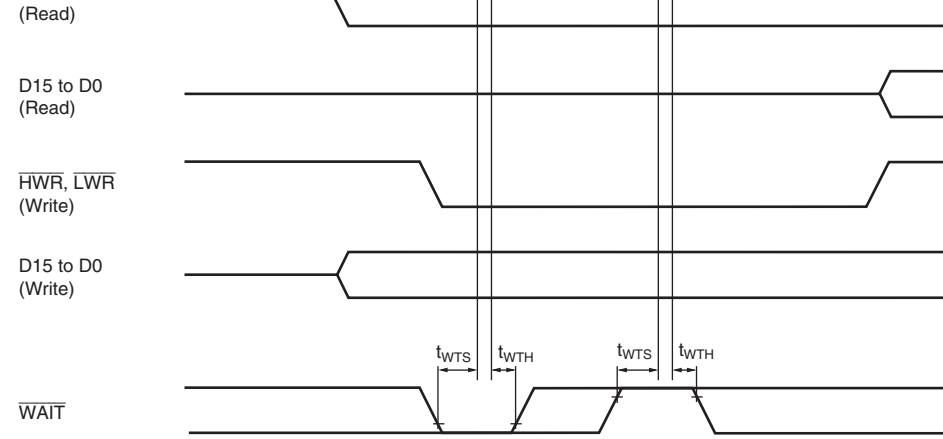
Note: * AS is multiplexed with IOS. Either the AS or IOS function can be selected by the IOSE bit of SYS

Figure 31.12 Basic Bus Timing/2-State Access



Note: * $\overline{AS}$ is multiplexed with $\overline{IOS}$. Either the $\overline{AS}$ or $\overline{IOS}$ function can be selected by the IOSE bit of SYSC

Figure 31.13 Basic Bus Timing/3-State Access



Note: * AS is multiplexed with IOS. Either the AS or IOS function can be selected by the IOSE bit of SYSCR.

Figure 31.14 Basic Bus Timing/3-State Access with One Wait State

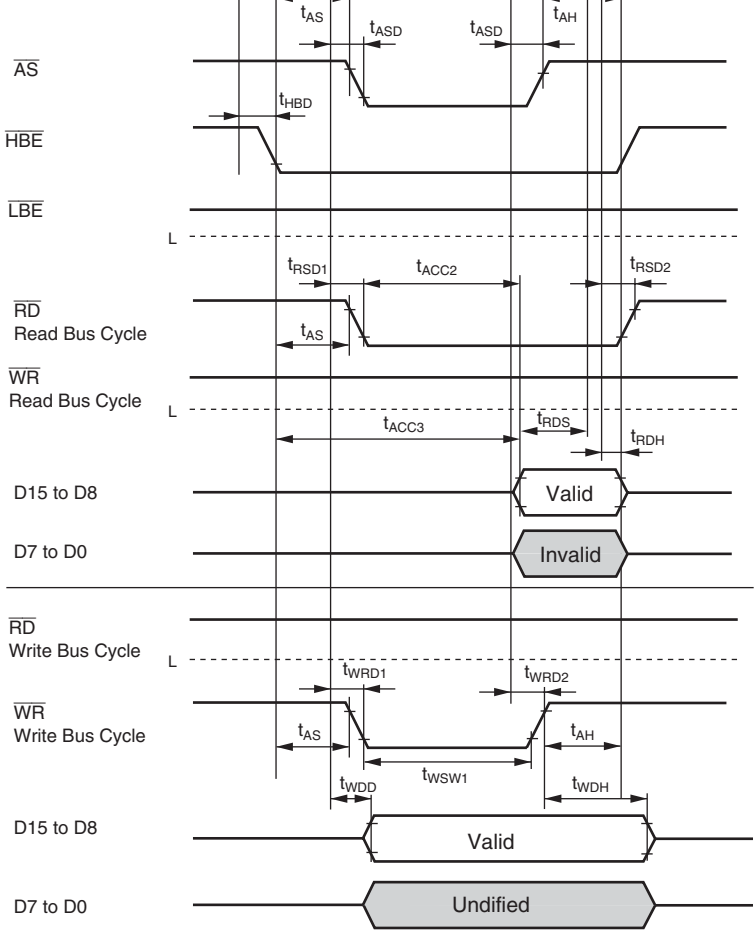


Figure 31.15 Even Byte Access (ADMXE = 0)

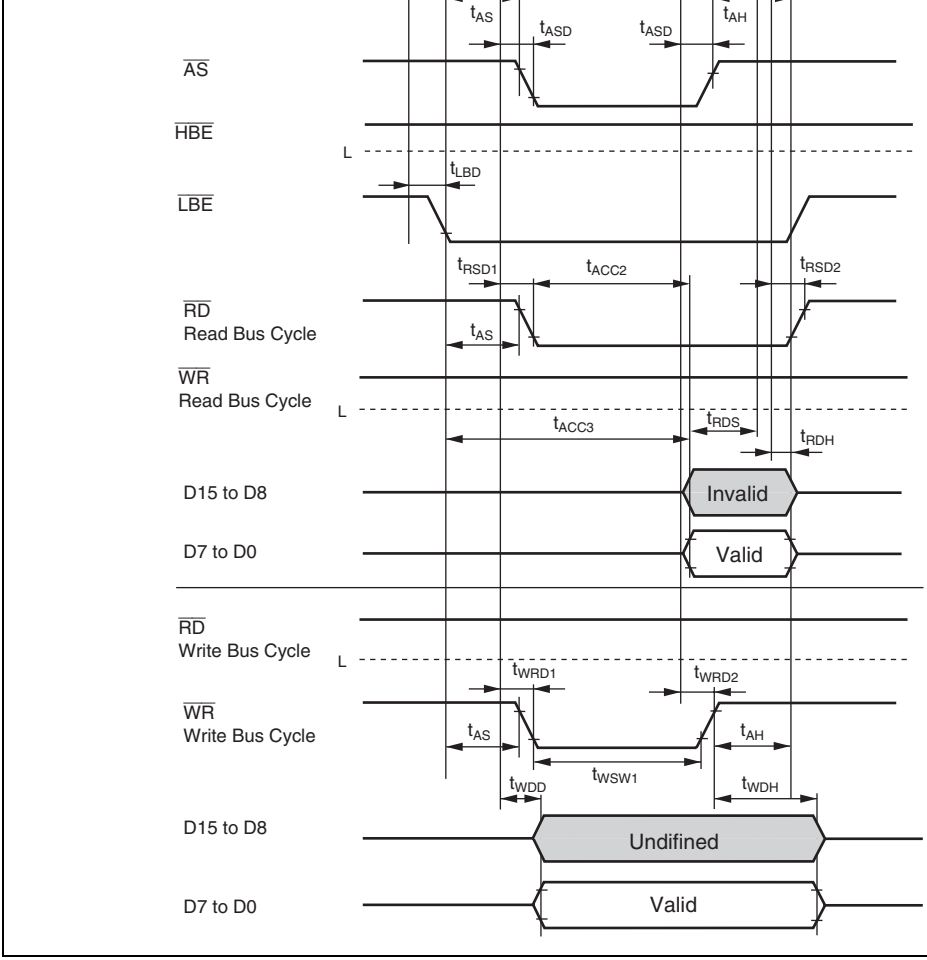


Figure 31.16 Odd Byte Access (ADMXE = 0)

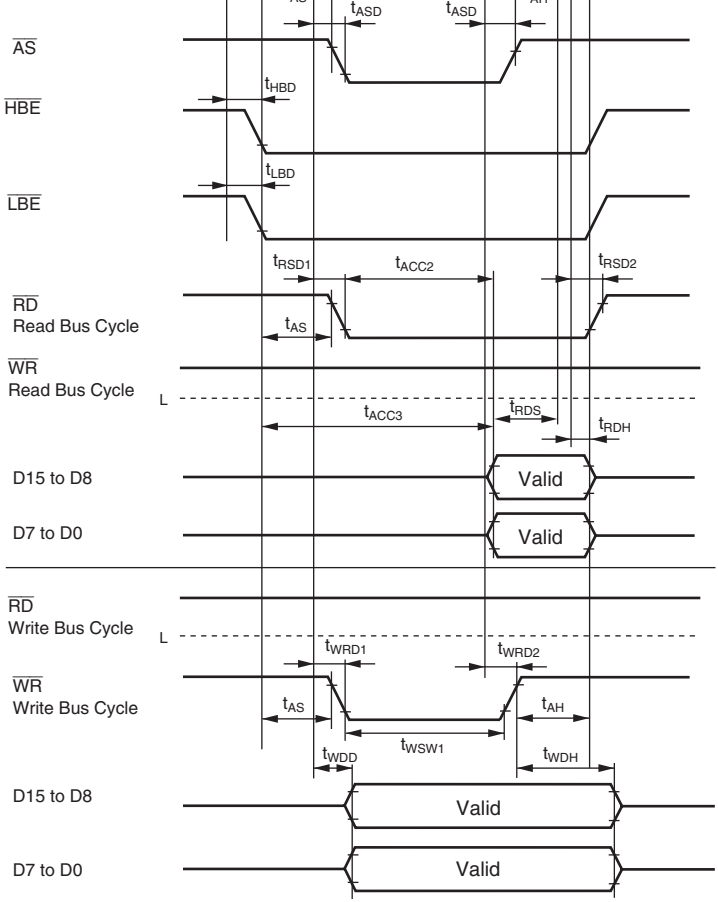
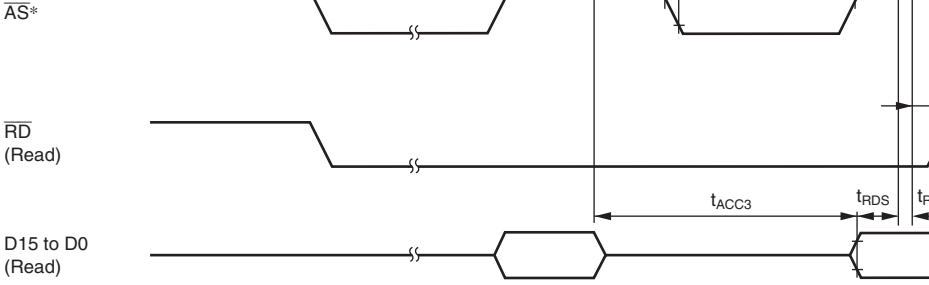
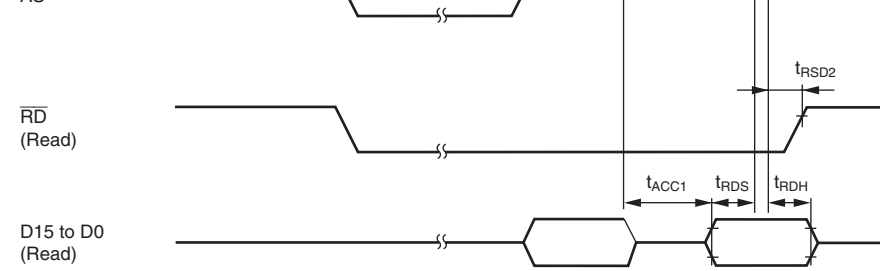


Figure 31.17 Word Access (ADMXE = 0)



Note: * AS is multiplexed with IOS. Either the AS or IOS function can be selected by the IOSE bit of SYSCR.

Figure 31.18 Burst ROM Access Timing/2-State Access



Note: * AS is multiplexed with IOS. Either the AS or IOS function can be selected by the IOSE bit of SYSC

Figure 31.19 Burst ROM Access Timing/1-State Access

Address setup time 2	t_{AS2}	$0.5 \times t_{cyc} - 14.7$	—	31.21
Address hold time 2	t_{AH2}	$0.5 \times t_{cyc} - 9.7$	—	
$\overline{CS}$ delay time ($\overline{IOS}$, $\overline{CS256}$)	t_{CSD}	—	14.7	
$\overline{AH}$ delay time	t_{AHD}	—	14.7	
$\overline{RD}$ delay time 1	t_{RSD1}	—	14.7	
$\overline{RD}$ delay time 2	t_{RSD2}	—	14.7	
Read data setup time	t_{RDS}	14.7	—	
Read data hold time	t_{RDH}	0	—	
Read data access time 2	t_{ACC2}	—	$1.5 \times t_{cyc} - 24.4$	
Read data access time 4	t_{ACC4}	—	$2.5 \times t_{cyc} - 24.4$	
Read data access time 6	t_{ACC6}	—	$3.5 \times t_{cyc} - 24.4$	
Read data access time 7	t_{ACC7}	—	$4.5 \times t_{cyc} - 24.4$	
$\overline{WR}$ delay time 1	t_{WRD1}	—	14.7	
$\overline{WR}$ delay time 2	t_{WRD2}	—	14.7	
$\overline{WR}$ pulse width time 1	t_{WSW1}	$1.0 \times t_{cyc} - 19.6$	—	
$\overline{WR}$ pulse width time 2	t_{WSW2}	$1.5 \times t_{cyc} - 19.6$	—	
Write data delay time	t_{WDD}	—	24.4	
Write data setup time	t_{WDS}	0	—	
Write data hold time	t_{WDH}	$0.5 \times t_{cyc} - 5$	—	

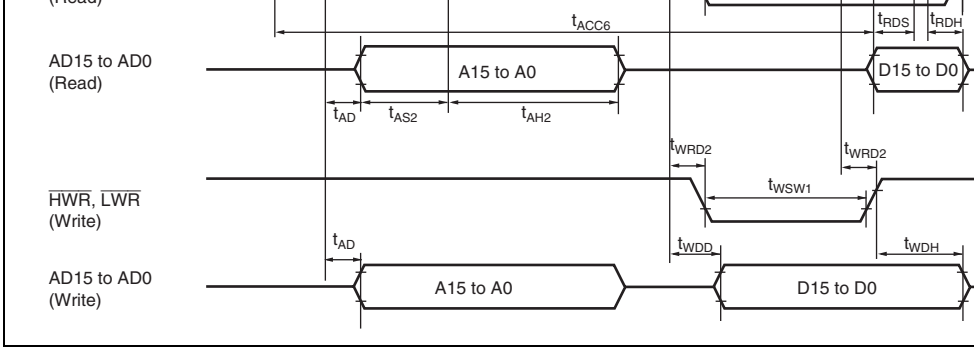


Figure 31.20 Multiplex Bus Timing/Data 2-State Access

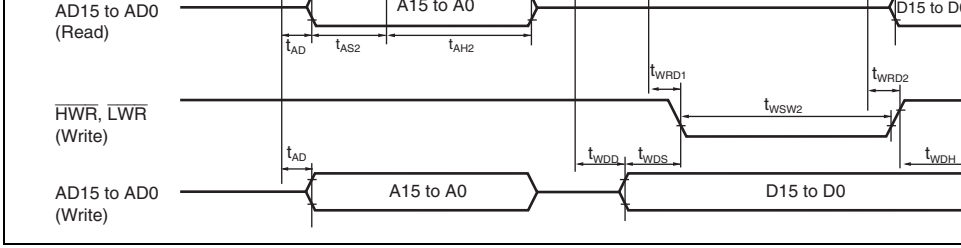


Figure 31.21 Multiplex Bus Timing/Data 3-State Access

	Synchronous		6	—		
	Input clock pulse width	t_{SCKW}	0.4	0.6	t_{Scyc}	
	Input clock rise time	t_{SCKr}	—	1.5	t_{cyc}	
	Input clock fall time	t_{SCKf}	—	1.5		
	Transmit data delay time (synchronous)	t_{TXD}	—	29.4	ns	Figure 31.
	Receive data setup time (synchronous)	t_{RXS}	19.6	—		
	Receive data hold time (synchronous)	t_{RXH}	19.6	—		
SCI (multiplexed with P51)	Receive data setup time (synchronous)	t_{RXS}	30.0	—	ns	Figure 31.
	Receive data hold time (synchronous)	t_{RXH}	30.0	—		
A/D converter	Trigger input setup time	t_{TRGS}	19.6	—		Figure 31.
WDT	$\overline{\text{RESO}}$ output delay time	t_{RESD}	—	50		Figure 31.
	$\overline{\text{RESO}}$ output pulse width	t_{RESOW}	132	—	t_{cyc}	

Note: * Only the peripheral modules that can be used in subclock operation.

Clock rising time		t_{RISE}	—	20	ns	
Clock falling time		t_{FALL}	—	20	ns	
Data input setup time	Master	t_{SU}	25	—	ns	
	Slave		30	—		
Data input hold time	Master	t_{H}	10	—	ns	
	Slave		10	—		
$\overline{SCS}$ setup time	Master	t_{LEAD}	2.5	—	t_{cyc}	
	Slave		2.5	—		
$\overline{SCS}$ hold time	Master	t_{LAG}	2.5	—	t_{cyc}	
	Slave		2.5	—		
Data output delay time	Master	t_{OD}	—	40	ns	
	Slave		—	40		
Data output hold time	Master	t_{OH}	30	—	ns	
	Slave		30	—		
Consecutive transmit delay time	Master	t_{TD}	2.5	—	t_{cyc}	
	Slave		2.5	—		
Slave access time		t_{SA}	—	1	t_{cyc}	Figure 3
Slave out release time		t_{REL}	—	1	t_{cyc}	Figure 3

Figure 31.22 I/O Port Input/Output Timing

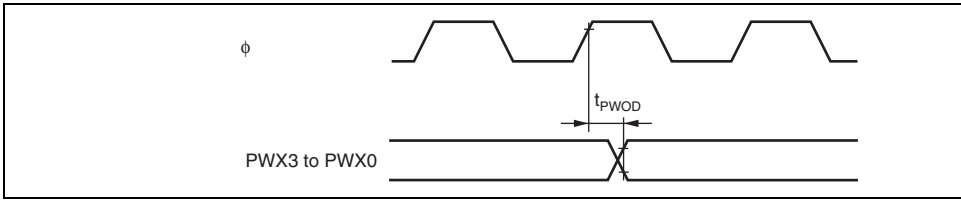


Figure 31.23 PWMX Output Timing

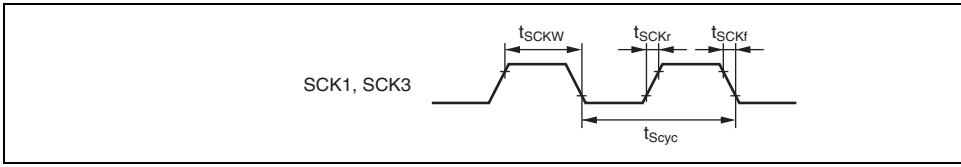


Figure 31.24 SCK Clock Input Timing

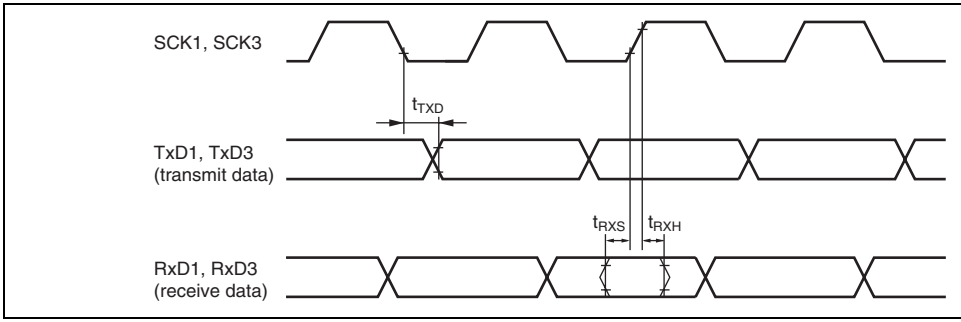


Figure 31.25 SCI Input/Output Timing (Clock Synchronous Mode)

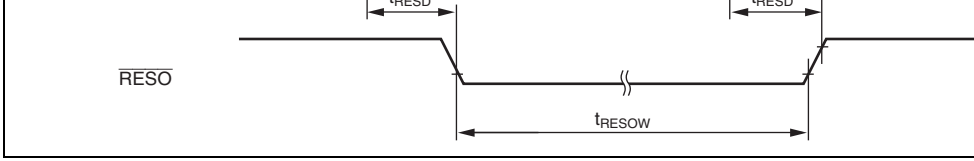


Figure 31.27 WDT Output Timing ($\overline{\text{RESO}}$)

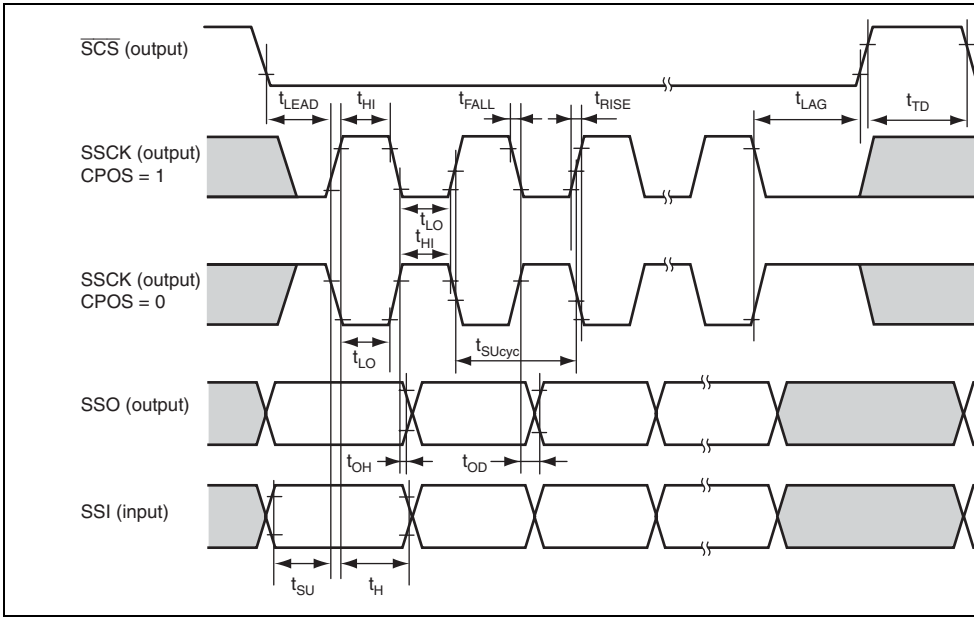


Figure 31.28 SSU Timing (Master, CPHS = 1)

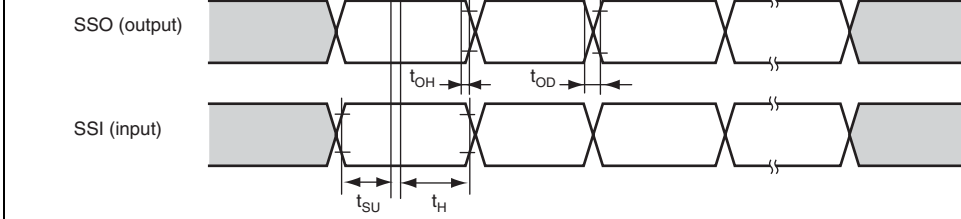


Figure 31.29 SSU Timing (Master, CPHS = 0)

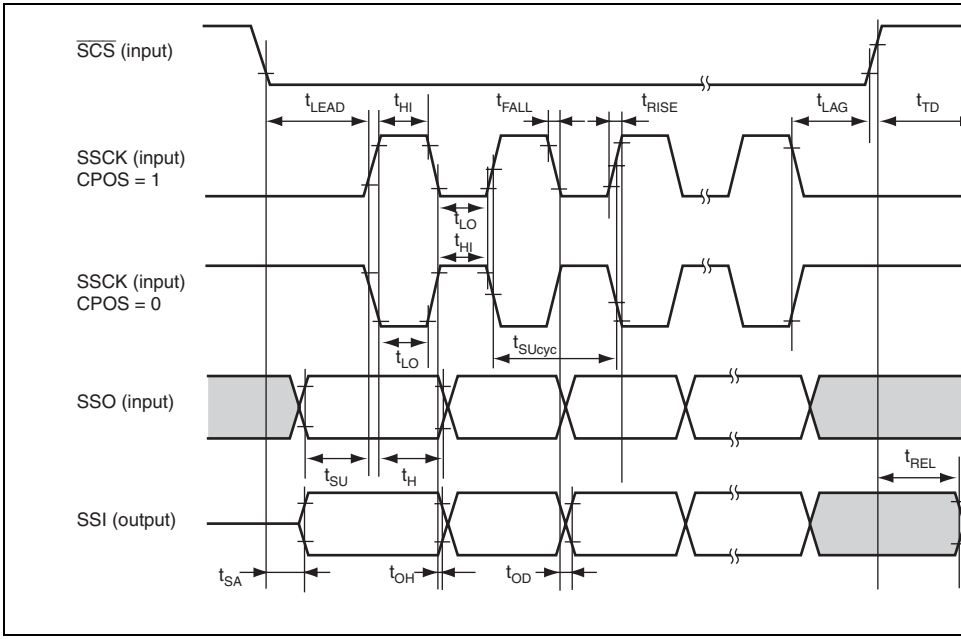


Figure 31.30 SSU Timing (Slave, CPHS = 1)

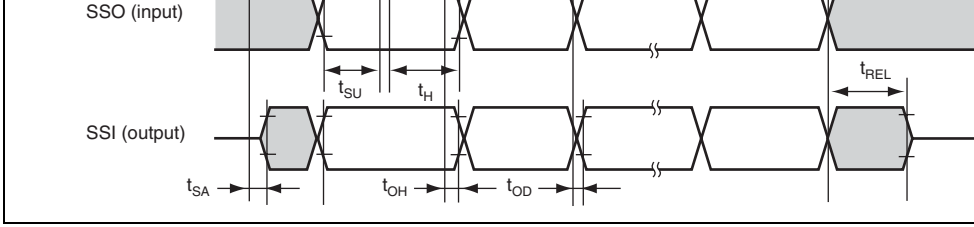
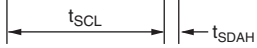


Figure 31.31 SSU Timing (Slave, CPHS = 0)

SCL, SDA output fall time	t_{Of}	$20 + 0.1\ C_b$	—	250	
SCL, SDA input spike pulse elimination time	t_{SP}	—	—	1	t_{cyc}
SDA input bus free time	t_{BUF}	5	—	—	
Start condition input hold time	t_{STAH}	3	—	—	
Repeated start condition input setup time	t_{STAS}	3	—	—	
Stop condition input setup time	t_{STOS}	3	—	—	
Data input setup time	t_{SDAS}	0.5	—	—	
Data input hold time	t_{SDAH}	0	—	—	ns
SCL, SDA capacitive load	C_b	—	—	400	pF

Note: * 17.5 t_{cyc} or 37.5 t_{cyc} can be set according to the clock selected for use by the IIC



Note: * S, P, and Sr indicate the following conditions:
S: Start condition
P: Stop condition
Sr: Repeated start condition

Figure 31.32 I²C Bus Interface Input/Output Timing

Table 31.13 LPC Module Timing

Conditions: VCC = 3.0 V to 3.6V, VSS = 0 V, ϕ = 20 MHz to 34 MHz

Item	Symbol	Min.	Typ.	Max.	Unit	Test Conditions
Input clock cycle	t_{Lcyc}	30	—	—	ns	Figure 31.33
Input clock pulse width (H)	t_{LCKH}	11	—	—		
Input clock pulse width (L)	t_{LCKL}	11	—	—		
Transmit signal delay time	t_{TXD}	2	—	11		
Transmit signal floating delay time	t_{OFF}	—	—	28		
Receive signal setup time	t_{RXS}	7	—	—		
Receive signal hold time	t_{RXH}	0	—	—		

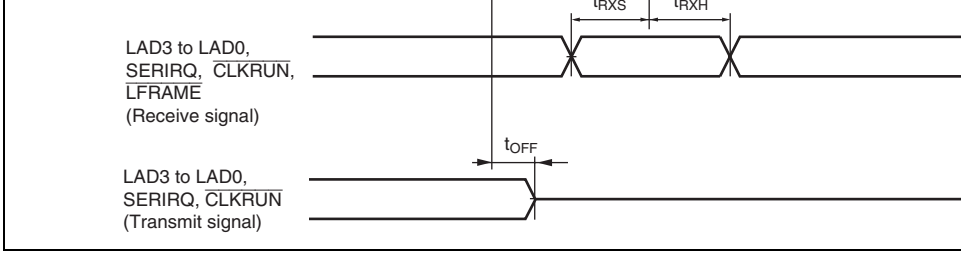


Figure 31.33 LPC Interface (LPC) Timing

RM_xxxx*[1] setup time	T_{su}	3	—	—	
RM_xxxx*[1] hold time	T_{hd}	1	—	—	
RM_xxxx*[1] rise/fall time	T_r/T_f	0.5	—	6	
MDIO setup time	t_{MDIOS}	10	—	—	Figure 3
MDIO hold time	t_{MDIOH}	10	—	—	
MDIO output data hold time*[2]	t_{MDIODH}	5	—	18	Figure 3
WOL output delay time	t_{WOLD}	1	—	20	Figure 3

Notes: 1. RM_TXD-EN, RM_TXD1, RM_TXD0, RM_CRS-DV, RM_RXD1, RM_RXD0, a RM_RX-ER

2. This specification must be satisfied by the user by software.

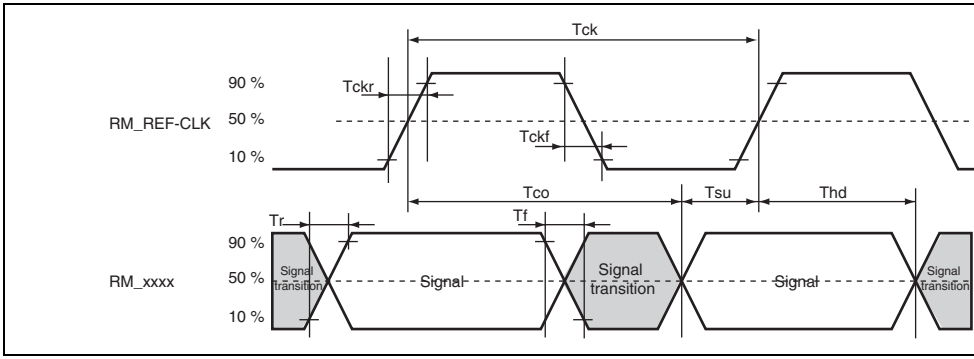


Figure 31.34 Timing of RM_REF-CLK and RMII Signals

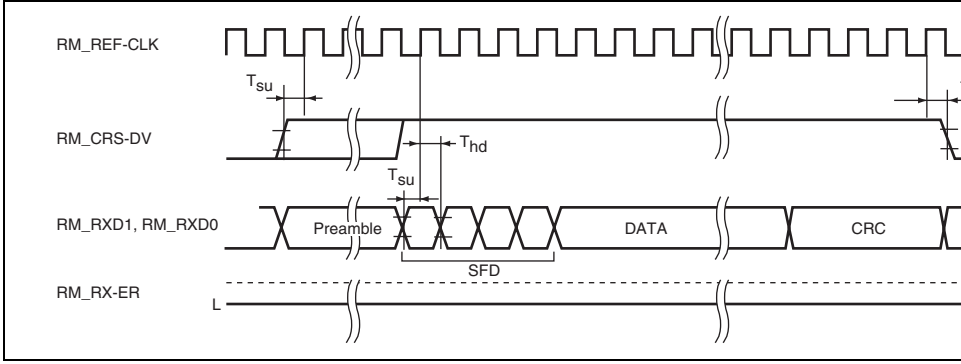


Figure 31.36 RMII Receive Timing (Normal Operation)

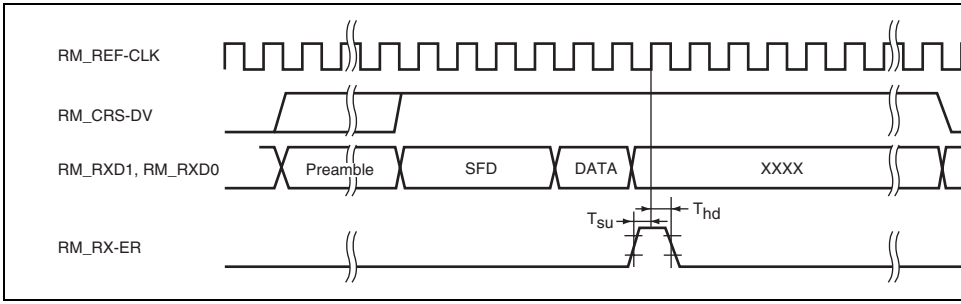


Figure 31.37 RMII Receive Timing (When an Error is Detected)

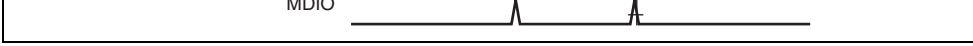


Figure 31.39 MDIO Output Timing

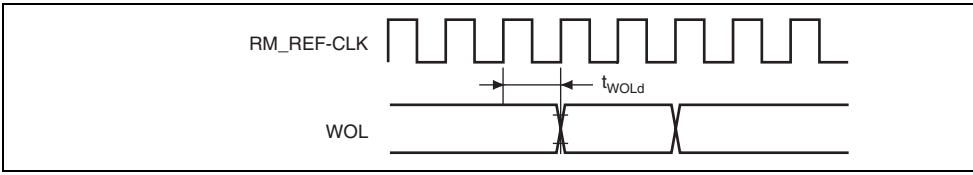


Figure 31.40 WOL Output Timing

	Differential common mode range	V_{CM}	0.8	2.5	V	
Output	Output high voltage	V_{OH}	2.8	—	V	I_{OH} = –200 µA
	Output low voltage	V_{OL}	—	0.3	V	I_{OL} = 2 mA
	Crossover voltage	V_{CRS}	1.3	2.0	V	
	Rising time	t_R	4	20	ns	
	Falling time	t_F	4	20	ns	
	Ratio of rising time to falling time	t_{RFM}	90	111.11	%	(T_R/T_F)
	Output resistance	Z_{DRV}	28	44	Ω	Including R_S = 22Ω

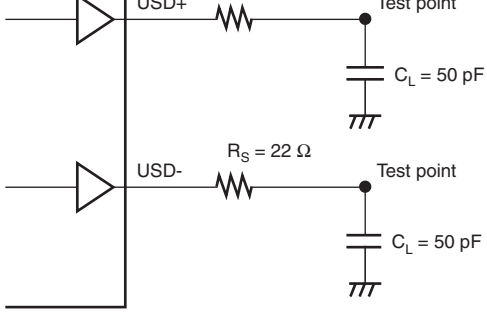


Figure 31.42 Load Condition

ETRST pulse width	t_{TRSTW}	20	—	t_{cyc}	Figure 31.4
Reset hold transition pulse width	t_{RSTHW}	3	—		
ETMS setup time	t_{TMSS}	20	—	ns	Figure 31.4
ETMS hold time	t_{TMSH}	20	—		
ETDI setup time	t_{TDIS}	20	—		
ETDI hold time	t_{TDIH}	20	—		
ETDO data delay time	t_{TDOD}	—	20		

Note: * When $t_{cyc} \leq t_{TCKcyc}$

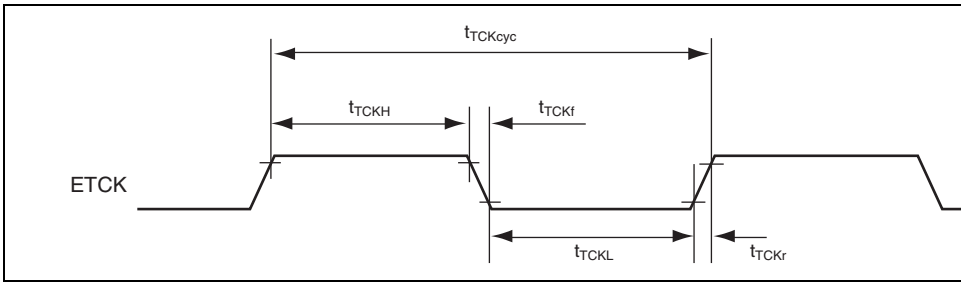


Figure 31.43 JTAG ETCK Timing

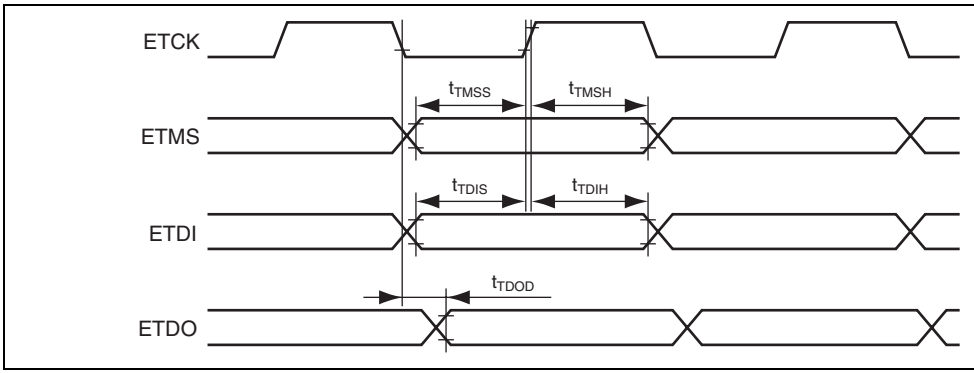


Figure 31.45 JTAG Input/Output Timing

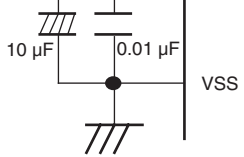
VSS = AVSS = 0 V, φ = 20 MHz to 34 MHz

	Condition A			Condition B			
Item	**Min.**	**Typ.**	**Max.**	**Min.**	**Typ.**	**Max.**	**Unit**
Resolution	10	10	10	10	10	10	Bits
Conversion time	—	—	4.0*[1]	—	—	4.7*[2]	µs
Analog input capacitance	—	—	20	—	—	20	pF
Permissible signal-source impedance	—	—	5	—	—	5	kΩ
Nonlinearity error	—	—	±7.0	—	—	±7.0	LSB
Offset error	—	—	±7.5	—	—	±7.5	
Full-scale error	—	—	±7.5	—	—	±7.5	
Quantization error	—	—	±0.5	—	—	±0.5	
Absolute accuracy	—	—	±8.0	—	—	±8.0	

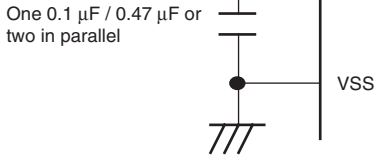
Notes: 1. Value when using the maximum operating frequency in single mode of 80 sta
2. Value when using the maximum operating frequency in single mode of 160 st

Item	Symbol	Min.	Typ.	Max.	Unit	Test Condi
Programming time*1*2*4	t_P	—	1	10	ms/128 bytes	
Erase time*1*2*4	t_E	—	40	130	ms/4-kbyte block	
		—	300	800	ms/32-kbyte block	
		—	600	1500	ms/64-kbyte block	
Programming time (total)*1*2*4	Σt_P	—	9.2	24	s/512 kbytes	Ta =
Erase time (total)*1*2*4	Σt_E	—	9.2	24	—	—
Programming and Erase time (total)*1*2*4	Σt_{PE}	—	18.4	48	—	—
Reprogramming count*5	N_{WEC}	100*3	1000	—	Times	
Data retention time*4	t_{DRP}	10	—	—	Years	

Notes: 1. Programming and erase time depends on the data.
2. Programming and erase time do not include data transfer time.
3. This value indicates the minimum number of which the flash memory are reprogrammed with all characteristics guaranteed. (The guaranteed value rang 1 to the minimum number.)
4. This value indicates the characteristics while the flash memory is reprogramme the specified range (including the minimum number).
5. Reprogramming count in each erase block.



It is recommended that a bypass capacitor be connected to the VCC pin. (The values are reference values.)
When connecting, place a bypass capacitor near the pin.



Do not connect Vcc power supply to the VCL
Always connect a capacitor for internal step-
stabilization.
Use one or two ceramic multilayer capacitor
(0.1 µF / 0.47 µF: connect in parallel when u
and place it (them) near the pin.

Figure 31.46 Connecting Capacitors to VCC and VCL Pins

A7 to A0	1 (DDR=1)			kept*	kept*	Address
Port 27 to 24	X	T	T	kept	kept	I/O port
Port 23 to 20	0 / 1 (DDR=0)	T	T	kept	kept	I/O port
A11 to A8	1 (DDR=1)			kept*	kept*	Address
Port 3	0	T	T	kept	kept	I/O port
D15 to D8	1			T	T	D15 to D
Port 47 to 44	0	T	T	kept	kept	I/O port
A15 to A12	1			kept*	kept*	A15 to A
Port 43 to 40	0 / 1 (8 bits)	T	T	kept	kept	I/O port
D7 to D4	1 (16 bits)			T	T	D7 to D4
Port 57	0	T	T	kept	kept	I/O port
$\overline{WR}$, $\overline{HWR}$	1			H	H	$\overline{WR}$, $\overline{HW}$
Port 56	0	T	T	T	T	Input por
EXCL	1 (DDR=0)					EXCL
ϕ	1 (DDR=1)			H	ϕ output	ϕ
Port 55 to 50	X	T	T	kept	kept	I/O port
Port 67 to 64	X	T	T	kept	kept	I/O port
Port 63 to 60	0 / 1 (8 bits)	T	T	kept	kept	I/O port
D3 to D0	1 (16 bits)			T	T	D3 to D0
Port 7	X	T	T	T	T	Input por
Port 8	X	T	T	kept	kept	I/O port

Port 92	0	T	T	kept	kept	I/O port
$\overline{HBE}$	1			H	H	$\overline{HBE}$
Port 91	0 / 1 (ADMXE=0)	T	T	kept	kept	I/O port
$\overline{AH}$	1 (ADMXE=1)			H	H	$\overline{AH}$
Port 90	0	T	T	kept	kept	I/O port
$\overline{LBE}$	1			H	H	$\overline{LBE}$
Port A7 to A2	0 / 1 (address 18=1)	T	T	kept	kept	I/O port
A23 to A18	1 (address 18=0)			kept*	kept*	A23 to A1
Port A1, A0	0 / 1 (address 13=1)	T	T	kept	kept	I/O port
A17, A16	1 (address 13=0)			kept*	kept*	A17, A16
Port B	X	T	T	kept	kept	I/O port
Port C7	0	T	T	kept	kept	I/O port
$\overline{RD}$	1			H	H	$\overline{RD}$
Port C6	0 / 1 (8 bits)	T	T	kept	kept	I/O port
$\overline{LWR}$	1 (16 bits)			H	H	$\overline{LWR}$
Port C5 to C0	X	T	T	kept	kept	I/O port

kept: Input port pins are in the high-impedance state (when DDR = 0 and PCR = 1, the up MOS remains on).

Output port pins retain their states.

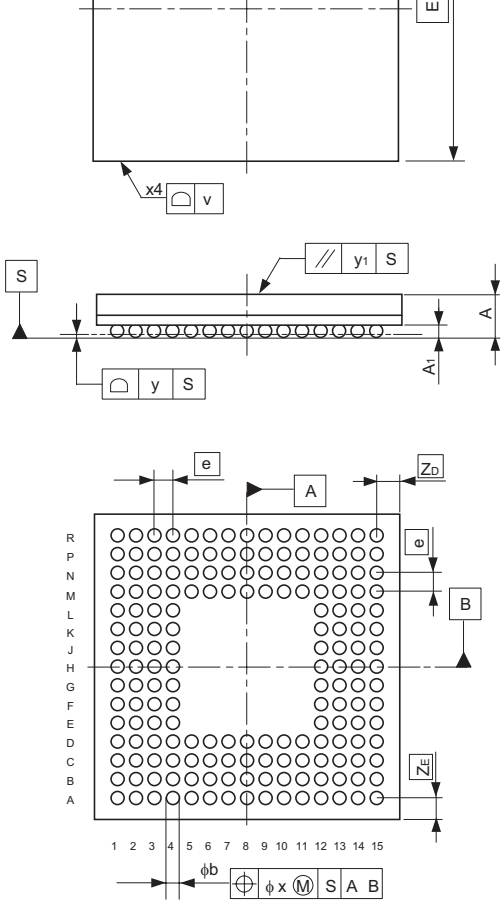
Functions of some pins will be changed to the I/O port function, which is determin DDR and DR, because the on-chip peripheral module associated with that pin fu initialized.

DDR: Data direction register

x: Don't care

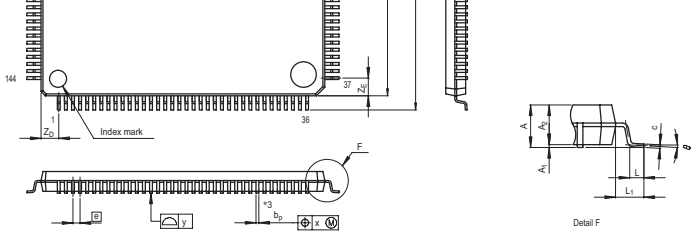
Note: * In the case of address output, the last address accessed is retained.

	(wide temperature specifications)			
H8S/2462	F-ZTAT version (regular specifications)	R4F2462	F2462VFQ34V	144-pin LQF (PLQP0144K
	F-ZTAT version (wide temperature specifications)	R4F2462	F2462VFQ34DV	



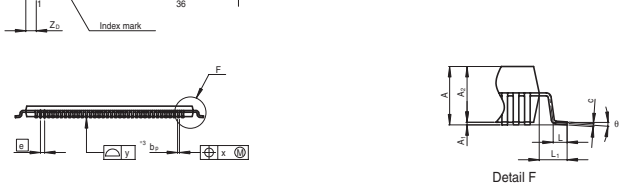
Reference Symbol	Dimension in M	
	Min	Nom
D	—	13.0
E	—	13.0
v	—	—
w	—	—
A	—	—
A_1	0.35	0.40
e	—	0.80
b	0.45	0.50
x	—	—
y	—	—
y_1	—	—
S_D	—	—
S_E	—	—
Z_D	—	0.90
Z_E	—	0.90

Figure C.1 Package Dimensions (PLBGA0176GA-A)



E	19.
A_2	—
H_D	21.
H_E	21.
A	—
A_1	0.0
b_p	0.1
b_1	—
c	0.0
c_1	—
θ	0
e	—
x	—
y	—
Z_D	—
Z_E	—
L	0.3
L_1	—

Figure C.2 Package Dimensions (PLQP0144KA-A)



E	–
A_2	–
H_D	17
H_E	17
A	–
A_1	0.
b_p	0.
b_1	–
c	0.
c_1	–
θ	
e	–
x	–
y	–
Z_D	–
Z_E	–
L	0
L_1	–

Figure C.3 Package Dimensions (PTQP0144LC-A)

Clock pulse generator

Notes:

1. Supported only by the H8S/2472 Group.
2. Supported only by the H8S/2472 Group and the H8 Group.

2 Added

- On-chip memory

ROM Type	Model	ROM	RAM	Remarks
Flash memory version	R4F2472	512 Kbytes	40 Kbytes	176 pins, USB incorpo
	R4F2463	512 Kbytes	40 Kbytes	144 pins, USB and PE incorporated
	R4F2462	512 Kbytes	40 Kbytes	144 pins, USB not inco

- Compact package

Package (code)	Body Size	Pin Pitch
PLBG0176GA-A	13 × 13 mm	0.8 mm
PTQP0144LC-A	16 × 16 mm	0.4 mm
PLQP0144KA-A	20 × 20 mm	0.5 mm

DTC: Data transfer controller
EVC: Event counter
SCI: Serial communication interface
SCIF: Serial communication interface with FIFO
IIC: I²C bus interface
EtherC: Ethernet controller
E-DMAC: Direct memory access controller for Ethernet controller
SSU: Synchronous serial communication unit
USB: USB function module
FRT: 16-bit free running timer
PWM: 14-bit PWM timer
LPC: LPC interface
WDT: Watchdog timer
JTAG: Boundary scan
PECI: PECI interface

Notes: 1. Supported only by the H8S/2472 Group.
2. Supported only by the H8S/2472 Group and the H8S/2462 Group.

Figure 1.2 Pin Assignments (H8S/2472 Group)	4	Replaced

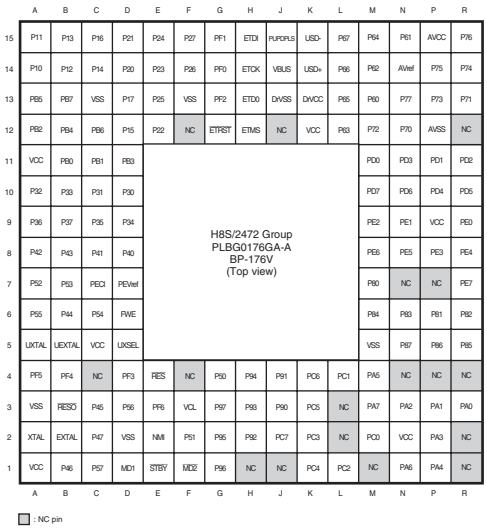


Figure 1.3 Pin Assignments (H8S/2463 Group)	5	Added

Table 1.1 Pin Assignments in Each Operating Mode	7,13	H8S/2463 added Pin No.

H8S/2462 (FP-144LV)	H8S/2463 (TFP-144V)	H8S/2472 (BP-176V)
1	1	A1
144	144	B2

	7	Added

Pin No.	Pin Name
H8S/2462 (FP-144LV)	Extended Mode (EXPE=1)
17	P97/CS256/WAIT

Table 1.2 Pin Functions	14, 18	H8S/2463 added

		Pin No.		
Type	**Symbol**	**H8S/2462 144-Pin**	**H8S/2463 144-Pin**	**H**
Power supply	VCC	1, 36, 86, 120	1, 36, 86, 120	A F A
Synchronous serial communication unit (SSU)	SSI	139	137	A
	$\overline{SCS}$	84	83	L

		(BCR2, WSCR2, PCSR, SYSCR2). 0: Area from H'FFFE88 to H'FFFE8 reserved. Area from H'FFFEA0 to H'FFFEE allocated to registers of AD, seria multiplexed functions, and I/O po Area from H'FFFF80 to H'FFFF8 allocated to control registers of p down states and on-chip periphe modules. 1: Area from H'FFFE88 to H'FFFE8 allocated to control registers of fl memory. Area from H'FFFEA0 to H'FFFEE reserved. Area from H'FFFF80 to H'FFFF8 reserved.
Section 4 Exception Handling 4.3.3 On-Chip Peripheral Modules after Reset is Cancelled	75	After a reset is cancelled, the module stop control regis (MSTPCR, MSTPCRA, and SUBMSTPB~~, and SUBMS~~ initialized, …

—: Reserved. The write value should always be 0

Notes: 1. Supported only by the H8S/2472 Group.

2. Supported only by the H8S/2472 Group and H8S/2462 Group.

Table 5.3 Interrupt Sources, Vector Addresses, and Interrupt Priorities	92	Amended

Origin of Interrupt Source
PECI*[2]
USB*[1]

Notes: 1. Supported only by the H8S/2472 Group.

2. Supported only by the H8S/2472 Group and H8S/2462 Group.

Section 8 I/O Ports	189	Description amended
8.1 I/O Ports for the H8S/2472 Group		Table 8.1 is a summary of the port functions. … For p and D0 to D5 pins, the on/off status of the input pull-u controlled by their respective DDR and the output data (ODR). Ports 1 to 4, and 6 have an input pull-up MOS register (PCR), …
8.1.9 Port 9	237	Deleted
(3) Pin Functions		• P97/$\overline{\text{WAIT}}$/$\overline{\text{CS256}}$

The pin function is switched as shown below acco the operating mode and the combination of the CS in SYSCR, the WMS1 bit in WSCR~~, the WMS21 b WSCR2,~~ and the P97DDR bit.

Operating Mode	Extended mode	
WMS1, ~~WMS21~~	~~All~~ 0	~~Either b~~

When the module stop mode is cleared in both the Ethe
E-DMAC, this pin functions as the WOL output pin.

8.1.15 Port F — 271

Amended

- PF1/RS9/MDC, PF0/RS8/MDIO

EtherC, E-DMAC	Ether of them is stopped		Both of them are s
PFnDDR	0	1	X
Pin function	PFn input pin	PFn output pin	MDC output pin/ MDIO input/output

8.2 I/O Ports for the H8S/2463 Group and the H8S/2462 Group — 272

Title and description amended

Table 8.9 is a summary of the port functions. … For po
and D0 to D5 pins, the on/off status of the input pull-up
controlled by their respective DDR and the output data
(ODR). Ports 1 to 4, and 6 have an input pull-up MOS c
register (PCR), …

8.2.9 Port 9 — 321

(3) Pin Functions

Deleted

- P97/$\overline{\text{WAIT}}$/$\overline{\text{CS256}}$

  The pin function is switched as shown below accord
  the operating mode and the combination of the CS2
  in SYSCR, the WMS1 bit in WSCR~~, the WMS21 bit
  WSCR2,~~ and the P97DDR bit.

Operating Mode	Extended mode	
WMS1, ~~WMS21~~	~~All~~ 0	~~Either bit~~

		when using the pin as the A21 or PA5 output pin. When the module stop mode is cleared in both the Et E-DMAC, this pin functions as the WOL output pin.
8.2.15 Port F	355	Amended • PF1/RS9/MDC

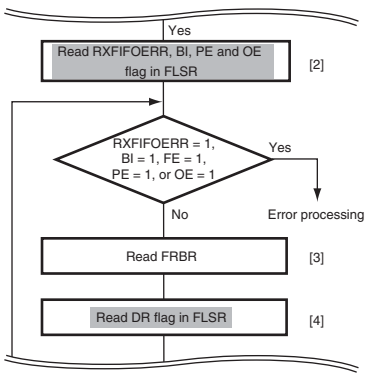
EtherC, E-DMAC	Either of them is stopped		Both of th stopped
PF1DDR	0	1	X
Pin function	PF1 input pin	PF1 output pin	MDC ou

Section 10 16-Bit Free-Running Timer (FRT) Figure 10.3 Timing of Output Compare A Output	383	Replaced
Section 12 Watchdog Timer (WDT) 12.3.2 Timer Control/ Status Register (TCSR) • TCSR_0	418	Amended

Bit	Bit Name	Initial Value	R/W
4	—	0	R/W

Communication Interface with FIFO (SCIF) 15.3 Register Descriptions		• Sub-chip module stop control register BL (SUBMS
Table 15.2 Register Access	510	Description in this table amended

SCIFE Bit in HICR5	0	
Bit 3 in SUBMSTPBL	0	1

Figure 15.4 Example of Data Transmission Flowchart	531	Modified [1] Confirm that the THRE flag in FLSR is 1, and write transmi FTHR. When FIFOs are used, write 1-byte to 16-byte trans When the OUT2 bit in FMCR and the ETBEI bit in FIER ar an FTHR empty interrupt occurs. When data is written to F transferred automatically to FTSR. The data is then transmitted from the TxDF pin in the orde start bit, transmit data, parity bit, and stop bit.
Figure 15.5 Example of DATA Reception Flowchart	532	Modified 

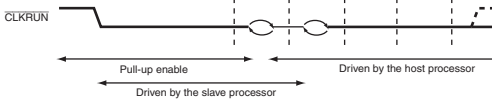
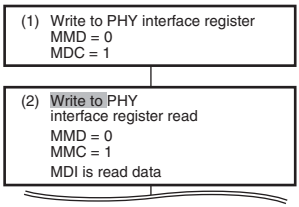


Table 19.14 Host Address Example	755	Amended **Register** **Host Address when LADR3 = H'A24F** **Host Address when LADR3 = H**
Section 20 Ethernet Controller (EtherC) Table 20.1 Pin Configuration	759	Amended and added **Type** **Abbreviation** **I/O** **Function** PHY register interface signals — MDC — Output — Management Data Cloc Reference clock signal information transfer via Others — EXOUT — Output — External Output
Figure 20.10 1-Bit Data Read Flowchart	784	Modified



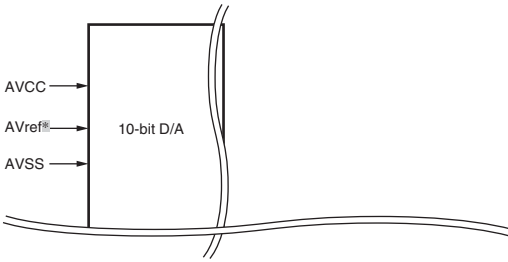
this LSI is reset, all bits are cleared to 0.

Bit	Bit Name	Initial value	R/W	Description
31 to 8	—	All 0	R	Reserved These bits are always read as 0 initial value should not be chang
7	RMAFCE	0	R/W	RMAF Bit Copy Directive 0: Indicates the RMAF bit state RFE of the receive descripto 1: Occurrence of the correspon interrupt is not indicated in b the receive descriptor
6, 5	—	All 0	R	Reserved These bits are always read as 0 initial value should not be chang
4	RRFCE	0	R/W	RRF Bit Copy Directive 0: Indicates the RRF bit state i of the receive descriptor 1: Occurrence of the correspon interrupt is not indicated in b the receive descriptor
3 to 0	—	All 0	R	Reserved These bits are always read as 0 initial value should not be chang

				~~transmit/receive status enable register.~~ 0: No error during trans 1: An error occurred du transmission
26 to 0	TFS26 to TFS0	All 0	R/W	Transmit Frame Status TFS26 to TFS9: Reserv write value shou be 0.) TFS8: Transmit Abort D (indicates any o TFS3 to TFS0 h set.) TFS7 to TFS4: Reserve write value sho be 0.) TFS3: Carrier Not Dete (corresponds to in EESR) TFS2: Detect Loss of C (corresponds to EESR) TFS1: Delayed Collisio (corresponds to EESR) TFS0: Transmit Retry C (corresponds to in EESR)

set.)

…

22.10.7 Restrictions on Peripheral Module Clock (ϕ) Operating Frequency	892	Modified When UXSEL is set to 0, connect U~~S~~EXTAL to the sys power supply (0 V).
Section 23 A/D Converter Figure 23.1 Block Diagram of the A/D Converter	894	Added  Note: * Supported only by the H8S/2472 Group and the H8S/2462 Gro
Table 23.1 Pin Configuration	895	Added **Pin Name** / **Function** Reference power supply pin / Reference voltage for A/D conve Supported only by the H8S/247 Group and the H8S/2462 Group

Interface Specification for Boot Mode (4) Inquiry and Selection States (b) Device Selection		This is fixed at 4
(9) Programming/ Erasing State (b) 128-byte programming	1006	Amended • Programming Address (4 bytes): Start address fo programming … (i.e. H'00, H'01, H'00, H'00 : H'00010000)

		the ETCK pin. The ETMS pin controls th internal state of the TAP controller. If th input, the ETMS pin is fixed to 1 by an i pull
	Test data input	Serial data inputPerforms serial input of instructions and data for JTAG registers sampled on the rise of the ETCK pin. If no input, the ETDI pin is fixed to 1 by an pull
	Test data output	Serial data outputPerforms serial output instructions and data from JTAG registe Transfer is performed in synchronizatio ETCK pin. If there is no output, the ETD goes to the high
	Test reset	Test reset input signalInitializes the JTA asynchronously. If there is no input, the pin is fixed to 1 by an internal pull

Table 26.3 Correspondence between Pins and Boundary Scan Register (H8S/2472 Group) — 1025, 1029

Amended

Pin No.	Pin Name	Input/Output	Bit No.
R6	P82	Input	231
		Enable	230
		Output	229
D13	P17	Input	108
		Enable	107
		Output	106

			Output	249
	36	VCC	—	—
			—	—
			—	—

26.6 Usage Notes	1045	Deleted 1. A reset must always be executed by driving the ET 0, regardless of whether or not the JTAG is to be ac If the JTAG is not to be activated, drive the ETRST, ETMS, and ETDI pins to 1 or the high-impedance s ~~These pins are internally pulled up and are noted in~~ ~~mode.~~
Section 28 Power-Down Modes 28.1.4 Sub-Chip Module Stop Control Registers BH, BL (SUBMSTPBH, SUBMSTPBL)	1062	Amended • SUBMSTPBL **Bit Name** / **Corresponding Module** SMSTPB4 / PECI This bit is not incorporated in the H8S/2463 G The initial values should not be changed.
Section 29 List of Registers 29.1 Register Addresses (Address Order)	1085	Added Notes: 1. The registers related to USB are supporte the H8S/2472 Group. 2. The registers related to PECI are supporte the H8S/2472 Group and the H8S/2462 G

The registers related to PECI are supported on H8S/2472 Group and the H8S/2462 Group.

29.3 Register States in Each Operating Mode	1116	Added Notes: 1. The registers related to USB are supported the H8S/2472 Group. 2. The registers related to PECI are supported the H8S/2472 Group and the H8S/2462 Gr
Section 30 Platform Environment Control Interface (PECI)	1117	Added If this module is not used, the PECI-related pins should handled as follows. PECI is not supported by the H8S/2 Group.
Section 31 Electrical Characteristics Table 31.1 Absolute Maximum Ratings	1119	Amended **Item** — **Symbol** — **Value** PECI reference power supply voltage — PEVref — –0.3 to +
Table 31.10 Timing of On-Chip Peripheral Modules	1145	Item added **Item** — **Symbol** — **Min.** SCI (multiplexed with P51) — Receive data setup time (synchronous) — t_{RXS} — 30.0 SCI (multiplexed with P51) — Receive data hold time (synchronous) — t_{RXH} — 30.0
Figure 31.30 SSU Timing (Slave, CPHS = 1)	1149	Figure title amended
Table 31.16 JTAG Timing	1159	Amended **Item** — **Symbol** — **Min.** — **Max.** ETCK clock cycle time — t_{TCKcyc} — 29.4* — 50*

		H8S/2462	F-ZTAT version (wide temperature specifications)	R4F2462	F2462VF
Figure C.3 Package Dimensions (PTQP0144LC-A)	1171	Added			

A

C

E

F

G

H

I

U

V

W

Renesas 16-Bit Single-Chip Microcomputer
Hardware Manual
H8S/2472, H8S/2463, H8S/2462 Group

Publication Date:	Rev.1.00, Mar. 12, 2008
	Rev.2.00, Aug. 20, 2008
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/2472, H8S/2463, H8S/2462 Group
Hardware Manual

Renesas Electronics Corporation
1753, Shimonumabe, Nakahara-ku, Kawasaki-shi, Kanagawa 211-8668 Japan

REJ09B0403-0200

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