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/2168Group

Hardware Manual

Renesas 16-Bit Single-Chip Microcomputer
H8S Family / H8S/2100 Series

H8S/2168 HD64F2168
H8S/2167 HD64F2167
H8S/2166 HD64F2166

Renesas Electronics
www.renesas.com

Rev.3.00 2004.03

party's rights, originating in the use of any product data, diagrams, charts, programs, algorithi
circuit application examples contained in these materials.

3. All information contained in these materials, including product data, diagrams, charts, progra
algorithms represents information on products at the time of publication of these materials, a
subject to change by Renesas Technology Corp. without notice due to product improvements
other reasons. It is therefore recommended that customers contact Renesas Technology Co
an authorized Renesas Technology Corp. product distributor for the latest product informatio
before purchasing a product listed herein.
The information described here may contain technical inaccuracies or typographical errors.
Renesas Technology Corp. assumes no responsibility for any damage, liability, or other loss
from these inaccuracies or errors.
Please also pay attention to information published by Renesas Technology Corp. by various
including the Renesas Technology Corp. Semiconductor home page (http://www.renesas.co
4. When using any or all of the information contained in these materials, including product data,
diagrams, charts, programs, and algorithms, please be sure to evaluate all information as a t
system before making a final decision on the applicability of the information and products. R
Technology Corp. assumes no responsibility for any damage, liability or other loss resulting f
information contained herein.
5. Renesas Technology Corp. semiconductors are not designed or manufactured for use in a d
system that is used under circumstances in which human life is potentially at stake. Please c
Renesas Technology Corp. or an authorized Renesas Technology Corp. product distributor w
considering the use of a product contained herein for any specific purposes, such as apparat
systems for transportation, vehicular, medical, aerospace, nuclear, or undersea repeater use
6. The prior written approval of Renesas Technology Corp. is necessary to reprint or reproduce
whole or in part these materials.
7. If these products or technologies are subject to the Japanese export control restrictions, they
be exported under a license from the Japanese government and cannot be imported into a c
other than the approved destination.
Any diversion or reexport contrary to the export control laws and regulations of Japan and/or
country of destination is prohibited.
8. Please contact Renesas Technology Corp. for further details on these materials or the produ
contained therein.

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.

- On-Chip Peripheral Modules

  The configuration of the functional description of each module differs according
  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

free running timer (FRT), an 8-bit timer (TMR), a watchdog timer (WDT), a serial comm
interface (SCI), an I^2C bus interface (IIC), an LPC interface (LPC), a D/A converter, an A
converter, and I/O ports as on-chip peripheral modules required for system configuration.

A data transfer controller (DTC) is included as a bus master.

A flash memory (F-ZTAT™*) version is available for this LSI's 256, 384, and 512-kbyte
The CPU and ROM are connected to a 16-bit bus, enabling byte data and word data to be
in a single state. This improves the instruction fetch and process speeds.

Two operating modes are provided, offering a choice of address space and single chip
mode/external extended mode. Boot programming into a flash memory, on-chip emulatio
boundary scan can be selected as special operating modes.

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

Target Users: This manual was written for users who use this LSI in the design of appli
systems. Target users are expected to understand the fundamentals of elec
circuits, logic circuits, and microcomputers.

Objective: This manual was written to explain the hardware functions and electrical
characteristics of this LSI to the target users.
Refer to the H8S/2600 Series, H8S/2000 Series Programming Manual for
detailed description of the instruction set.

Notes on reading this manual:

- In order to understand the overall functions of the chip
  Read this manual in the order of the table of contents. This manual can be roughly cat
  into the descriptions on the CPU, system control functions, peripheral functions and e
  characteristics.

Number notation:	Binary is B'xxxx, hexadecimal is H'xxxx, decim
Signal notation:	An overbar is added to a low-active signal: $\overline{xxxx}$

Related Manuals: The latest versions of all related manuals are available from our we
Please ensure you have the latest versions of all documents you rec
http://www.renesas.com/eng/

H8S/2168 Group manuals:

Document Title	Docume
H8S/2168 Group Hardware Manual	This mar
H8S/2600 Series, H8S/2000 Series Programming Manual	ADE-602

User's manuals for development tools:

Document Title	Docume
H8S, H8/300 Series C/C++ Compiler, Assembler, Optimizing Linkage Editor User's Manual	ADE-702
H8S, H8/300 Series Simulator/Debugger User's Manual	ADE-702
H8S, H8/300 Series High-performance Embedded Workshop, High-performance Debugging Interface Tutorial	ADE-702
High-performance Embedded Workshop User's Manual	ADE-702

Section 7 Data Transfer Controller (DTC)

Section 11 16-Bit Free-Running Timer (FRT)

Appendix

14-bit PWM timer (PWMX)

16-bit free-running timer (FRT)

8-bit timer (TMR)

Watchdog timer (WDT)

Asynchronous or clocked synchronous serial communication interface (SCI)

CRC operation circuit (CRC)

I^2C bus interface (IIC)

LPC interface (LPC)

8-bit D/A converter

10-bit A/D converter

Boundary scan (JTAG)

Clock pulse generator

- On-chip memory

ROM Type	Model	ROM	RAM	Remark
Flash memory Version	HD64F2168	256 kbytes	40 kbytes	
Flash memory Version	HD64F2167	384 kbytes	40 kbytes	
Flash memory Version	HD64F2166	512 kbytes	40 kbytes	

- General I/O ports

  I/O pins: 106

  Input-only pins: 9
- Supports various power-down states
- Compact package

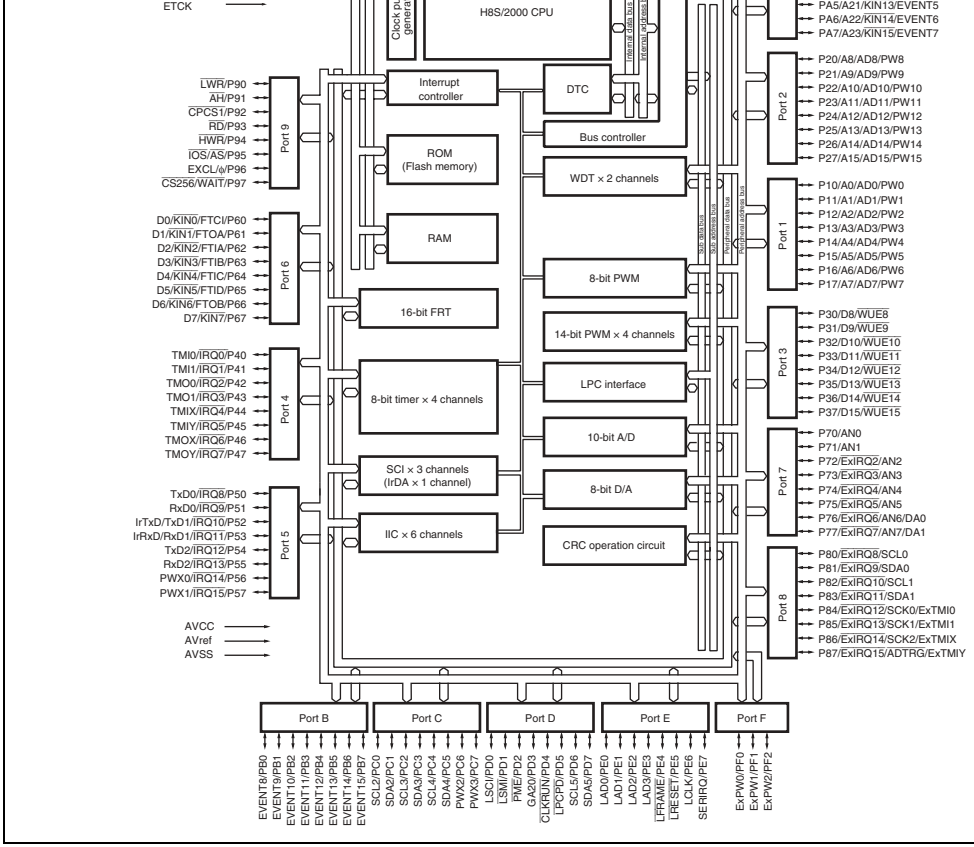


Figure 1.1 Internal Block Diagram

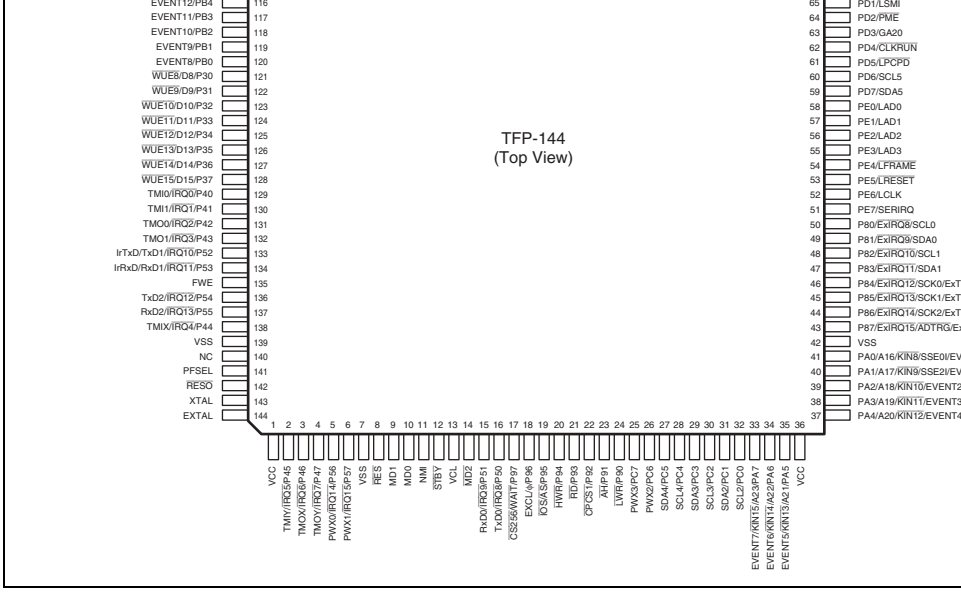


Figure 1.2 Pin Arrangement (TFP-144)

5	P56/IRQ14/PWX0	P56/IRQ14/PWX0	NC
6	P57/IRQ15/PWX1	P57/IRQ15/PWX1	NC
7	VSS	VSS	VSS
8	RES	RES	RES
9	MD1	MD1	VSS
10	MD0	MD0	VSS
11	NMI	NMI	FA9
12	STBY	STBY	VCC
13	VCL	VCL	VCL
14	MD2	MD2	VCC
15	P51/IRQ9/RxD0	P51/IRQ9/RxD0	FA17
16	P50/IRQ8/TxD0	P50/IRQ8/TxD0	NC
17	P97/WAIT/CS256	P97	VCC
18	P96/φ/EXCL	P96/φ/EXCL	NC
19	AS/IOS	P95	FA16
20	HWR	P94	FA15
21	RD	P93	WE
22	P92/ CPCS1	P92	VSS
23	P91/AH	P91	VCC
24	P90/ LWR	P90	VCC
25	PC7/PWX3	PC7/PWX3	NC
26	PC6/PWX2	PC6/PWX2	NC
27	PC5/SDA4	PC5/SDA4	NC

35	PA5/A21/$\overline{\text{KIN13}}$/EVENT5	PA5/$\overline{\text{KIN13}}$/EVENT5	NC
36	VCC	VCC	VCC
37	PA4/A20/$\overline{\text{KIN12}}$/EVENT4	PA4/$\overline{\text{KIN12}}$/EVENT4	NC
38	PA3/A19/$\overline{\text{KIN11}}$/EVENT3	PA3/$\overline{\text{KIN11}}$/EVENT3	NC
39	PA2/A18/$\overline{\text{KIN10}}$/EVENT2	PA2/$\overline{\text{KIN10}}$/EVENT2	NC
40	PA1/A17/$\overline{\text{KIN9}}$/EVENT1/SSE2I	PA1/$\overline{\text{KIN9}}$/EVENT1/SSE2I	NC
41	PA0/A16/$\overline{\text{KIN8}}$/EVENT0/SSE0I	PA0/$\overline{\text{KIN8}}$/EVENT0/SSE0I	NC
42	VSS	VSS	VSS
43	P87/$\overline{\text{ExIRQ15}}$/$\overline{\text{ADTRG}}$/ExTMIY	P87/$\overline{\text{ExIRQ15}}$/$\overline{\text{ADTRG}}$/ExTMIY	NC
44	P86/$\overline{\text{ExIRQ14}}$/SCK2/ExTMIX	P86/$\overline{\text{ExIRQ14}}$/SCK2/ExTMIX	NC
45	P85/$\overline{\text{ExIRQ13}}$/SCK1/ExTMI1	P85/$\overline{\text{ExIRQ13}}$/SCK1/ExTMI1	NC
46	P84/$\overline{\text{ExIRQ12}}$/SCK0/ExTMI0	P84/$\overline{\text{ExIRQ12}}$/SCK0/ExTMI0	NC
47	P83/$\overline{\text{ExIRQ11}}$/SDA1	P83/$\overline{\text{ExIRQ11}}$/SDA1	NC
48	P82/$\overline{\text{ExIRQ10}}$/SCL1	P82/$\overline{\text{ExIRQ10}}$/SCL1	NC
49	P81/$\overline{\text{ExIRQ9}}$/SDA0	P81/$\overline{\text{ExIRQ9}}$/SDA0	NC
50	P80/$\overline{\text{ExIRQ8}}$/SCL0	P80/$\overline{\text{ExIRQ8}}$/SCL0	NC
51	PE7/SERIRQ	PE7/SERIRQ	NC
52	PE6/LCLK	PE6/LCLK	NC
53	PE5/$\overline{\text{LRESET}}$	PE5/$\overline{\text{LRESET}}$	NC
54	PE4/$\overline{\text{LFRAME}}$	PE4/$\overline{\text{LFRAME}}$	NC
55	PE3/LAD3	PE3/LAD3	NC
56	PE2/LAD2	PE2/LAD2	NC
57	PE1/LAD1	PE1/LAD1	NC
58	PE0/LAD0	PE0/LAD0	NC

66	PD0/LSCI	PD0/LSCI	NC
67	AVSS	AVSS	VSS
68	P70/AN0	P70/AN0	NC
69	P71/AN1	P71/AN1	NC
70	P72/$\overline{\text{ExIRQ2}}$/AN2	P72/$\overline{\text{ExIRQ2}}$/AN2	NC
71	P73/$\overline{\text{ExIRQ3}}$/AN3	P73/$\overline{\text{ExIRQ3}}$/AN3	NC
72	P74/$\overline{\text{ExIRQ4}}$/AN4	P74/$\overline{\text{ExIRQ4}}$/AN4	NC
73	P75/$\overline{\text{ExIRQ5}}$/AN5	P75/$\overline{\text{ExIRQ5}}$/AN5	NC
74	P76/$\overline{\text{ExIRQ6}}$/AN6/DA0	P76/$\overline{\text{ExIRQ6}}$/AN6/DA0	NC
75	P77/$\overline{\text{ExIRQ7}}$/AN7/DA1	P77/$\overline{\text{ExIRQ7}}$/AN7/DA1	NC
76	AVCC	AVCC	VCC
77	AVref	AVref	VCC
78	P60/FTCI/$\overline{\text{KIN0}}$/D0	P60/FTCI/$\overline{\text{KIN0}}$	NC
79	P61/FTOA/$\overline{\text{KIN1}}$/D1	P61/FTOA/$\overline{\text{KIN1}}$	NC
80	P62/FTIA/$\overline{\text{KIN2}}$/D2	P62/FTIA/$\overline{\text{KIN2}}$	NC
81	P63/FTIB/$\overline{\text{KIN3}}$/D3	P63/FTIB/$\overline{\text{KIN3}}$	NC
82	P64/FTIC/$\overline{\text{KIN4}}$/D4	P64/FTIC/$\overline{\text{KIN4}}$	NC
83	P65/FTID/$\overline{\text{KIN5}}$/D5	P65/FTID/$\overline{\text{KIN5}}$	NC
84	P66/FTOB/$\overline{\text{KIN6}}$/D6	P66/FTOB/$\overline{\text{KIN6}}$	NC
85	P67/$\overline{\text{KIN7}}$/D7	P67/$\overline{\text{KIN7}}$	VSS
86	VCC	VCC	VCC
87	ETMS	ETMS	NC
88	ETDO	ETDO	NC
89	ETDI	ETDI	NC

97	P26/A14/AD14	P26/PW14	FA14
98	P25/A13/AD13	P25/PW13	FA13
99	P24/A12/AD12	P24/PW12	FA12
100	P23/A11/AD11	P23/PW11	FA11
101	P22/A10/AD10	P22/PW10	FA10
102	P21/A9/AD9	P21/PW9	$\overline{OE}$
103	P20/A8/AD8	P20/PW8	FA8
104	P17/A7/AD7	P17/PW7	FA7
105	P16/A6/AD6	P16/PW6	FA6
106	P15/A5/AD5	P15/PW5	FA5
107	P14/A4/AD4	P14/PW4	FA4
108	P13/A3/AD3	P13/PW3	FA3
109	P12/A2/AD2	P12/PW2	FA2
110	P11/A1/AD1	P11/PW1	FA1
111	VSS	VSS	VSS
112	P10/A0/AD0	P10/PW0	FA0
113	PB7/EVENT15	PB7/EVENT15	NC
114	PB6/EVENT14	PB6/EVENT14	NC
115	PB5/EVENT13	PB5/EVENT13	NC
116	PB4/EVENT12	PB4/EVENT12	NC
117	PB3/EVENT11	PB3/EVENT11	NC
118	PB2/EVENT10	PB2/EVENT10	NC
119	PB1/EVENT9	PB1/EVENT9	NC
120	PB0/EVENT8	PB0/EVENT8	NC

128	P37/D15/WUE15	P37/WUE15	FO7
129	P40/$\overline{\text{IRQ0}}$/TMI0	P40/$\overline{\text{IRQ0}}$/TMI0	NC
130	P41/$\overline{\text{IRQ1}}$/TMI1	P41/$\overline{\text{IRQ1}}$/TMI1	NC
131	P42/$\overline{\text{IRQ2}}$/TMO0	P42/$\overline{\text{IRQ2}}$/TMO0	NC
132	P43/$\overline{\text{IRQ3}}$/TMO1	P43/$\overline{\text{IRQ3}}$/TMO1	NC
133	P52/$\overline{\text{IRQ10}}$/TxD1/IrTxD	P52/$\overline{\text{IRQ10}}$/TxD1/IrTxD	FA18
134	P53/$\overline{\text{IRQ11}}$/RxD1/IrRxD	P53/$\overline{\text{IRQ11}}$/RxD1/IrRxD	NC
135	FWE	FWE	FWE
136	P54/$\overline{\text{IRQ12}}$/TxD2	P54/$\overline{\text{IRQ12}}$/TxD2	NC
137	P55/$\overline{\text{IRQ13}}$/RxD2	P55/$\overline{\text{IRQ13}}$/RxD2	NC
138	P44/$\overline{\text{IRQ4}}$/TMIX	P44/$\overline{\text{IRQ4}}$/TMIX	NC
139	VSS	VSS	VSS
140	NC	NC	NC
141	PFSEL	PFSEL	VCC
142	$\overline{\text{RESO}}$	$\overline{\text{RESO}}$	NC
143	XTAL	XTAL	XTAL
144	EXTAL	EXTAL	EXTAL

		139		
Clock	XTAL	143	Input	For connection to a crystal resonator. A clock can be supplied from the EXTAL example of crystal resonator connectio section 22, Clock Pulse Generator.
	EXTAL	144	Input	
	ϕ	18	Output	Supplies the system clock to external d
	EXCL	18	Input	32.768-kHz external clock for sub clock supplied.
	PFSEL	141	Input	Pin for use by PLL. For an example of F connection, see section 22, Clock Pulse Generator.
Operating mode control	$\overline{\text{MD2}}$ MD1 MD0	14 9 10	Input	These pins set the operating mode. Inp these pins should not be changed durin operation.
System control	$\overline{\text{RES}}$	8	Input	Reset pin. When this pin is low, the chip
	$\overline{\text{RESO}}$	142	Output	Outputs a reset signal to an external de
	$\overline{\text{STBY}}$	12	Input	When this pin is low, a transition is mad hardware standby mode.
	FWE	135	Input	Pin for use by flash memory.
Address bus	A23 to A16	33 to 35 37 to 41	Output	Address output pins
	A15 to A0	96 to 110 112		
Data bus	D15 to D8	128 to 121	Input/ Output	Upper bidirectional data bus
	D7 to D0	85 to 78		Lower bidirectional data bus

	$\overline{\text{LWR}}$	24	Output	This pin is low when the external addres is to be written to, and the lower half of t bus is enabled.
	$\overline{\text{AS}}$/$\overline{\text{IOS}}$	19	Output	This pin is low when address output on t address bus is valid.
	$\overline{\text{CS256}}$	17	Output	Indicates that the 256k-byte area from H to H'FBFFFF is accessed.
	$\overline{\text{CPCS1}}$	22	Output	Indicates that the CP extended area is a
	$\overline{\text{AH}}$	23	Output	Address latch signal for address/data mu bus.
Interrupts	NMI	11	Input	Nonmaskable interrupt request input pin
	$\overline{\text{IRQ15}}$ to $\overline{\text{IRQ0}}$	6, 5, 137, 136, 134, 133, 15, 16, 4 to 2, 138, 132 to 129	Input	These pins request a maskable interrupt Selectable to which pin of $\overline{\text{IRQn}}$ or $\overline{\text{ExIRQ}}$ insert IRQ15 to IRQ2 interrupts.
	$\overline{\text{ExIRQ15}}$ to $\overline{\text{ExIRQ2}}$	43 to 50 75 to 70		
Boundary scan	$\overline{\text{ETRST}}$	91	Input	Boundary scan interface pins
	ETMS	87	Input	
	ETDO	88	Output	
	ETDI	89	Input	
	ETCK	90	Input	

	FTID			
8-bit timer (TMR_0, TMR_1, TMR_X, TMR_Y)	TMO0 TMO1 TMOX TMOY	131 132 3 4	Output	Waveform output pins with output comp function
	TMI0 TMI1 TMIX TMIY ExTMI0 ExTMI1 ExTMIX ExTMIY	129 130 138 2 46 45 44 43	Input	External event input pins and counter r pins. Selectable to which pin of TMIn o to insert external event and counter res
Serial communi-cation Interface (SCI_0, SCI_1, SCI_2)	TxD0 to TxD2	16, 133 136	Output	Transmit data output pins
	RxD0 to RxD2	15, 134 137	Input	Receive data input pins
	SCK0 to SCK2	46, 45 44	Input/ Output	Clock input/output pins. Output format i push-pull output.
	SSE0I	41	Input	Input pin to halt SCI_0
	SSE2I	40	Input	Input pin to halt SCI_2
SCI with IrDA (SCI)	IrTxD	133	Output	Encoded data output pin for IrDA
	IrRxD	134	Input	Encoded data input pin for IrDA
I^2C bus interface (IIC)	SCL0 to SCL5	50, 48, 32, 30, 28, 60	Input/ Output	IIC clock input/output pins. These pins bus directly with the NMOS open drain
	SDA0 to SDA5	49, 47, 31, 29, 27, 59	Input/ Output	IIC data input/output pins. These pins c bus directly with the NMOS open drain

D/A converter (DAC)	DA0 DA1	74 75	Output	Analog output pins
A/D converter (ADC) D/A converter (DAC)	AVCC	76	Input	Analog power supply pins for the A/D co and D/A converter. When the A/D conve D/A converter are not used, these pins s connected to the system power supply (-
	AVref	77	Input	Reference voltage input pin for the A/D c and D/A converter. When the A/D conve D/A converter are not used, this pin shou connected to the system power supply (-
	AVSS	67	Input	Ground pins for the A/D converter and D converter. These pins should be connec system power supply (0 V).

	GA20	63	Input/ Output	GATE A20 control signal output pin. Th input of an output state is enabled.
	$\overline{\text{CLKRUN}}$	62	Input/ Output	LCLK restart request I/O pin
	$\overline{\text{LPCPD}}$	61	Input	LPC module shutdown control input pin
Event Counter	EVENT15 to EVENT0	113 to 120, 33 to 35, 37 to 41	Input	Event counter input pins.

P67 to P60	85 to 78	Input/ Output	Eight input/output pins
P77 to P70	75 to 68	Input	Eight input pins
P87 to P80	43 to 50	Input/ Output	Eight input/output pins
P97 to P90	17 to 24	Input/ Output	Eight input/output pins (P96 input p
PA7 to PA0	33 to 35, 37 to 41	Input/ Output	Eight input/output pins
PB7 to PB0	113 to 120	Input/ Output	Eight input/output pins
PC7 to PC0	25 to 32	Input/ Output	Eight input/output pins
PD7 to PD0	59 to 66	Input/ Output	Eight input/output pins
PE7 to PE0	51 to 58	Input/ Output	Eight input/output pins
PF2 to PF0	92 to 94	Input/ Output	Three input/output pins

— Can execute H8/300 CPU and H8/300H CPU object programs

- General-register architecture
  — Sixteen 16-bit general registers also usable as sixteen 8-bit registers or eight 32-b
- Sixty-five basic instructions
  — 8/16/32-bit arithmetic and logic instructions
  — Multiply and divide instructions
  — Powerful bit-manipulation instructions
- Eight addressing modes
  — Register direct [Rn]
  — Register indirect [@ERn]
  — Register indirect with displacement [@(d:16,ERn) or @(d:32,ERn)]
  — Register indirect with post-increment or pre-decrement [@ERn+ or @–ERn]
  — Absolute address [@aa:8, @aa:16, @aa:24, or @aa:32]
  — Immediate [#xx:8, #xx:16, or #xx:32]
  — Program-counter relative [@(d:8,PC) or @(d:16,PC)]
  — Memory indirect [@@aa:8]
- 16 Mbytes address space
  — Program: 16 Mbytes
  — Data: 16 Mbytes
- High-speed operation
  — All frequently-used instructions are executed in one or two states
  — 8/16/32-bit register-register add/subtract: 1 state
  — 8 × 8-bit register-register multiply: 12 states (MULXU.B), 13 states (MULXS.B)
  — 16 ÷ 8-bit register-register divide: 12 states (DIVXU.B)
  — 16 × 16-bit register-register multiply: 20 states (MULXU.W), 21 states (MULXS
  — 32 ÷ 16-bit register-register divide: 20 states (DIVXU.W)

- Register configuration

  The MAC register is supported only by the H8S/2600 CPU.
- Basic instructions

  The four instructions MAC, CLRMAC, LDMAC, and STMAC are supported only by H8S/2600 CPU.
- The number of execution states of the MULXU and MULXS instructions

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

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

space.

- Enhanced instructions
  - Addressing modes of bit-manipulation instructions have been enhanced.
  - Signed multiply and divide instructions have been added.
  - Two-bit shift and two-bit rotate 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 are executed twice as fast.

2.1.3 Differences from H8/300H CPU

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

- Additional control register
  - One 8-bit control register has been added.
- Enhanced instructions
  - Addressing modes of bit-manipulation instructions have been enhanced.
  - Two-bit shift and two-bit rotate 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 are executed twice as fast.

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

- 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 extended register En is used as a 16-bit register it can contain any value, even v corresponding general register (Rn) is used as an address register. (If general register referenced in the register indirect addressing mode with pre-decrement (@–Rn) or pos increment (@Rn+) and a carry or borrow occurs, the value in the corresponding exten 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 tal branch address is stored per 16 bits. The exception vector table in normal mode is sho 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) operan providing a 16-bit branch address. Branch addresses can be stored in the top area fron to H'00FF. Note that this area is also used for the exception vector table.

- Stack structure

  In normal mode, when the program counter (PC) is pushed onto the stack in a subrout in normal mode, and the PC and condition-code register (CCR) are pushed onto the st exception handling, they are stored as shown in figure 2.2. The extended control regis (EXR) is not pushed onto the stack. For details, see section 4, Exception Handling.

Note: Normal mode is not available in this LSI.

Figure 2.1 Exception Vector Table (Normal Mode)

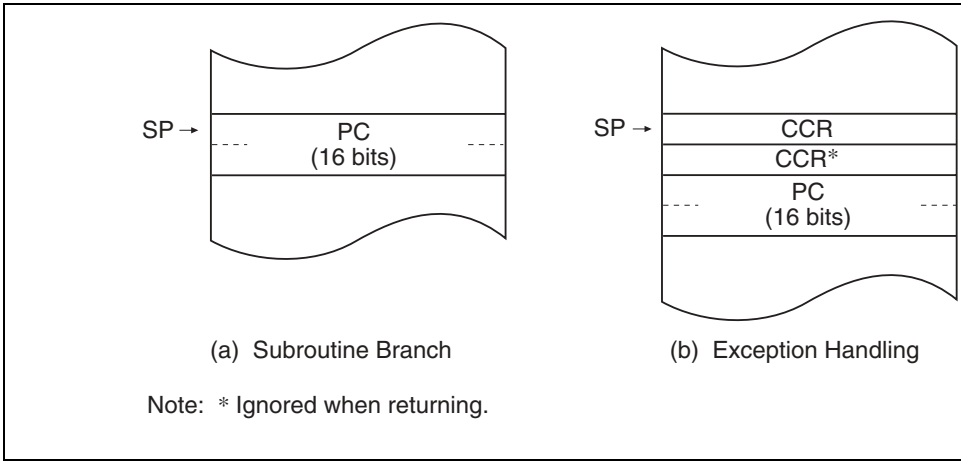


Figure 2.2 Stack Structure in Normal Mode

table in 32-bit units. In each 32 bits, the upper eight bits are ignored and a branch addr
stored in the lower 24 bits (see figure 2.3). For details of the exception vector table, se
4, Exception Handling.

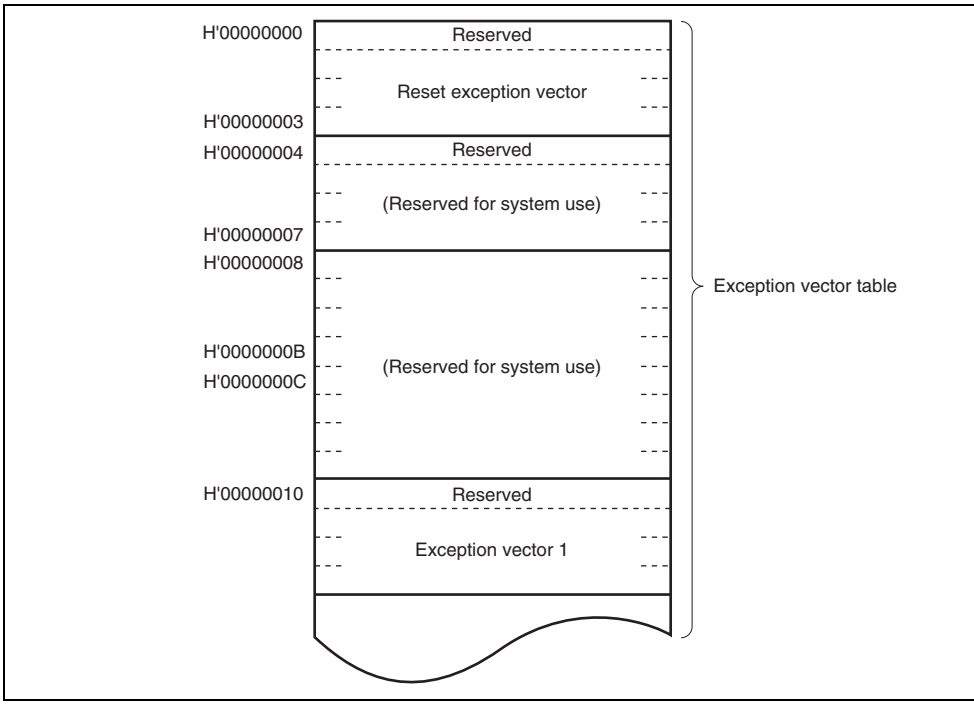


Figure 2.3 Exception Vector Table (Advanced Mode)

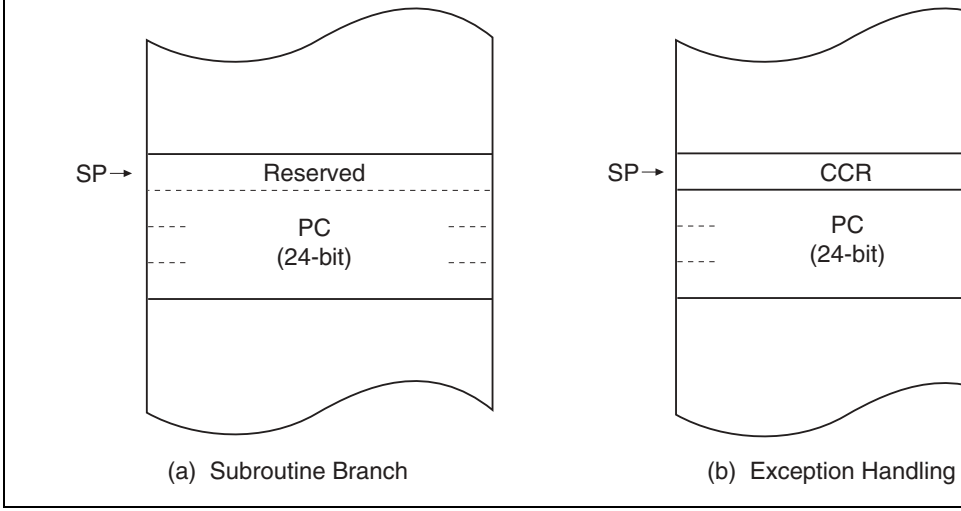


Figure 2.4 Stack Structure in Advanced Mode

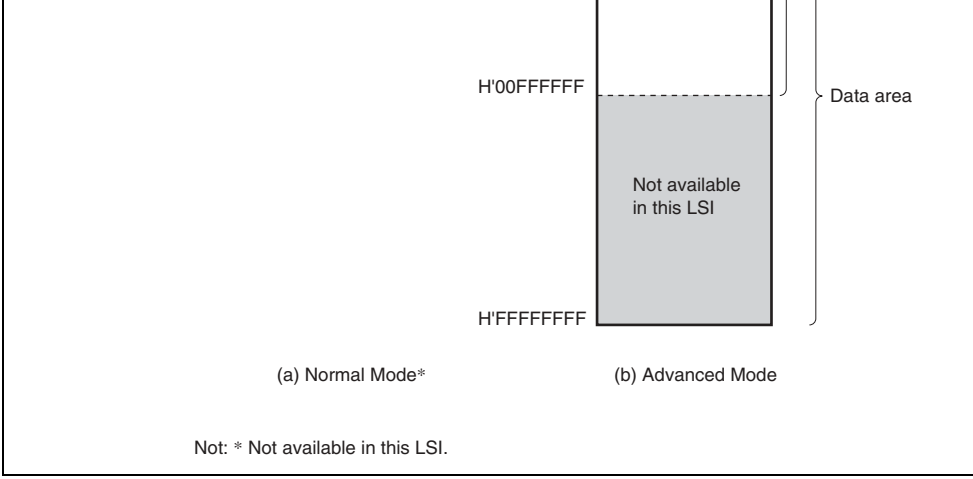
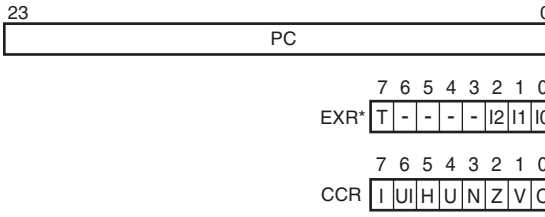


Figure 2.5 Memory Map

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

Control Registers



[Legend]

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

Note: * Does not affect operation in this LSI.

Figure 2.6 CPU Internal Registers

When the general registers are used as 8-bit registers, the R registers are divided into 8-bi registers designated by the letters RH (R0H to R7H) and RL (R0L to R7L). These registe functionally equivalent, providing sixteen 8-bit registers at the maximum.

The usage of each register can be selected independently.

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

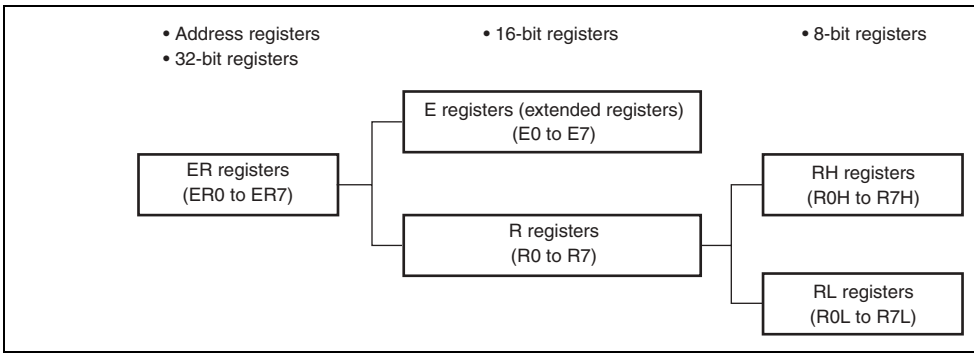


Figure 2.7 Usage of General Registers

2.4.2 Program Counter (PC)

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

2.4.3 Extended Control Register (EXR)

EXR does not affect operation in this LSI.

Bit	Bit Name	Initial Value	R/W	Description
7	T	0	R/W	Trace Bit
				Does not affect operation in this LSI.
6 to 3	–	All 1	R	Reserved
				These bits are always read as 1.
2 to 0	I2	1	R/W	Interrupt Mask Bits 2 to 0
	I1	1		Do not affect operation in this LSI.
	I0	1		

				at the start of an exception-handling sequence. Fo see section 5, Interrupt Controller.
6	UI	Undefined	R/W	User Bit or Interrupt Mask Bit Can be written to and read from by software using LDC, STC, ANDC, ORC, and XORC instructions.
5	H	Undefined	R/W	Half-Carry Flag When the ADD.B, ADDX.B, SUB.B, SUBX.B, CMI NEG.B instruction is executed, this flag is set to 1 a carry or borrow at bit 3, and cleared to 0 otherwi the ADD.W, SUB.W, CMP.W, or NEG.W instructio executed, the H flag is set to 1 if there is a carry o at bit 11, and cleared to 0 otherwise. When the AD SUB.L, CMP.L, or NEG.L instruction is executed, is set to 1 if there is a carry or borrow at bit 27, an to 0 otherwise.
4	U	Undefined	R/W	User Bit Can be written to and read from by software using LDC, STC, ANDC, ORC, and XORC instructions.
3	N	Undefined	R/W	Negative Flag Stores the value of the most significant bit of data bit.
2	Z	Undefined	R/W	Zero Flag Set to 1 to indicate zero data, and cleared to 0 to i non-zero data.
1	V	Undefined	R/W	Overflow Flag Set to 1 when an arithmetic overflow occurs, and 0 otherwise.

2.4.3 Initial Register Values

Reset exception handling loads the CPU's program counter (PC) from the vector table, c
trace (T) bit in EXR to 0, and sets the interrupt mask (I) bits in CCR and EXR to 1. The
CCR bits and the general registers are not initialized. Note that the stack pointer (ER7) i
undefined. The stack pointer should therefore be initialized by an MOV.L instruction ex
immediately after a reset.

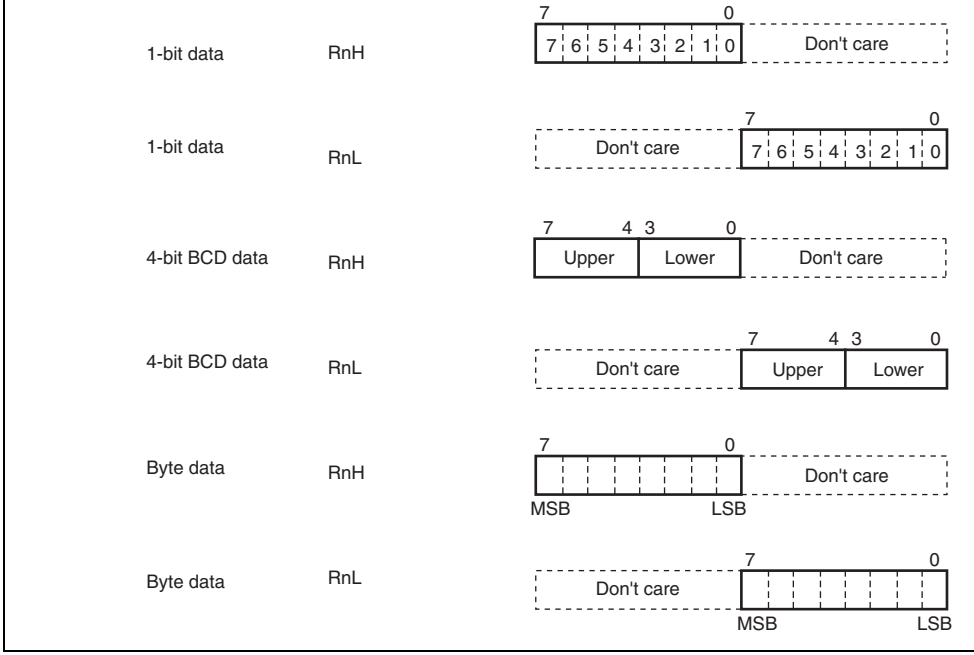
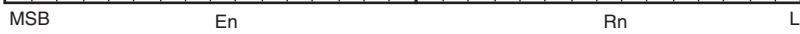


Figure 2.9 General Register Data Formats (1)



[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)

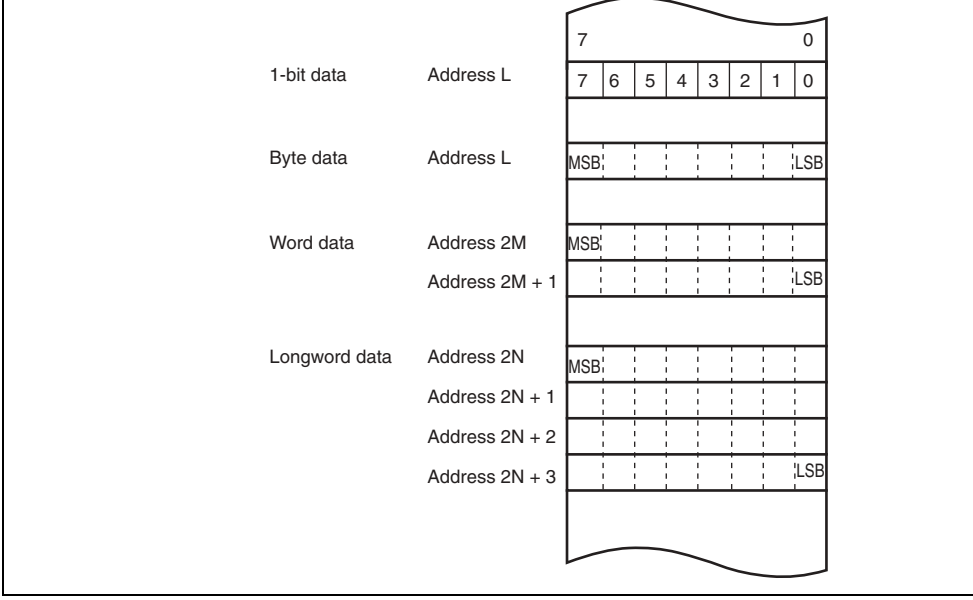


Figure 2.10 Memory Data Formats

Arithmetic operations	ADD, SUB, CMP, NEG	B/W/L
	ADDX, SUBX, DAA, DAS	B
	INC, DEC	B/W/L
	ADDS, SUBS	L
	MULXU, DIVXU, MULXS, DIVXS	B/W
	EXTU, EXTS	W/L
	TAS	B
Logic operations	AND, OR, XOR, NOT	B/W/L
Shift	SHAL, SHAR, SHLL, SHLR, ROTL, ROTR, ROTXL, ROTXR	B/W/L
Bit manipulation	BSET, BCLR, BNOT, BTST, BLD, BILD, BST, BIST, BAND, BIAND, BOR, BIOR, BXOR, BIXOR	B
Branch	B_{cc}*4, JMP, BSR, JSR, RTS	–
System control	TRAPA, RTE, SLEEP, LDC, STC, ANDC, ORC, XORC, NOP	–
Block data transfer	EEPMOV	–

Notes: B: Byte size; W: Word size; L: Longword size.

1. POP.W Rn and PUSH.W Rn are identical to MOV.W @SP+, Rn and MOV.W SP. POP.L ERn and PUSH.L ERn are identical to MOV.L @SP+, ERn and M @-SP.
2. Since register ER7 functions as the stack pointer in an STM/LDM instruction, be used as an STM/LDM register.
3. Cannot be used in this LSI.
4. B_{cc} is the general name for conditional branch instructions.

(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 exclusive OR
→	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).

		Pushes a general register onto the stack. PUSH.W Rn is ident MOV.W Rn, @-SP. PUSH.L ERn is identical to MOV.L ERn, @
LDM*[2]	L	@SP+ → Rn (register list) Pops two or more general registers from the stack.
STM*[2]	L	Rn (register list) → @-SP Pushes two or more general registers onto the stack.

Notes: 1. Size refers to the operand size.
B: Byte
W: Word
L: Longword
2. Since register ER7 functions as the stack pointer in an STM/LDM instruction,
be used as an STM/LDM register.

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

[Legend]

*: Size refers to the operand size.

B: Byte

W: Word

L: Longword

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	B	@ERd – 0, 1 → (<bit 7> of @ERd) Tests memory contents, and sets the most significant bit (bit 7

[Legend]

*: Size refers to the operand size.

B: Byte

W: Word

L: Longword

NOT	B/W/L	~ Rd → Rd
		Takes the one's complement (logical complement) of data in a g register.

[Legend]

*: Size refers to the operand size.

B: Byte

W: Word

L: Longword

Table 2.6 Shift Instructions

Instruction	Size*	Function
SHAL	B/W/L	Rd (shift) → Rd
SHAR		Performs an arithmetic shift on data in a general register. 1-bit o shift is possible.
SHLL	B/W/L	Rd (shift) → Rd
SHLR		Performs a logical shift on data in a general register. 1-bit or 2 b possible.
ROTL	B/W/L	Rd (rotate) → Rd
ROTR		Rotates data in a general register. 1-bit or 2 bit rotation is possil
ROTXL	B/W/L	Rd (rotate) → Rd
ROTXR		Rotates data including the carry flag in a general register. 1-bit rotation is possible.

[Legend]

*: Size refers to the operand size.

B: Byte

W: Word

L: Longword

		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 clears the Z flag accordingly. The bit number is specified by 3- immediate data or the lower three bits of a general register.
BAND	B	C ∧ (<bit-No.> of <EAd>) → C Logically ANDs the carry flag with a specified bit in a general r memory operand and stores the result in the carry flag.
BIAND	B	C ∧ (<bit-No.> of <EAd>) → C Logically ANDs the carry flag with the inverse of a specified bit general register or memory operand and stores the result in th flag. The bit number is specified by 3-bit immediate data.
BOR	B	C ∨ (<bit-No.> of <EAd>) → C Logically ORs the carry flag with a specified bit in a general re memory operand and stores the result in the carry flag.
BIOR	B	C ∨ (~ <bit-No.> of <EAd>) → C Logically ORs the carry flag with the inverse of a specified bit i general register or memory operand and stores the result in th flag. The bit number is specified by 3-bit immediate data.

[Legend]

*: Size refers to the operand size.

B: Byte

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

[Legend]

*: Size refers to the operand size.

B: Byte

	(high or same)	
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 memo operand. Although CCR and EXR are 8-bit registers, word-size transfers are performed between them and memory. The uppe bits are valid.
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 exclusive-ORs the CCR or EXR contents with immed
NOP	–	PC + 2 → PC Only increments the program counter.

[Legend]

*: Size refers to the operand size.

B: Byte

W: Word

	data for the number of bytes set in R4L or R4 to the address set in ER6. Execution of the next instruction begins as soon as the transf completed.

2.6.2 Basic Instruction Formats

The H8S/2000 CPU instructions consist of 2-byte (1-word) units. An instruction consist operation field (op), a register field (r), an effective address extension (EA), and a condi (cc).

Figure 2.11 shows examples of instruction formats.

- Operation field
  Indicates the function of the instruction, the addressing mode, and the operation to b out on the operand. The operation field always includes the first four bits of the instr Some instructions have two operation fields.
- Register field
  Specifies a general register. Address registers are specified by 3-bit, and data registe or 4-bit. Some instructions have two register fields, and some have no register field.
- Effective address extension
  8-, 16-, or 32-bit specifying immediate data, an absolute address, or a displacement.
- Condition field
  Specifies the branching condition of Bcc instructions.

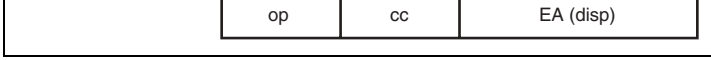


Figure 2.11 Instruction Formats (Examples)

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	@ERn+
	Register indirect with pre-decrement	@–ERn
5	Absolute address	@aa:8/@aa:16/@aa:24/@aa:32
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 code specifies an 8-, 16-, or 32-bit general register w
contains the operand. R0H to R7H and R0L to R7L can be specified as 8-bit registers. R
and E0 to E7 can be specified as 16-bit registers. ER0 to ER7 can be specified as 32-bit

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

Register Indirect with Post-Increment—@ERn+: The register field of the instruction c specifies an address register (ERn) which contains the address of a memory operand. Aft operand is accessed, 1, 2, or 4 is added to the address register contents and the sum is sto address register. The value added is 1 for byte access, 2 for word access, and 4 for longw access. For word or longword transfer instructions, the register value should be even.

Register Indirect with Pre-Decrement—@-ERn: The value 1, 2, or 4 is subtracted fro address register (ERn) specified by the register field in the instruction code, and the resul becomes the address of a memory operand. The result is also stored in the address registe value subtracted is 1 for byte access, 2 for word access, and 4 for longword access. For w longword transfer instructions, the register value should be even.

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 16 bits are all assumed to be 1 (H For a 16-bit absolute address, the upper 16 bits are a sign extension. For a 32-bit absolute the entire address space is accessed.

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).

The 8-bit (#xx:8), 16-bit (#xx:16), or 32-bit (#xx:32) immediate data contained in a instr code can be used directly as an operand.

The ADDS, SUBS, INC, and DEC instructions implicitly contain immediate data in thei instruction codes. Some bit manipulation instructions contain 3-bit immediate data in th instruction code, specifying a bit number. The TRAPA instruction contains 2-bit immed in its instruction code, specifying a vector address.

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

This mode can be used by the Bcc and BSR instructions. An 8-bit or 16-bit displacemen contained in the instruction code is sign-extended to 24-bit and added to the 24-bit addre indicated by the PC value to generate a 24-bit branch address. Only the lower 24-bit of t address are valid; the upper eight bits are all assumed to be 0 (H′00). The PC value to wh displacement is added is the address of the first byte of the next instruction, so the possib branching range is –126 to +128-byte (–63 to +64 words) or –32766 to +32768-byte (–1 +16384 words) from the branch instruction. The resulting value should be an even numb

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 the instruction code to fetched at the address preceding the specified address. (For further information, see sectio Memory Data Formats.)

Note: * Not available in this LSI.

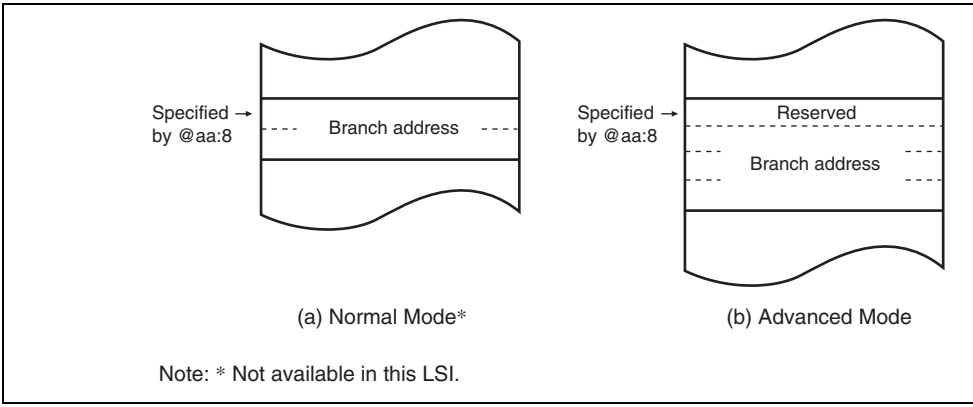
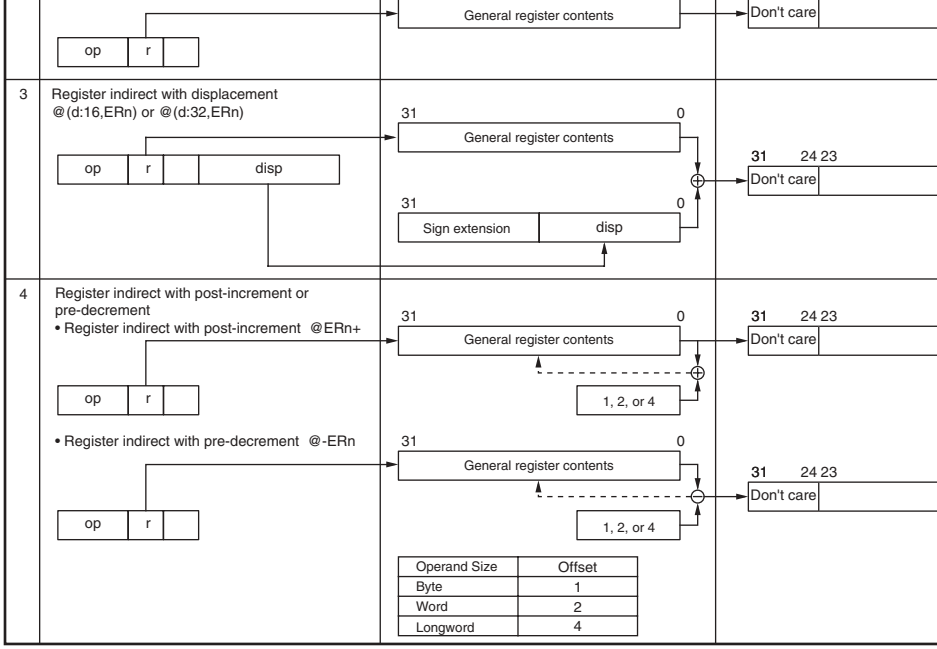
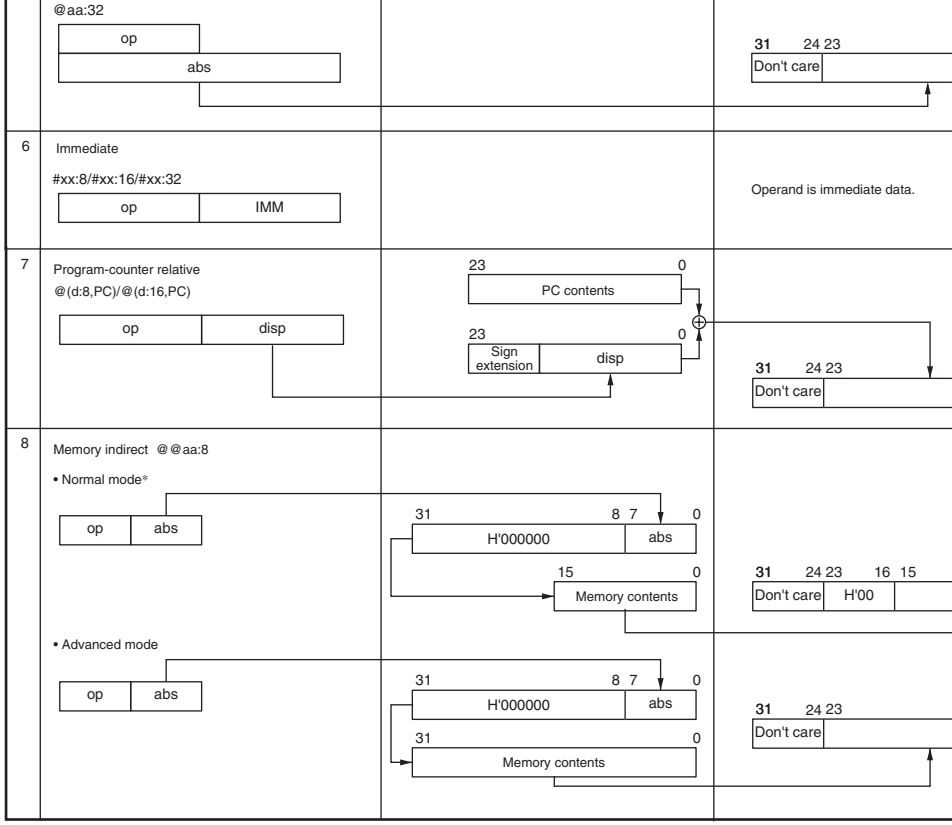


Figure 2.12 Branch Address Specification in Memory Indirect Addressing M


op
r
General register contents
Don't care
3
Register indirect with displacement
@(d:16,ERn) or @(d:32,ERn)
op
r
disp
31
0
General register contents
31
0
Sign extension
disp
31
24 23
Don't care
4
Register indirect with post-increment or
pre-decrement
• Register indirect with post-increment @ERn+
op
r
31
0
General register contents
1, 2, or 4
31
24 23
Don't care
• Register indirect with pre-decrement @-ERn
op
r
31
0
General register contents
1, 2, or 4
31
24 23
Don't care
Operand Size
Offset
Byte
1
Word
2
Longword
4



Note: * 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, see section 4, Exception Handling.

- Program execution state

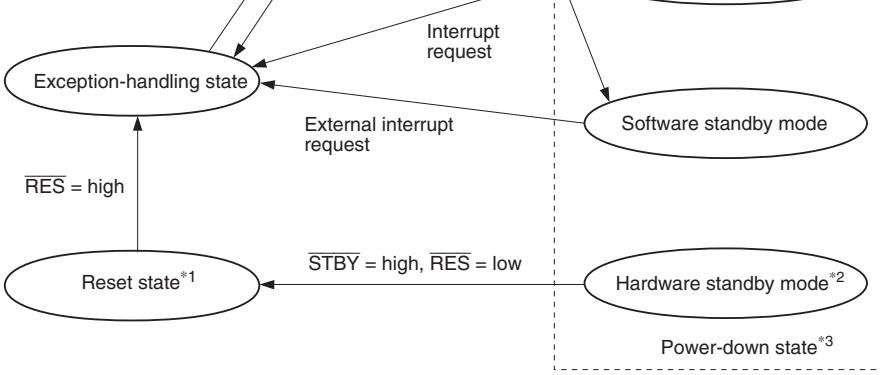
  In this state the CPU executes program instructions in sequence.

- Bus-released state

  In a product which has a bus master other than the CPU, such as a data transfer contr
  (DTC), the bus-released state occurs when the bus has been released in response to a
  request from a bus master other than the CPU. While the bus is released, the CPU ha
  operations.

- 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 hardware standby mode. Fo
  see section 23, Power-Down Modes.



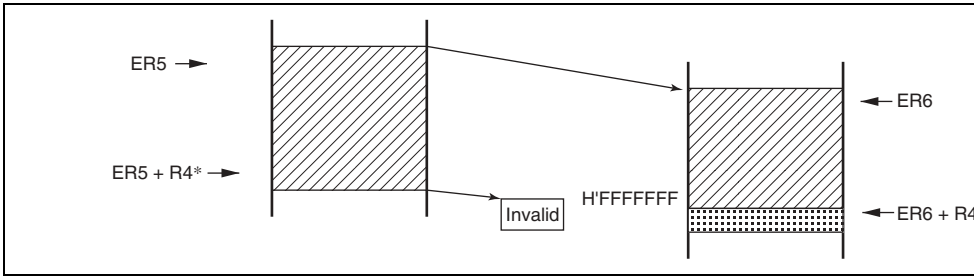
Notes: 1. From any state except hardware standby mode, a transition to the reset state occurs whenever goes low. A transition can also be made to the reset state when the watchdog timer overflows
2. From any state, a transition to hardware standby mode occurs when STBY goes low.
3. The power-down state also includes watch mode, subactive mode, subsleep mode, etc. For de refer to section 23, Power-Down Modes.

Figure 2.13 State Transitions

manipulated for a port.

In addition, the BCLR instruction can be used to clear the flag of the internal I/O registe
case, if the flag to be cleared has been set to 1 by an interrupt processing routine, the fla
be read before executing the BCLR instruction.

2. Set R4* and ER6 so that the end address of the destination address (value of ER6 + R
not exceed H'00FFFFFF (the value of ER6 must not change from H'00FFFFFF to H'0
during execution).



Note: * For byte transfer R4L is used.

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.

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 1 0	MDS2 MDS1 MDS0	—* —* —*	R R R	Mode Select 2 to 0 These bits indicate the input levels at mode pi MD1, and MD0) (the current operating mode). MDS2, MDS1, and MDS0 correspond to $\overline{MD2}$ and MD0, respectively. MDS2 to MDS0 are re bits and they cannot be written to. The mode p MD1, and MD0) input levels are latched into th when MDCR is read. These latches are cance reset.

Note: * The initial values are determined by the settings of the $\overline{MD2}$, MD1, and MD0 p

				1: CS256 pin Outputs low when a 256-kbyte expansion addresses H'F80000 to H'FBFFFF is acce
6	IOSE	0	R/W	IOS Enable Enables or disables $\overline{AS}$/$\overline{IOS}$ pin function in e mode. 0: $\overline{AS}$ pin Outputs low when an external area is acce 1: $\overline{IOS}$ pin Outputs low when an IOS expansion area addresses H'FFF000 to H'FFF7FF is acce
5	INTM1	0	R	These bits select the control mode of the inte
4	INTM0	0	R/W	controller. For details on the interrupt control see section 5.6, Interrupt Control Modes and Operation. 00: Interrupt control mode 0 01: Interrupt control mode 1 10: Setting prohibited 11: Setting prohibited
3	XRST	1	R	External Reset This bit indicates the reset source. A reset is by an external reset input, or when the watch overflows. 0: A reset is caused when the watchdog time overflows. 1: A reset is caused by an external reset.
2	NMIEG	0	R/W	NMI Edge Select Selects the valid edge of the NMI interrupt in 0: An interrupt is requested at the falling edg input 1: An interrupt is requested at the rising edge input

				control register of the KINn pin in an area fr H'FFFFF0 to H'FFFFF7 and from H'FFFFF H'FFFFFF.
0	RAME	1	R/W	RAM Enable Enables or disables on-chip RAM. The RAME 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 b
5	IICX0	0	R/W	select a transfer rate in master mode togeth bits CKS2 to CKS0 in the I^2C bus mode regi (ICMR). For details on the transfer rate, see 15.3. The IICXn bit controls IIC_n. (n = 0 to

				H'FFFFDF. 1: IIC_1 registers are accessed in an area H'FFFF88 to H'FFFF89 and from H'FFF H'FFFF8F. PWMX registers are accessed in an are H'FFFFA0 to H'FFFFA1 and from H'FFF H'FFFFA7. IIC_0 registers are accessed in an area H'FFFFD8 to H'FFFFD9 and from H'FFF H'FFFFDF.
3	FLSHE	0	R/W	Flash Memory Control Register Enable Enables or disables CPU access for flash r registers (FCCS, FPCS, FECS, FKEY, FM FTDAR), control registers of power-down s (SBYCR, LPWRCR, MSTPCRH, MSTPCR control registers of on-chip peripheral modu (BCR2, WSCR2, PCSR, SYSCR2). 0: Area from H'FFFE88 to H'FFFE8F is re Control registers of power-down states chip peripheral modules are accessed i from H'FFFF80 to H'FFFF87. 1: Control registers of flash memory are a an area from H'FFFE88 to H'FFFE8F. Area from H'FFFF80 to H'FFFF87 is re
2	—	0	R/W	Reserved The initial value should not be changed.
1 0	ICKS1 ICKS0	0 0	R/W R/W	Internal Clock Source Select 1, 0 These bits select a clock to be input to the counter (TCNT) and a count condition toge bits CKS2 to CKS0 in the timer control regi (TCR). For details, see section 12.3.4, Tim Register (TCR).

Port 6 functions as a data bus port when the ABW bit in WSCR is cleared to 0.

Multiplex extended mode:
When 8-bit bus is specified, port 2 functions as the port for address output and data input/
regardless of the setting of the data direction register (DDR). Port 1 can be used as a gene
When 16-bit bus is specified, ports 1 and 2 function as the port for address output and dat
input/output regardless of the setting of the data direction register (DDR).

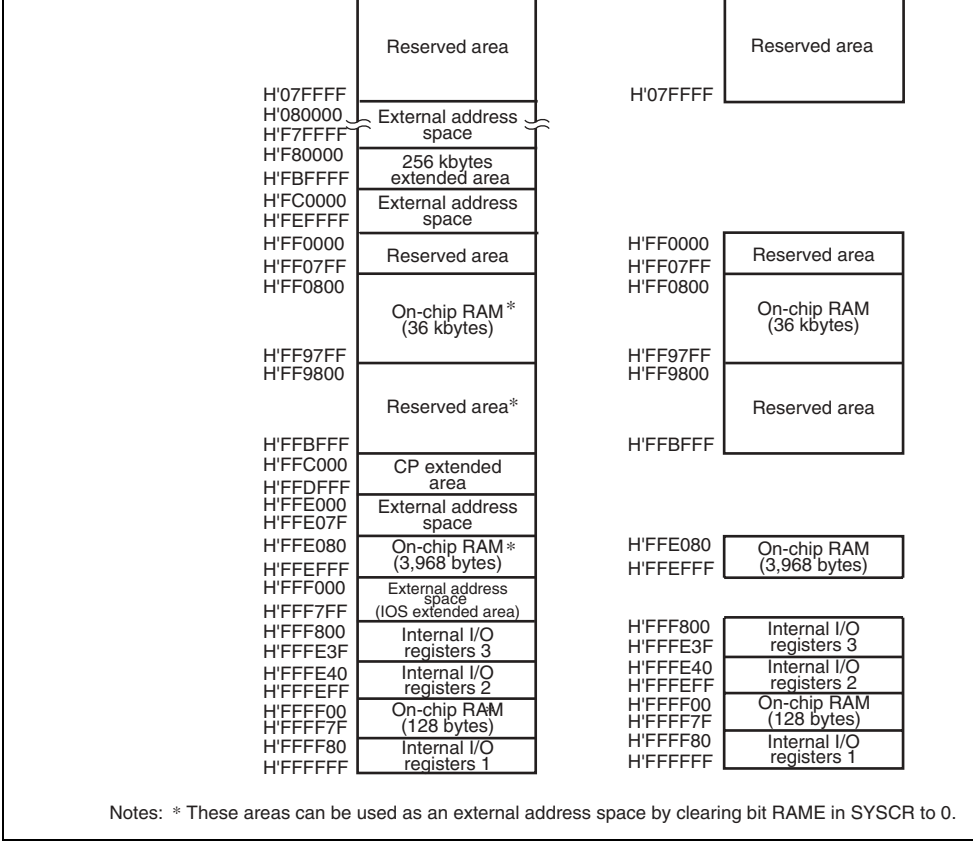
3.3.2 Pin Functions in Each Operating Mode

Pin functions of ports 1 to 3, 6, 9, and A depend on the extended mode. Table 3.2 shows p
functions in each operating mode.

	I/O port95 to I/O port93	I/O port* or Control signal output	I/O port* or Control signal
	I/O port92	I/O port* or Control signal output	I/O port* or Control signal
	I/O port91	I/O port* or Control signal output	I/O port* or Control signal
	I/O port90	I/O port* or Control signal output	I/O port* or Control signal
Port A		I/O port* or Address bus output	I/O port*

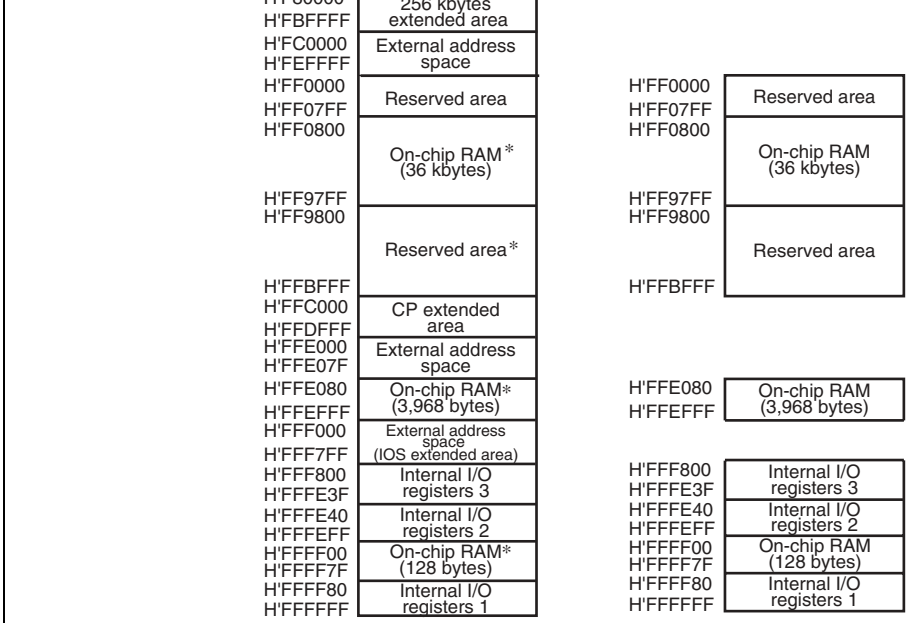
[Legend]

*: After reset



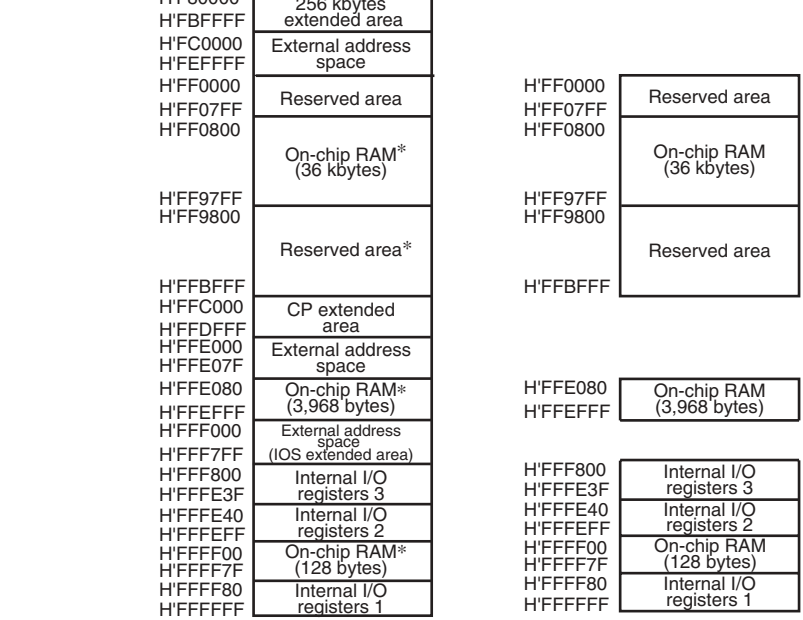
Notes: * These areas can be used as an external address space by clearing bit RAME in SYSCR to 0.

Figure 3.1 H8S/2168 Address Map



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

Figure 3.2 H8S/2167 Address Map



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

Figure 3.3 H8S/2166 Address Map

		pin, or when the watchdog timer overflows.
	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.
	Direct transition	Starts when a direct transition occurs as the result SLEEP instruction execution.
Low	Trap instruction	Started by execution of a trap (TRAPA) instruction instruction exception handling requests are accep times in program execution state.

Direct transition		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
Direct transition (clock switchover)		12	H'000030 to H'000033
Reserved for system use		13 ⎮ 15	H'000034 to H'000037 ⎮ 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
External interrupt	KIN7 to KIN0	30	H'000078 to H'00007B
	KIN15 to KIN8	31	H'00007C to H'00007F
	Reserved	32	H'000080 to H'000083
	WUE15 to WUE8	33	H'000084 to H'000087

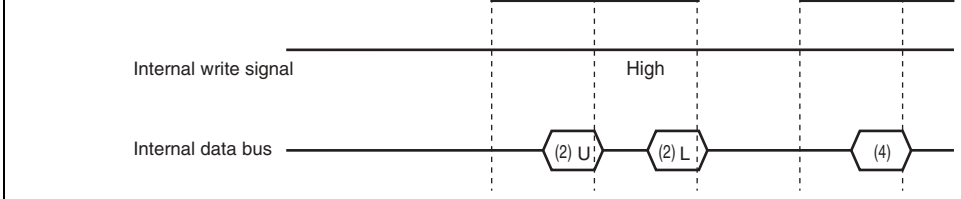
	IRQ13	61	H'0000F4 to H'0000F7
	IRQ14	62	H'0000F8 to H'0000FB
	IRQ15	63	H'0000FC to H'0000FF
Internal interrupt*		64 | 119	H'000100 to H'000103 | H'0001DC to H'0001DF

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

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, SUB and SUBMSTPA) are initialized, and all modules except the DTC operate in module sto 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.

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

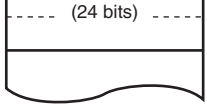


Figure 4.2 Stack Status after Exception Handling

```
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.

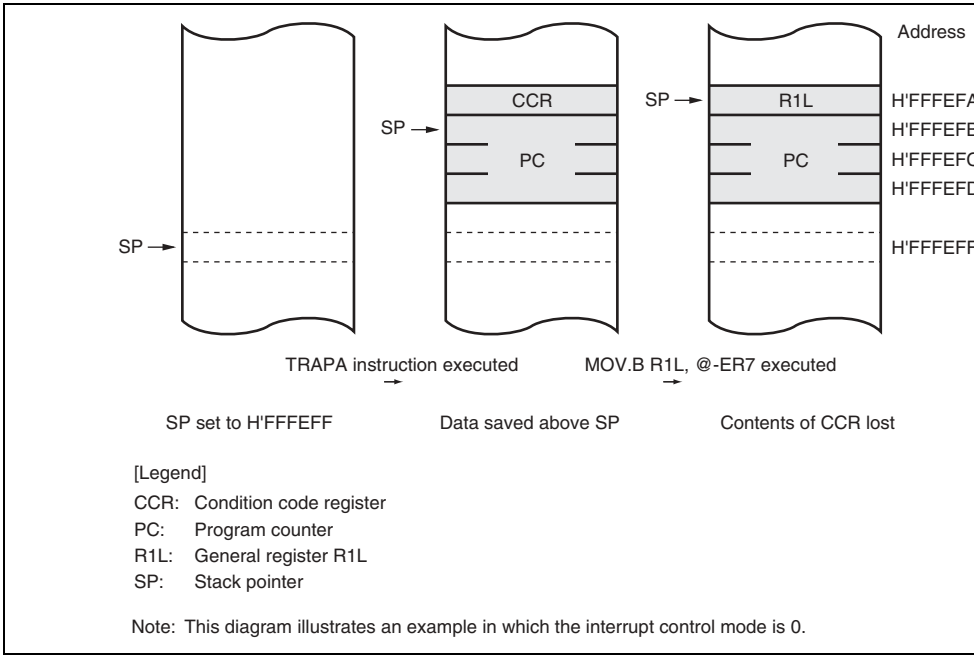


Figure 4.3 Operation when SP Value Is Odd

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

- Independent vector addresses

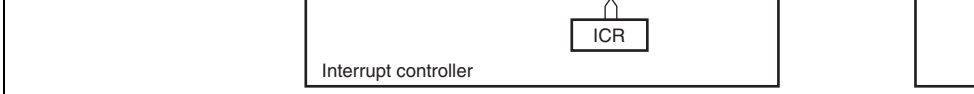
  All interrupt sources are assigned independent vector addresses, making it unnecessa source to be identified in the interrupt handling routine.

- Forty-one 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 $\overline{\text{IRQ15}}$ to $\overline{\text{IRQ0}}$. An interrupt is requested at the falling e $\overline{\text{KIN15}}$ to $\overline{\text{KIN0}}$ and $\overline{\text{WUE15}}$ to $\overline{\text{WUE8}}$.

- DTC control

  The DTC can be activated by an interrupt request.



[Legend]
ICR: Interrupt control register
ISCR: IRQ sense control register
IER: IRQ enable register
ISR: IRQ status register
KMIMR: Keyboard matrix interrupt mask register
WUEMR: Wake-up event interrupt mask register
SYSCR: System control register

Figure 5.1 Block Diagram of Interrupt Controller

		An interrupt is requested at falling edge.
$\overline{\text{WUE15}}$ to $\overline{\text{WUE8}}$	Input	Maskable external interrupts An interrupt is requested at falling edge.

- IRQ status registers (ISR16, ISR)
- Keyboard matrix interrupt mask registers (KMIMRA, KMIMR6)
- Wake-up event interrupt mask register (WUEMR3)

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 ta

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 in control level 0 (no priority) 1: Corresponding interrupt source is in control level 1 (priority)

[Legend]
n: A to D

0	IICIn0	WDT_1	IIC_4, IIC_3	—

[Legend]]

n: A to D

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

5.3.2 Address Break Control Register (ABRKCR)

ABRKCR controls the address breaks. When both the CMF flag and BIE flag are set to 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 address specified by BARA to BARC is p [Clearing condition] When an exception handling is executed address break interrupt. [Setting condition] When an address specified by BARA to B prefetched while the BIE flag is set to 1.
6 to 1	—	All 0	R	Reserved These bits are always read as 0 and can modified.
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 A8 in the 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 A in the internal address bus.
0	—	0	R	Reserved This bit is always read as 0 and canno modified.

2	IRQ13SCA	0	R/W	of IRQn or ExIRQn input
1	IRQ12SCB	0	R/W	10: Interrupt request generated at risi
0	IRQ12SCA	0	R/W	IRQn or ExIRQn input
				11: Interrupt request generated at bot
				and rising edges of IRQn or ExIR
				(n = 15 to 12)

- ISCR16L

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

(n = 7 to 4)

- 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
4	IRQ2SCA	0	R/W	$\overline{\text{IRQn}}$ or $\overline{\text{ExIRQn}}$* input
3	IRQ1SCB	0	R/W	01: Interrupt request generated at fallin
2	IRQ1SCA	0	R/W	of $\overline{\text{IRQn}}$ or $\overline{\text{ExIRQn}}$* input
1	IRQ0SCB	0	R/W	10: Interrupt request generated at risin
0	IRQ0SCA	0	R/W	$\overline{\text{IRQn}}$ or $\overline{\text{ExIRQn}}$* input
				11: Interrupt request generated at both
				and rising edges of $\overline{\text{IRQn}}$ or $\overline{\text{ExIRQ}}$
				(n = 3 to 0)

Note: * $\overline{\text{ExIRQn}}$ stands for $\overline{\text{ExIRQ3}}$ or $\overline{\text{ExIRQ2}}$.

7 to 0	IRQ7E to IRQ0E	All 0	R/W	IRQn Enable (n = 7 to 0) The IRQn interrupt request is enabled bit is 1.

				• When interrupt exception handling executed when low-level detection and $\overline{\text{IRQn}}$ or $\overline{\text{ExIRQn}}$ input is high • When IRQn interrupt exception har executed when falling-edge, rising- both-edge detection is set (n = 15 to 8)

• 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 ISCR registers occurs [Clearing conditions] • When reading IRQnF flag when IR then writing 0 to IRQnF flag • When interrupt exception handling executed when low-level detection and $\overline{\text{IRQn}}$ or $\overline{\text{ExIRQn}}$* input is high • When IRQn interrupt exception har executed when falling-edge, rising- both-edge detection is set (n = 7 to 0)

Note: * $\overline{\text{ExIRQn}}$ stands for $\overline{\text{ExIRQ7}}$ to $\overline{\text{ExIRQ2}}$.

				input interrupt request (KIN15 to KIN8 0: Enables a key-sensing input interru 1: Disables a key-sensing input interr request

- KMIMR6

Bit	Bit Name	Initial Value	R/W	Description
7 to 0	KMIM7 to KMIM0	All 1	R/W	Keyboard Matrix Interrupt Mask These bits enable or disable a k input interrupt request (KIN7 to KIN0) 0: Enables a key-sensing input interru 1: Disables a key-sensing input interr request

- WUEMR3

Bit	Bit Name	Initial Value	R/W	Description
7 to 0	WUEM15 to WUEM8	All 1	R/W	Wake-Up Event Interrupt Mask These bits enable or disable a wake-u input interrupt request (WUE15 to WU 0: Enables a wake-up event input inte request 1: Disables a wake-up event input int request

IRQ15 to IRQ0 Interrupts: Interrupts IRQ15 to IRQ0 are requested by an input signal t
$\overline{\text{IRQ15}}$ to $\overline{\text{IRQ0}}$ or pins $\overline{\text{ExIRQ15}}$ to $\overline{\text{ExIRQ2}}$. 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 $\overline{\text{ExIRQ2}}$.
- 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.

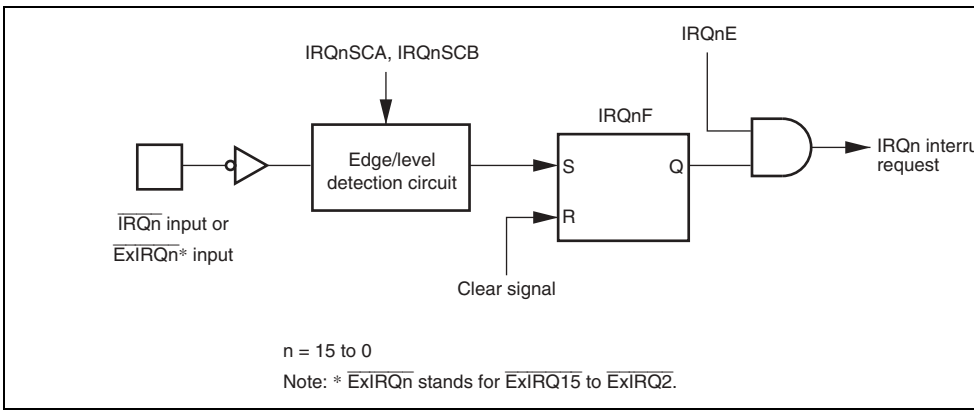


Figure 5.2 Block Diagram of Interrupts IRQ15 to IRQ0

The detection of KIN15 to KIN0 and WUE15 to WUE8 interrupts does not depend on w
relevant pin has been set for input or output. However, when a pin is used as an external
input pin, clear the corresponding port DDR to 0 so that it is not used as an I/O pin for a
function.

A block diagram of interrupts KIN15 to KIN0 and WUE15 to WUE8 is shown in figure

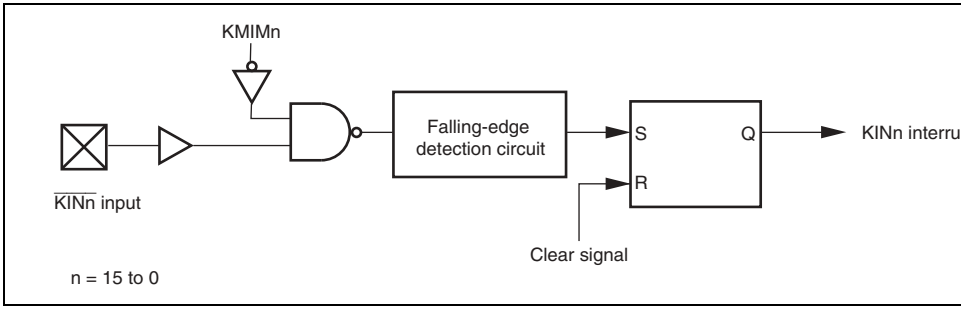


**Figure 5.3 Block Diagram of Interrupts KIN15 to KIN0 and WUE15 to WU
(Example of KIN15 to KIN0)**

Origin of Interrupt Source	Name	Vector Number	Vector Address Advanced Mode	ICR	F
External pin	NMI	7	H'00001C	—	
	IRQ0	16	H'000040	ICRA7	
	IRQ1	17	H'000044	ICRA6	
	IRQ2 IRQ3	18 19	H'000048 H'00004C	ICRA5	
	IRQ4 IRQ5	20 21	H'000050 H'000054	ICRA4	
	IRQ6 IRQ7	22 23	H'000058 H'00005C	ICRA3	
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	—	
External pin	KIN7 to KIN0 KIN15 and KIN8 WUE15 to WUE8	30 31 33	H'000078 H'00007C H'000084	—	
TMR_X	CMIAX (Compare match A) CMIBX (Compare match B) OVIX (Overflow) ICIX (Input capture)	44 45 46 47	H'0000B0 H'0000B4 H'0000B8 H'0000BC	ICRB4	

	IRQ11	59	H'0000EC	
	IRQ12	60	H'0000F0	ICRD6
	IRQ13	61	H'0000F4	
	IRQ14	62	H'0000F8	
	IRQ15	63	H'0000FC	
TMR_0	CMIA0 (Compare match A)	64	H'000100	ICRB3
	CMIB0 (Compare match B)	65	H'000104	
	OVI0 (Overflow)	66	H'000108	
TMR_1	CMIA1 (Compare match A)	68	H'000110	ICRB2
	CMIB1 (Compare match B)	69	H'000114	
	OVI1 (Overflow)	70	H'000118	
TMR_Y	CMIAY (Compare match A)	72	H'000120	ICRB1
	CMIBY (Compare match B)	73	H'000124	
	OVIY (Overflow)	74	H'000128	
IIC_2	IICI2	76	H'000130	ICRC2
IIC_3	IICI3	78	H'000138	
SCI_0	ERI0 (Reception error 0)	80	H'000140	ICRC7
	RXI0 (Reception completion 0)	81	H'000144	
	TXI0 (Transmission data empty 0)	82	H'000148	
	TEI0 (Transmission end 0)	83	H'00014C	
SCI_1	ERI1 (Reception error 1)	84	H'000150	ICRC6
	RXI1 (Reception completion 1)	85	H'000154	
	TXI1 (Transmission data empty 1)	86	H'000158	
	TEI1 (Transmission end 1)	87	H'00015C	
SCI_2	ERI2 (Reception error 2)	88	H'000160	ICRC5
	RXI2 (Reception completion 2)	89	H'000164	
	TXI2 (Transmission data empty 2)	90	H'000168	
	TEI2 (Transmission end 2)	91	H'00016C	

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

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

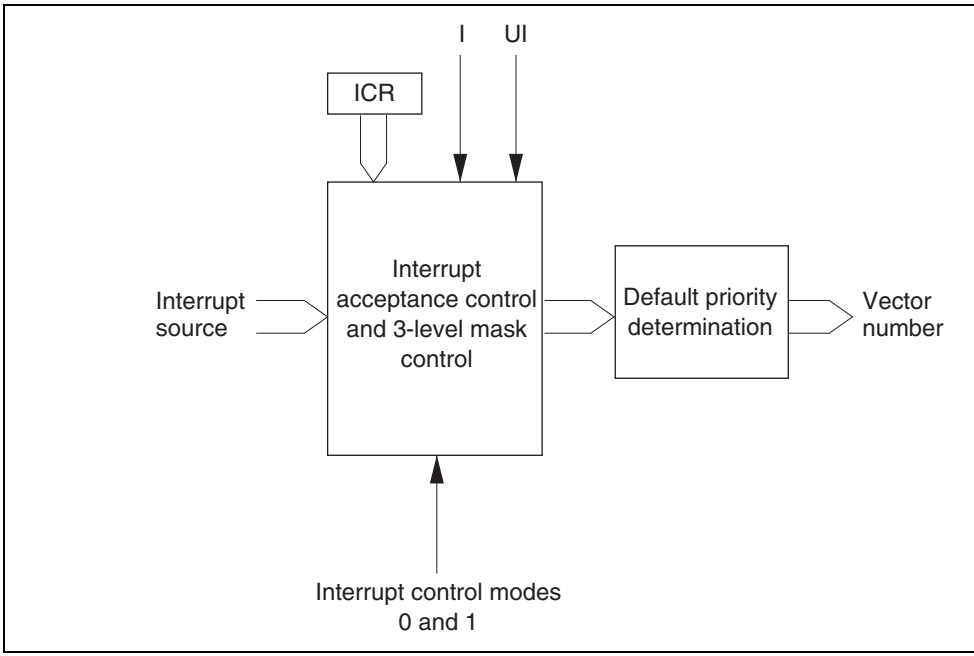


Figure 5.4 Block Diagram of Interrupt Control Operation

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

[Legend]

*: Don't care

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

Interrupt sources with a lower priority than the accepted interrupt source are held pendin

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

5.6.1 Interrupt Control Mode 0

In interrupt control mode 0, interrupts other than NMI are masked by ICR and the I bit of in the CPU. Figure 5.5 shows a flowchart of the interrupt acceptance operation.

1. If an interrupt source occurs when the corresponding interrupt enable bit is set to 1, ar interrupt request is sent to the interrupt controller.
2. According to the interrupt control level specified in ICR, the interrupt controller accep interrupt request with interrupt control level 1 (priority), and holds pending an interru with interrupt control level 0 (no priority). If several interrupt requests are issued, an i 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 accep the interrupt controller, and other interrupt requests are held pending. If the I bit is cle any interrupt request is accepted. KIN, WUE, and EVENTI interrupts are enabled or d by the I bit.
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. Next, the I bit in CCR is set to 1. This masks all interrupts except for NMI and addres 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.

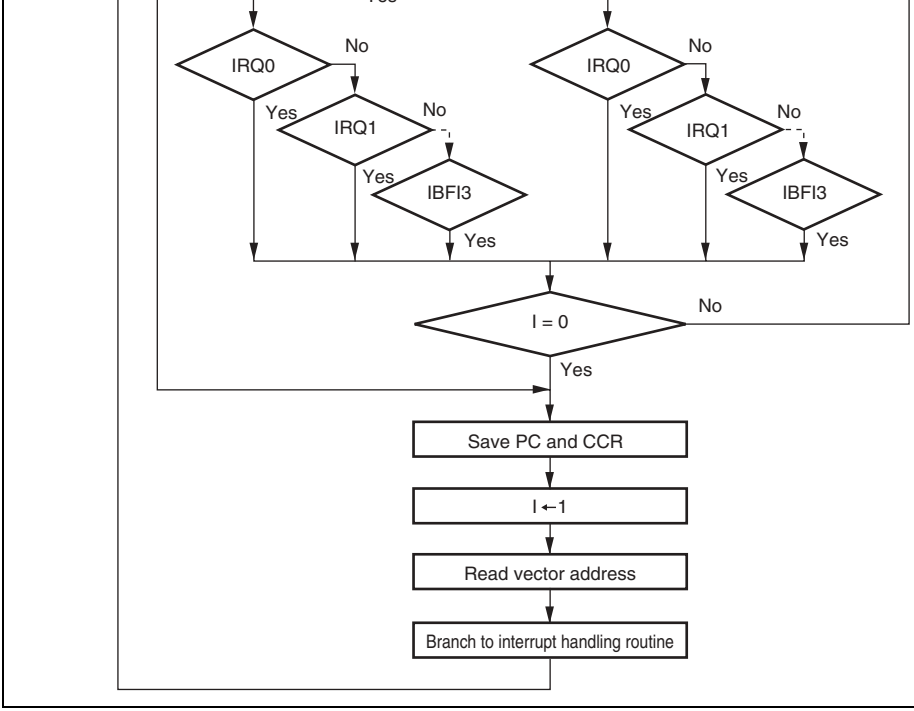


Figure 5.5 Flowchart of Procedure up to Interrupt Acceptance in Interrupt Contr

set to interrupt control level 1, and other interrupts are set to interrupt control level 0) is s
below. Figure 5.6 shows a state transition diagram.

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

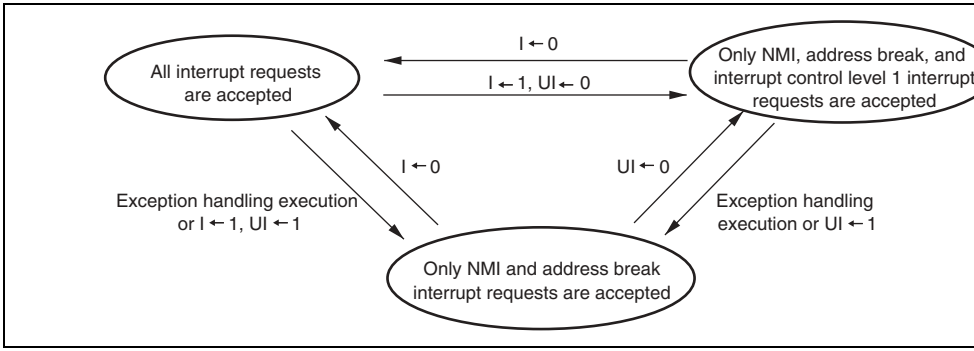


Figure 5.6 State Transition in Interrupt Control Mode 1

Figure 5.7 shows a flowchart of the interrupt acceptance operation.

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 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. The I and UI bits in CCR are set to 1. This masks all interrupts except for NMI and a break interrupts.
7. The CPU generates a vector address for the accepted interrupt and starts execution o interrupt handling routine at the address indicated by the contents of the vector addre vector table.

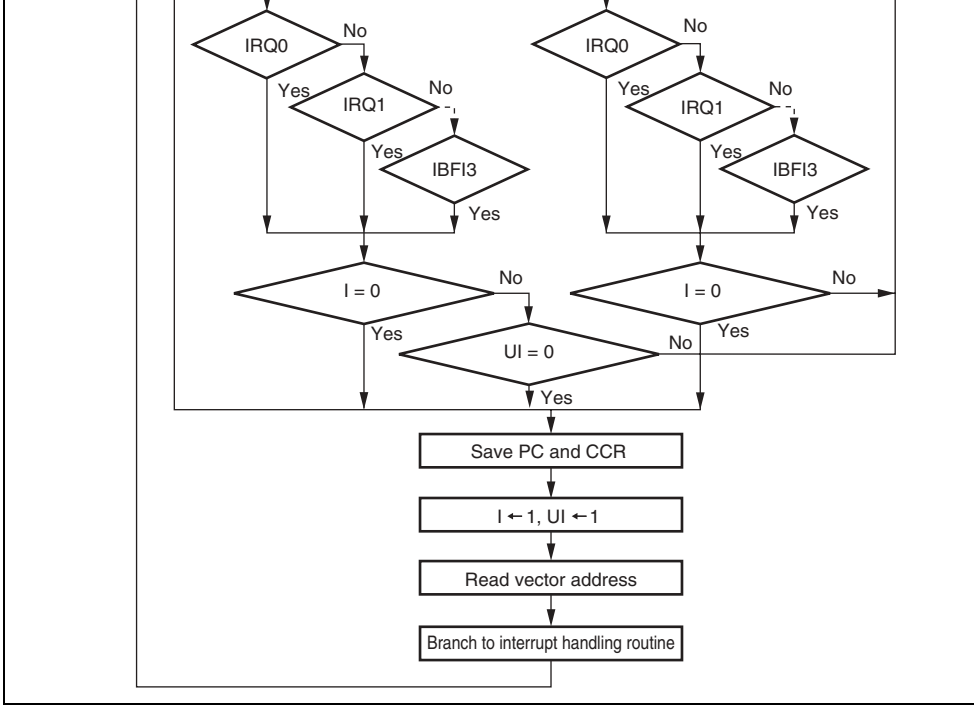


Figure 5.7 Flowchart of Procedure Up to Interrupt Acceptance in Interrup Control Mode 1

5.6.3 Interrupt Exception Handling Sequence

Figure 5.8 shows the interrupt exception handling sequence. The example shown is for th where interrupt control mode 0 is set in advanced mode, and the program area and stack a in on-chip memory.

Figure 5.8 Interrupt Exception Handling

5	Instruction fetch*[3]	2·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 vec

Table 5.8 Number of States in Interrupt Handling Routine Execution Status

	Object of Access				
		External Device			
		8-Bit Bus		16-Bit B	
Symbol	**Internal Memory**	**2-State Access**	**3-State Access**	**2-State Access**	**3-State A**
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.

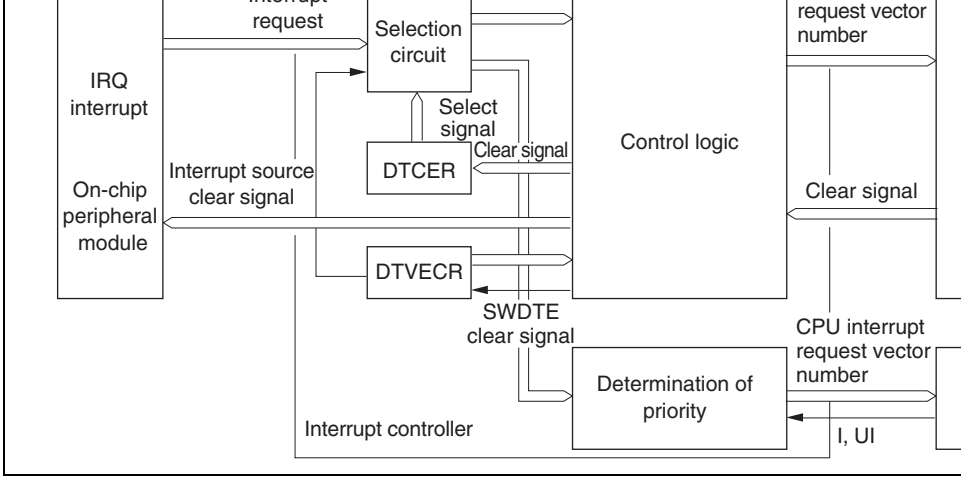


Figure 5.9 Interrupt Control for DTC

The interrupt controller has three main functions in DTC control.

Selection of Interrupt Source: It is possible to select DTC activation request or CPU in request with the DTCE bit of DTCERA to DTCERE in the DTC. After a DTC data trans DTCE bit can be cleared to 0 and an interrupt request sent to the CPU in accordance wit specification of the DISEL bit of MRB in the DTC. When the DTC performs the specifi of data transfers and the transfer counter reaches 0, following the DTC data transfer the is cleared to 0 and an interrupt request is sent to the CPU.

Determination of Priority: The DTC activation source is selected in accordance with t priority order, and is not affected by mask or priority levels. See section 7.5, Location o Information and DTC Vector Table, for the respective priorities.

0	*	×	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.

*: Don’t care

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.

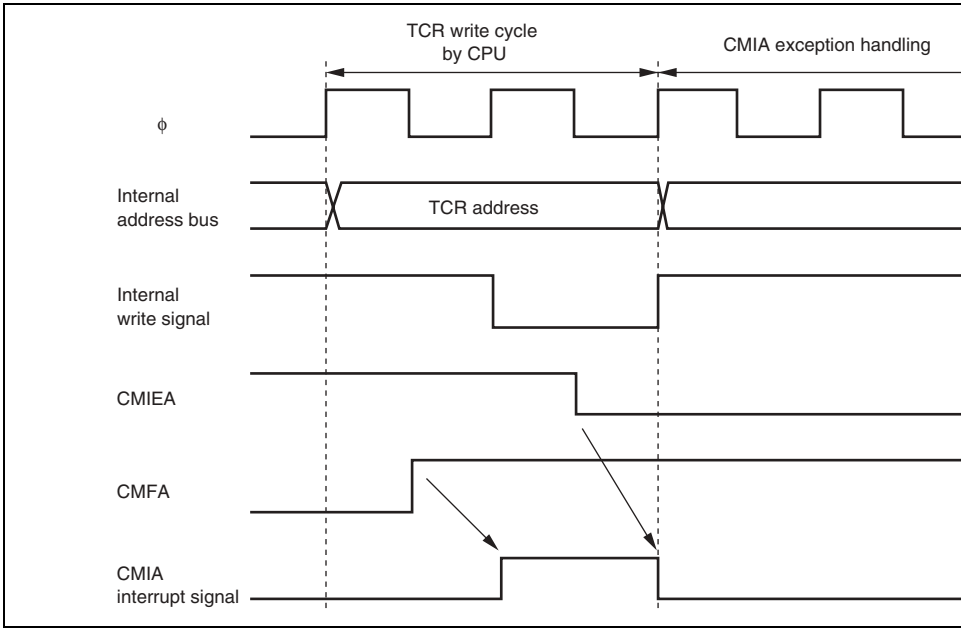


Figure 5.10 Conflict between Interrupt Generation and Disabling

is not accepted until the move is completed.

With the EEPMOV.W instruction, if an interrupt request is issued during the transfer, int exception handling starts at a break in the transfer cycle. The PC value saved on the stack case is the address of the next instruction. Therefore, if an interrupt is generated during ex 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 IS should be read after a reset, and then write 0 in IRQnF (n = 15 to 0).

Address-data multiplex extended mode: Multiplex extension (when the ADMUXE bit in SYSCR2 is 1)

- 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 extended

  A CP extended area can be set and controlled independently of basic extended areas.

- Address pin reduction

  In normal extended mode:

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

  A CP extended area (8 kbytes, basic mode) from H'FFC000 to H'FFDFFF can be sel using 13 address pins and the $\overline{\text{CPCS1}}$ signal.

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

  In address-data multiplex extended mode:

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

H'F80000 to H'F8FFFF	64 kbytes	256-kbyte extended area
H'FFC000 to H'FFDFFF	8 kbytes	CP extended area
H'FFF000 to H'FFF7FF	2 kbytes	IOS extended area

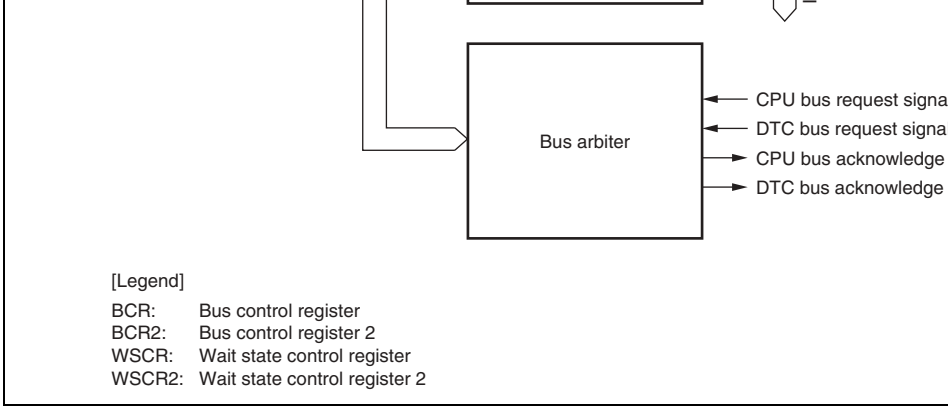
  These areas can be selected using 8 pins or 16 pins, which is a total of address pins a input/output pins.

- Control address hold signal and aria select signal polarity

  The output polarity of $\overline{\text{IOS}}$, $\overline{\text{CS256}}$, $\overline{\text{CPCS1}}$, and $\overline{\text{AH}}$ can be inverted by the PNCCS PNCAH bits in 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 and DTC.



[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).
$\overline{\text{CPCS1}}$	Output	Chip select signal indicating that the CP extended ar being accessed (when the CPCSE bit in BCR2 is 1).
$\overline{\text{CS256}}$	Output	Chip select signal indicating that the 256-kbyte exter 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 enabled.
$\overline{\text{LWR}}$	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 enabled.
$\overline{\text{WAIT}}$	Input	Wait request signal when accessing the external spa
$\overline{\text{AH}}$	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.

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 idle cyc between successive external read and external write 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 address spa 0: Basic bus interface 1: Burst ROM interface When the CS256E bit in SYSCR and the CPCSE bit i are set to 1, burst ROM interface cannot be selected 256 -kbyte extended area and CP extended area.
4	BRSTS1	1	R/W	Valid only in the normal extended mode. Burst Cycle Select 1 Selects the number of states in the burst cycle of the ROM interface. 0: 1 state 1: 2 states

See table 6.15.

6.3.2 Bus Control Register 2 (BCR2)

BCR2 is used to specify the access mode for the CP extended area.

Bit	Bit Name	Initial Value	R/W	Description
7, 6	—	All 0	R/W	Reserved The initial value should not be changed.
5	ABWCP	1	R/W	CP Extended Area Bus Width Control Selects the bus width for access to the CP extended when the CPCSE bit is set to 1 0: 16-bit bus 1: 8-bit bus
4	ASTCP	1	R/W	CP Extended Area Access State Control Selects the number of states for access to the CP ext area when the CPCSE bit is set to 1. This bit also ena disables wait-state insertion. [ADMXE = 0] Normal extension 0: 2-state access space. Wait state insertion disabled 1: 3-state access space. Wait state insertion enabled [ADMXE = 1] Address-data multiplex extension 0: 2-state data access space. Wait state insertion disa 1: 3-state data access space. Wait state insertion ena

				external extended area access.
1	—	1	R/W	Reserved The initial value should not be changed.
0	CPCSE	0	R/W	CP Extended Area Enable Selects the extended area to be accessed. 0: External address space 1: CP extended area

				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 256-kb extended area when the CS256E bit in SYSCR is set This bit also enables or disables wait-state insertion. [ADMXE = 0] Normal extension 0: 2-state access space. Wait state insertion disabled 1: 3-state access space. Wait state insertion enabled [ADMXE = 1] Address-data multiplex extension 0: 2-state data access space. Wait state insertion dis 1: 3-state data access space. Wait state insertion ena
5	ABW	1	R/W	Basic Extended Area Bus Width Control Selects the bus width for access to the basic extende 0: 16-bit bus 1: 8-bit bus When the CS256E bit in SYSCR and the CPCSE bit are set to 1, this bit setting is ignored in 256-kbyte ex area access and CP extended area access.

				are set to 1, this bit setting is ignored in 256-kbyte e area access and CP extended area access.
3 2	WMS1 WMS0	0 0	R/W R/W	Basic Extended Area Wait Mode Select 1 and 0 Selects the wait mode for access to the basic exten 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 and the CPCSE bi are set to 1, this bit setting is ignored in 256-kbyte e area access and CP extended area access.
1 0	WC1 WC0	1 1	R/W R/W	Basic Extended Area Wait Count 1 and 0 Selects the number of program wait states to be ins when the basic extended area is accessed when the is set to 1. The program wait state is only inserted into dat 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 and the CPCSE bi are set to 1, this bit setting is ignored in 256-kbyte e area access and CP extended area access.

				1: Wait disabled mode
6 5	WC11 WC10	1 1	R/W R/W	256-kbyte Extended Area Wait Count 1 and 0 Selects the number of program wait states to be inse the data cycle for access to the 256-kbyte extended a when the CS256E bit in SYSCR and the AST256 bit i 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 3	WMS21 WMS20	0 0	R/W R/W	CP Extended Area Wait Mode Select 1 and 0 Selects the wait mode for access to the CP extended when the CPCSE and ASTCP bits in BCR2 are set to 00: Program wait mode 01: Wait disabled mode 10: Pin wait mode 11: Pin auto-wait mode

				011: 3 program wait states are inserted 100: (Setting prohibited) 101: (Setting prohibited) 110: (Setting prohibited) 111: (Setting prohibited)

- When ADMXE = 1

Bit	Bit Name	Initial Value	R/W	Description
2	WC22	1	R/W	Address-Data Multiplex Extended Area Address Cyc Count 2 Selects the number of program wait states to be ins into the address cycle for access to the address-dat multiplex extended area. 0: Program wait state is not inserted 1: 1 program wait state is inserted in the address cy
1	WC21	1	R/W	CP Extended Area Data Cycle Wait Count 1 and 0
0	WC20	1	R/W	Selects the number of program wait states to be ins the data cycle for access to the CP extended area w CPCSE and ASTCP bits in BCR2 are set to 1. 00: Program wait state is not inserted in the data cy 01: 1 program wait state is inserted in the data cycle 10: 2 program wait states are inserted in the data cy 11: 3 program wait states are inserted in the data cy

(b) Number of Access States: Two or three access states can be selected via the AST an AST256 bits in WSCR, and the ASTCP bit in BCR2. When the 2-state access space is de 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 as a 3-state access space by the AST bit in WSCR, the wait mode and the number of prog states to be inserted automatically is selected by the WMS1, WMS0, WC1, and WC0 bits WSCR. From 0 to 3 program wait states can be selected.

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.

When the CP extended area is specified as a 3-state access space by the ASTCP bit in BC wait mode and the number of program wait states to be inserted automatically is selected WMS21, WMS20, WC21, and WC20 bits in WSCR2. From 0 to 3 program wait states ca 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 6.6 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'FF0800 to H'FFBFFF (46 kbytes)	Δ When RAME = 0, used as basic extended area.	—
H'FFC000 to H'FFDFFF (8 kbytes) CP extended area	Δ When CPCSE = 0, used as basic extended area.	When CPCSE = 1, $\overline{CP}$ output in the CP exter and address pins A12 used.
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, $\overline{IOS}$ 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 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 a 256-kbyte extended area or CP extende

1	0	0	Burst ROM interface*	Used as burst ROM interface	Used as burst ROM interface
		1	ABW, AST, WMS0, WC1, WC0, BRSTS1, BRSTS0		ABWCP, ASTCP, WMS20, WC21, W
	1	0		ABW256, AST256, WMS10, WC11, WC10	Same as when CS
		1			

Note: * In the burst ROM interface, the bus width is specified by the ABW bit in WSCR 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.

					1			3
1	0	*	*	*	*	8	2	0
	1	0	1	*	*	8	3	0
		Other than WMS1 = 0 and WMS0 = 1		0	0		3	0
					1			1
				1	0			2
					1			3

[Legend]

*: Don’t care

1	0	*	*	*	8	2	0
	1	1	*	*	8	3	0
		0	0	0		3	0
				1			1
			1	0			2
				1			3

[Legend]

*: Don’t care

					1			3
1	0	*	*	*	*	8	2	0
	1	0	1	*	*	8	3	0
		Other than WMS21 = 0 and WMS20 = 1		0	0		3	0
					1			1
				1	0			2
					1			3

[Legend]

*: Don't care

wait mode and the number of program wait states to be inserted automatically is selected
WMS1, WMS0, WC1, and WC0 bits in WSCR. Zero or one program wait state can be in
into address cycle. From zero to three program wait states can be selected for data cycle.

ii) 256-kbyte Extended Area

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. Zero or one program wait sta
inserted into address cycle. From zero to three program wait states can be selected for dat

iii) CP Extended Area

When the CP extended area is specified as a 3-state access space by the ASTCP bit in BC
wait mode and the number of program wait states to be inserted automatically is selected
WMS21, WMS20, WC22, WC21, and WC20 bits in WSCR2. Zero or one program wait
be inserted into address cycle. From zero to three program wait states can be selected for
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 6.7 to 6.14 show address-data multiplex address space and the bus specifications f
basic bus interface of each area.

H'FFA0000 to H'FFAFFFF (64 kbytes)		
256-kbyte extended area H'FB0000 to H'FBFFFF (64 kbytes)	—	No condition
H'FC0000 to H'FFBFFF (240 kbytes)	—	No condition
CP extended area H'FFC000 to H'FFDFFF (8 kbytes)	O	When CPCSE = 1, $\overline{\text{CPCS1}}$ is output, and addres AD15 to AD0 or AD7 to AD0 are used.
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.

	1		ABWCP, ASTCP WMS21, WMS2 WC20
1	0	ABW256, AST256, WMS10, WC11, WC10	Same as when C = 0
	1		

Table 6.9 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 6.10 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	Prog State
0	—	—	—	2	0
1	1	—	—	3	0
	0	0	0	3	0
			1		1
		1	0		2
			1		3

Table 6.13 Bus Specifications for CP Extended Area/Multiplex Bus Interface (Address Cycle)

ASTCP	WMS21	WMS20	WC22	WC21	WC20	Number of Access States	N P W S
—	—	—	0	—	—	2	0
			1	—	—		1

6.4.2 Advanced Mode

The external address space (H'FFF000 to H'FFF7FF) can be accessed by specifying the $\overline{A}$
pin as an I/O strobe pin. The 256-kbyte extended area (H'F80000 to H'FBFFFF) and CP e
area (H'FFC000 to H'FFDFFF) can be accessed by the $\overline{CS256}$ pin and $\overline{CPCS1}$ pin functio
respectively.

The external address space is initialized as the basic bus interface and a 3-state access spa
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 reserv
are enabled when the RAME bit in SYSCR is set to 1, and disabled when the RAME bit i
to 0. Addresses H'FF0800 to H'FFBFFF, H'FFE080 to H'FFEFFF, and H'FFFF00 to H'FF
the on-chip RAM area and its reserved area are always specified as the external address s

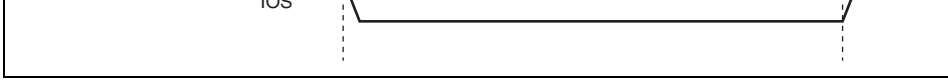


Figure 6.2 $\overline{\text{IOS}}$ Signal Output Timing

Enabling or disabling $\overline{\text{IOS}}$ signal output is performed by the IOSE bit in SYSCR. In the
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 th
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
as shown in table 6.15.

Table 6.15 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	(Ini

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 A
lower data bus (D7 to D0/AD7 to AD0) is used when the external address space is access
according to the bus specifications for the area being accessed (8-bit access space or 16-b
space) and the data size.

(1) 8-Bit Access Space: Figure 6.3 illustrates data alignment control for the 8-bit access s
With the 8-bit access space, the upper data bus (D15 to D8/AD15 to AD8) is always used
accesses. The amount of data that can be accessed at one time is one byte: a word access
performed as two byte accesses, and a longword access, as four byte accesses.

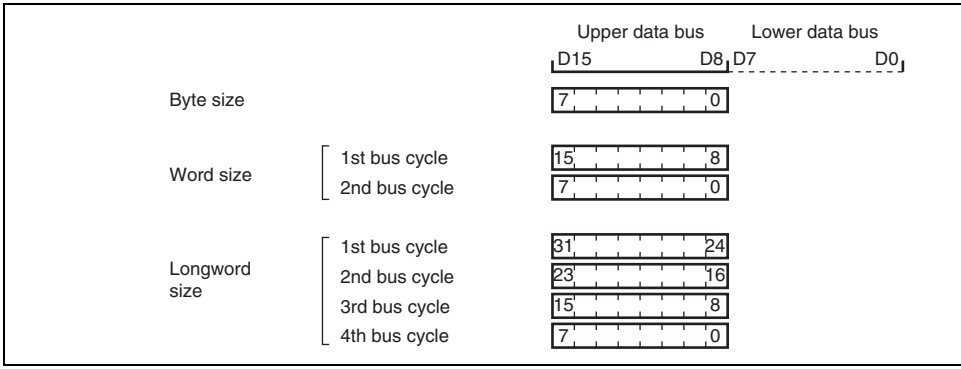


Figure 6.3 Access Sizes and Data Alignment Control (8-bit Access Space)

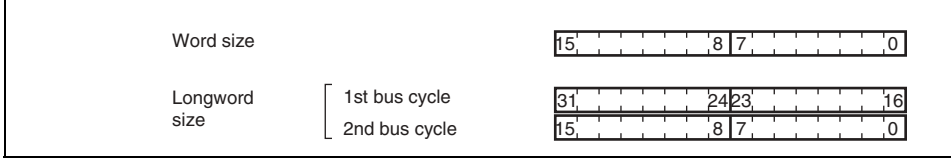


Figure 6.4 Access Sizes and Data Alignment Control (16-bit Access Space

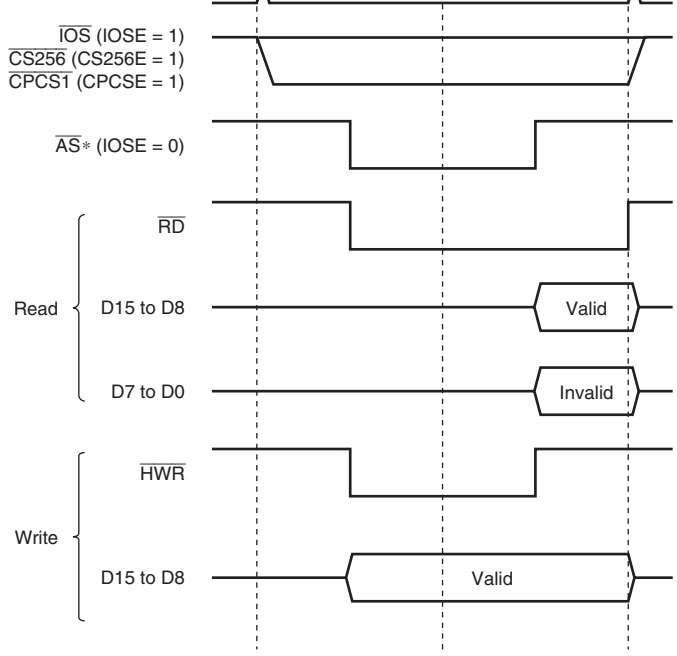
space		Write	—	$\overline{HWR}$		Ports o
16-bit access space	Byte	Read	Even	$\overline{RD}$	Valid	Invalid
			Odd		Invalid	Valid
		Write	Even	$\overline{HWR}$	Valid	Undefi
			Odd	$\overline{LWR}$	Undefined	Valid
	Word	Read	—	$\overline{RD}$	Valid	Valid
		Write	—	$\overline{HWR}$, $\overline{LWR}$	Valid	Valid

[Legend]

Undefined: Undefined data is output.

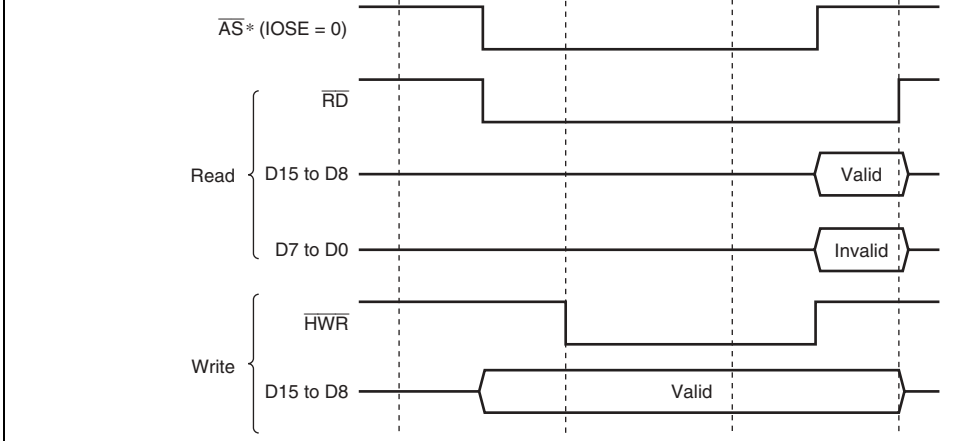
Invalid: Input state with the input value ignored.

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



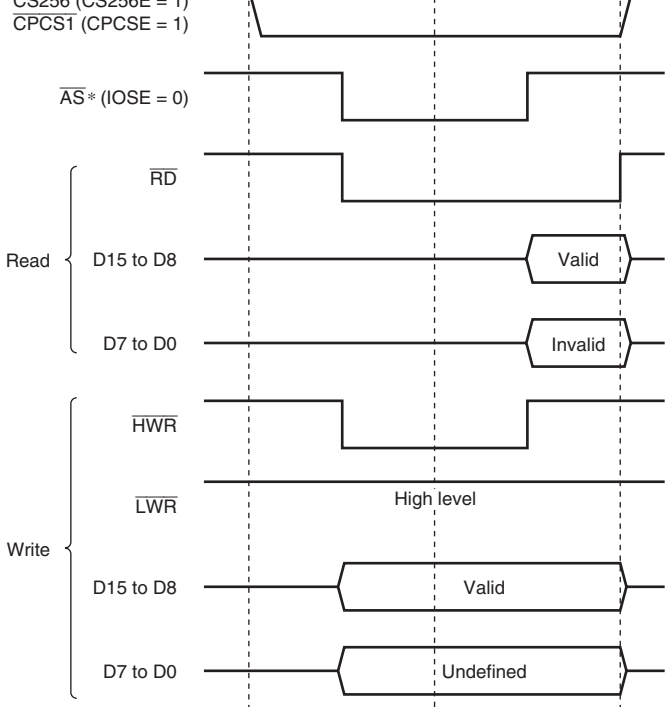
Note: * For external address space access, this signal is not output when the 256-kbyte expansion are
is accessed with CS256E = 1 and when the CP expansion area is accessed with CPCSE = 1.

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



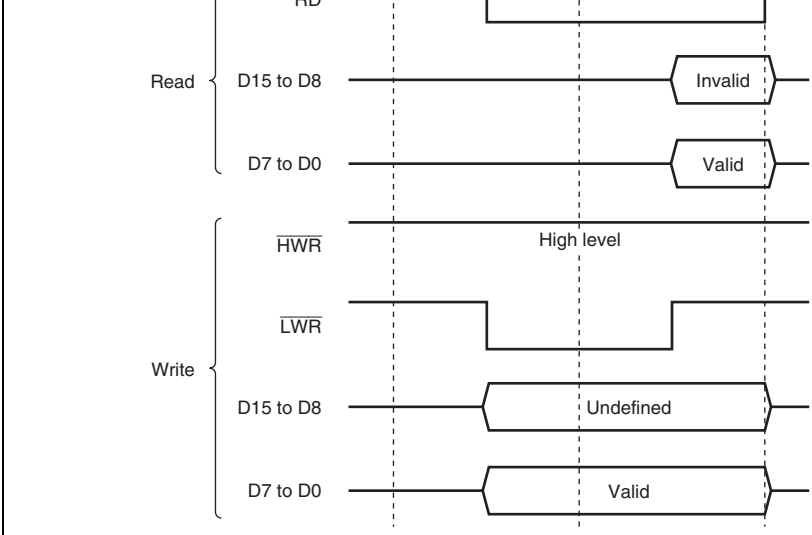
Note: * For external address space access, this signal is not output when the 256-kbyte expansion area is accessed with CS256E = 1 and when the CP expansion area is accessed with CPCSE = 1.

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



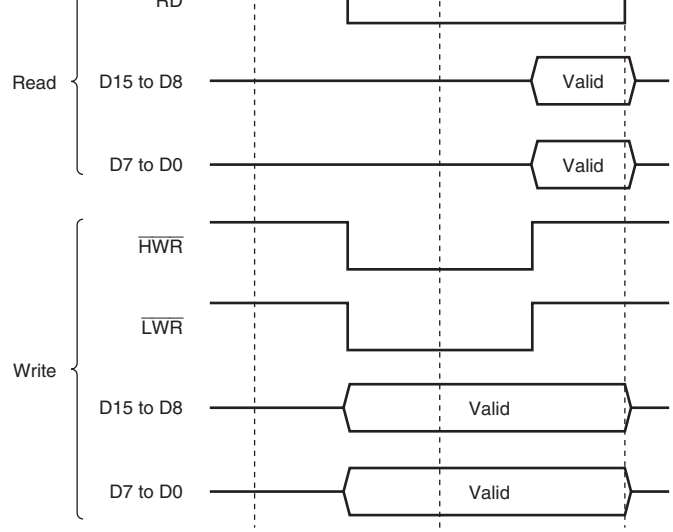
Note: * For external address space access, this signal is not output when the 256-kbyte expansion are
is accessed with CS256E = 1 and when the CP expansion area is accessed with CPCSE = 1.

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



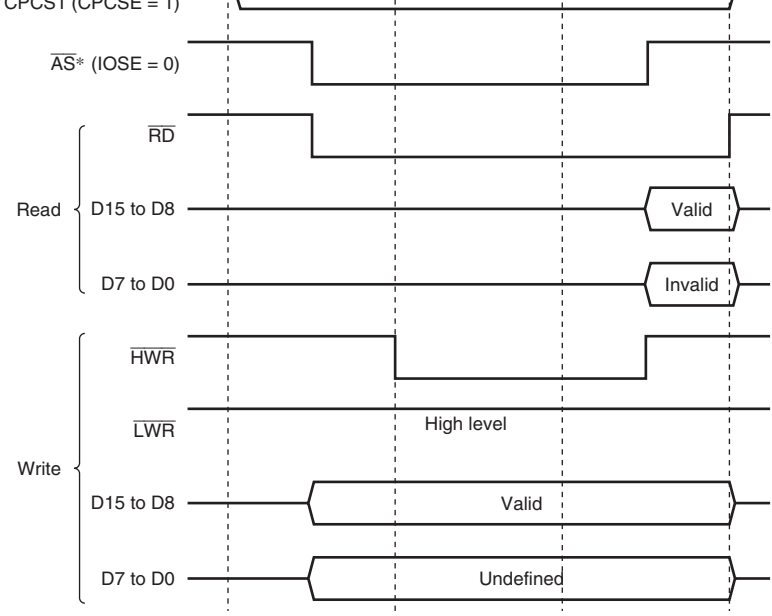
Note: * For external address space access, this signal is not output when the 256-kbyte expansion area is accessed with CS256E = 1 and when the CP expansion area is accessed with CPCSE = 1.

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



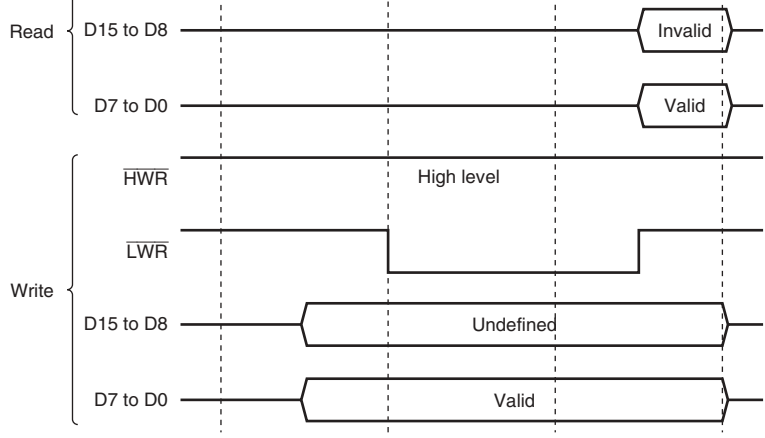
Note: * For external address space access, this signal is not output when the 256-kbyte expansion are is accessed with CS256E = 1 and when the CP expansion area is accessed with CPCSE = 1.

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



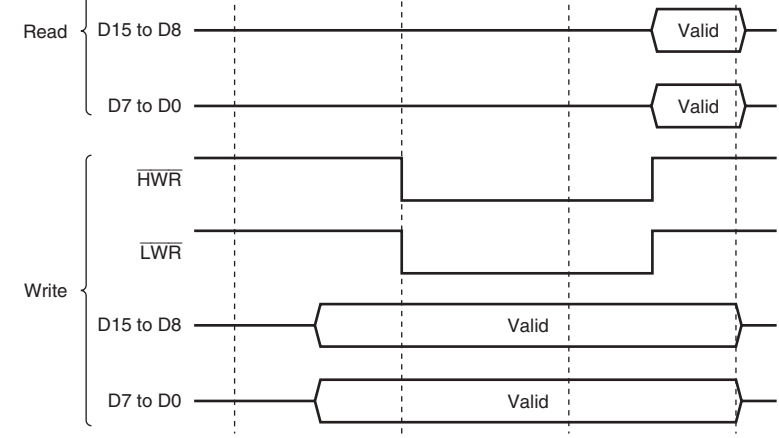
Note: * For external address space access, this signal is not output when the 256-kbyte expansion area is accessed with CS256E = 1 and when the CP expansion area is accessed with CPCSE = 1.

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 expansion are
is accessed with CS256E = 1 and when the CP expansion area is accessed with CPCSE = 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 expansion area is accessed with CS256E = 1 and when the CP expansion area is accessed with CPCSE = 1.

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

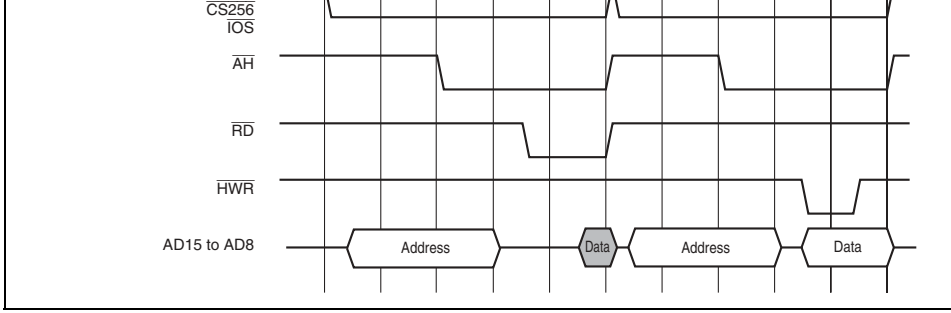


Figure 6.13 Bus Timing for 8-Bit, 2-State Access Space

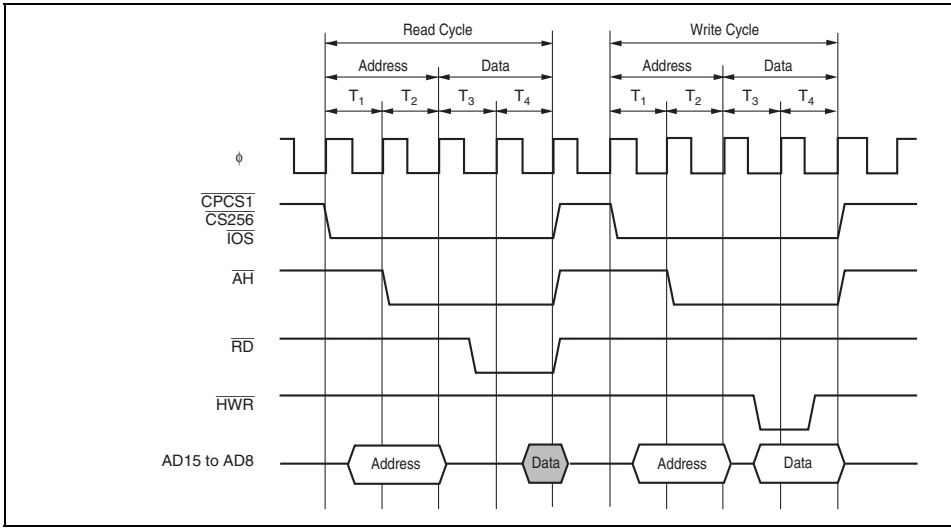


Figure 6.14 Bus Timing for 8-Bit, 2-State Access Space

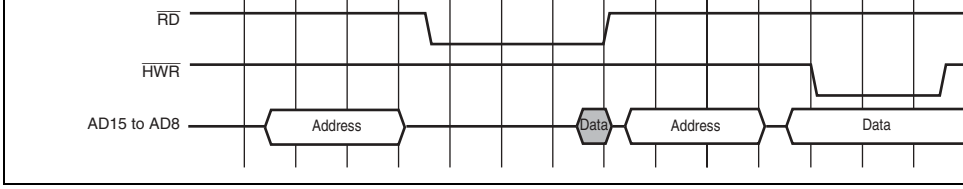


Figure 6.15 Bus Timing for 8-Bit, 3-State Access Space

(3) 16-Bit, 2-State Data Access Space: Figures 6.16 to 6.21 show bus timings for a 16-b access space. When a 16-bit access space is accessed, the upper half (AD15 to AD8) of th bus is used for even addresses, and the lower half (AD7 to AD0) for odd addresses. Wait cannot be inserted.

Figure 6.16 Bus Timing for 16-Bit, 2-State Access Space (1) (Even Byte Acc

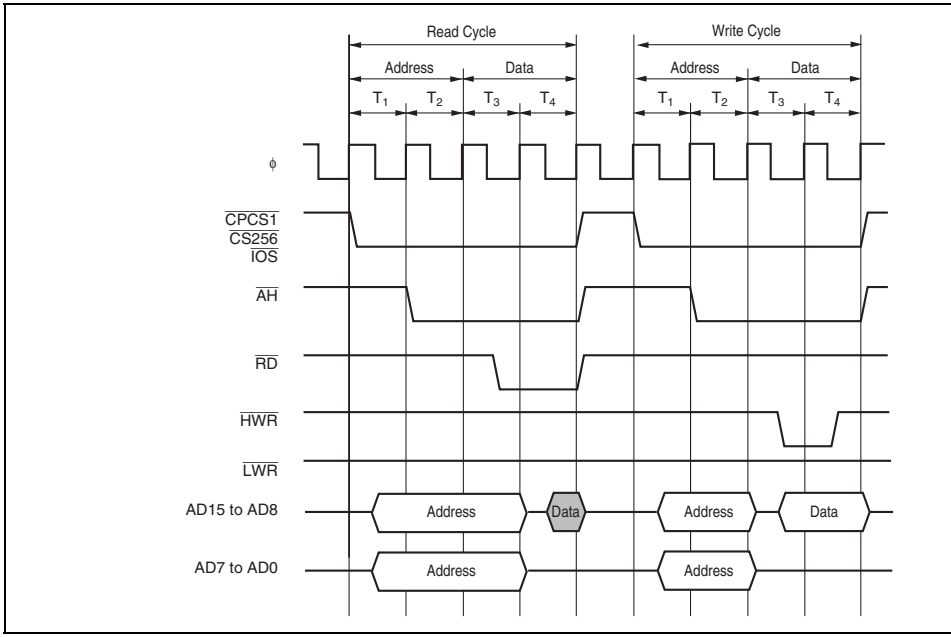


Figure 6.17 Bus Timing for 16-Bit, 2-State Access Space (2) (Even Byte Acc

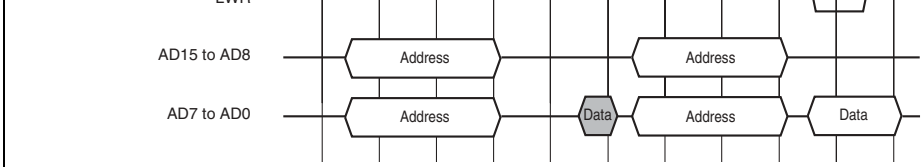


Figure 6.18 Bus Timing for 16-Bit, 2-State Access Space (3) (Odd Byte Acces

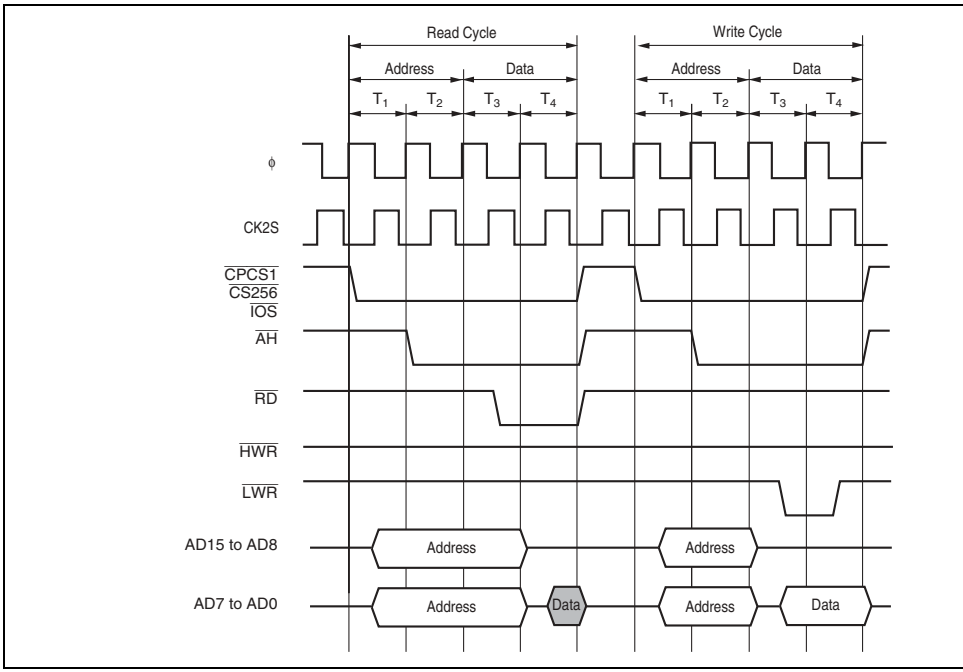


Figure 6.19 Bus Timing for 16-Bit, 2-State Access Space (4) (Odd Byte Acces

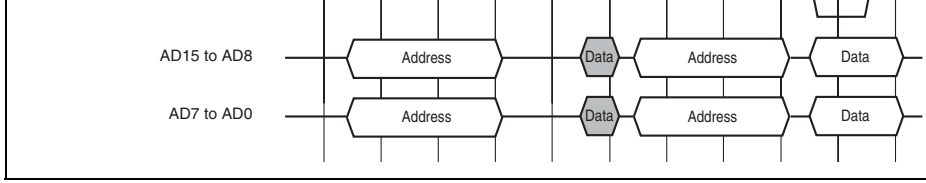


Figure 6.20 Bus Timing for 16-Bit, 2-State Access Space (5) (Word Acces

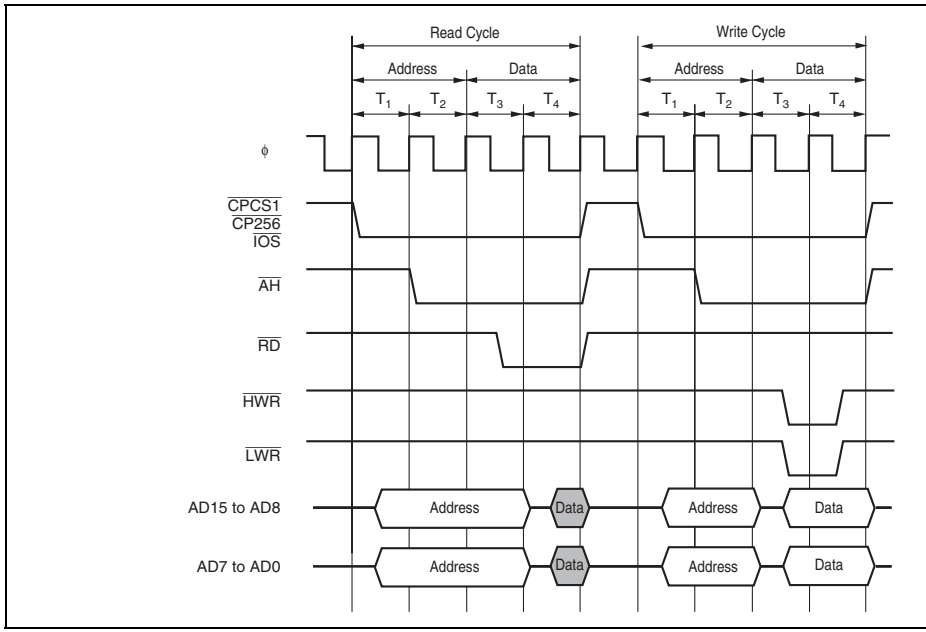


Figure 6.21 Bus Timing for 16-Bit, 2-State Access Space (6) (Word Acces

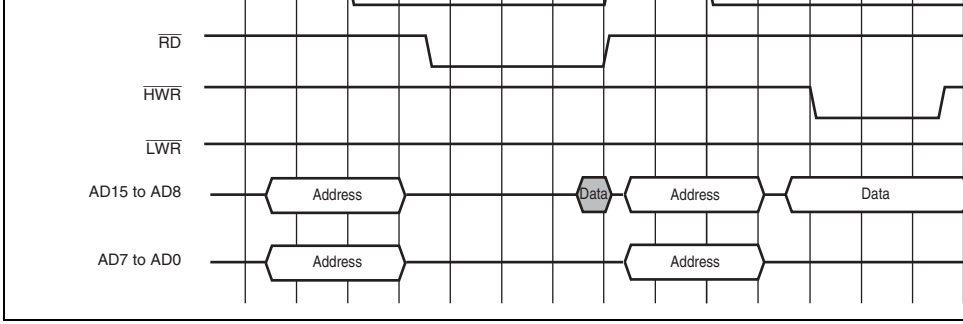


Figure 6.22 Bus Timing for 16-Bit, 3-State Access Space (1) (Even Byte Acce

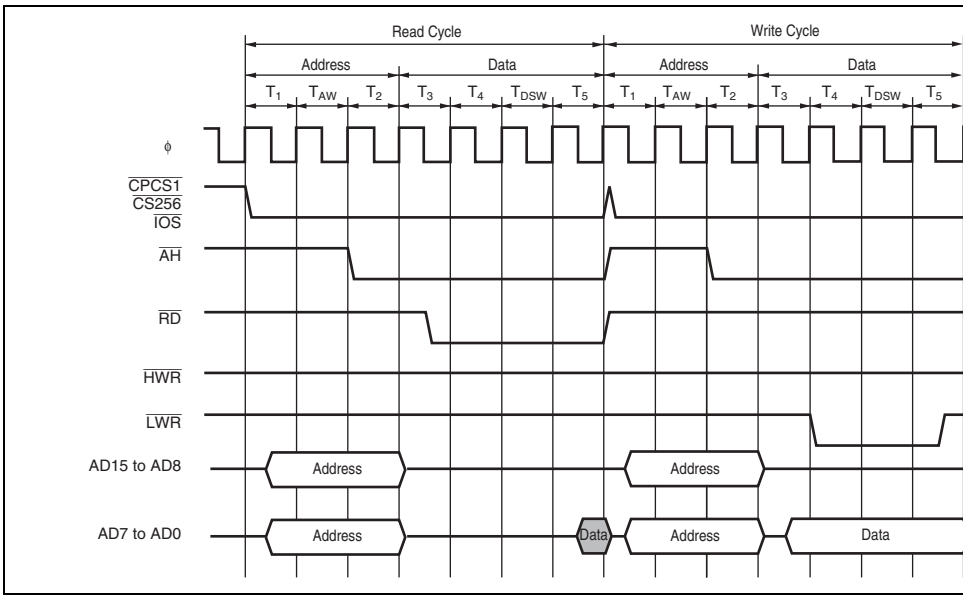


Figure 6.23 Bus Timing for 16-Bit, 3-State Access Space (2) (Odd Byte Acces

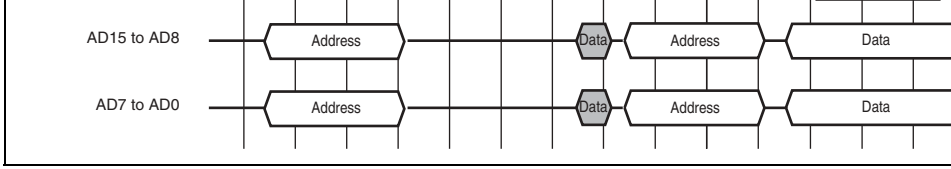


Figure 6.24 Bus Timing for 16-Bit, 3-State Access Space (3) (Word Acces

6.5.5 Wait Control

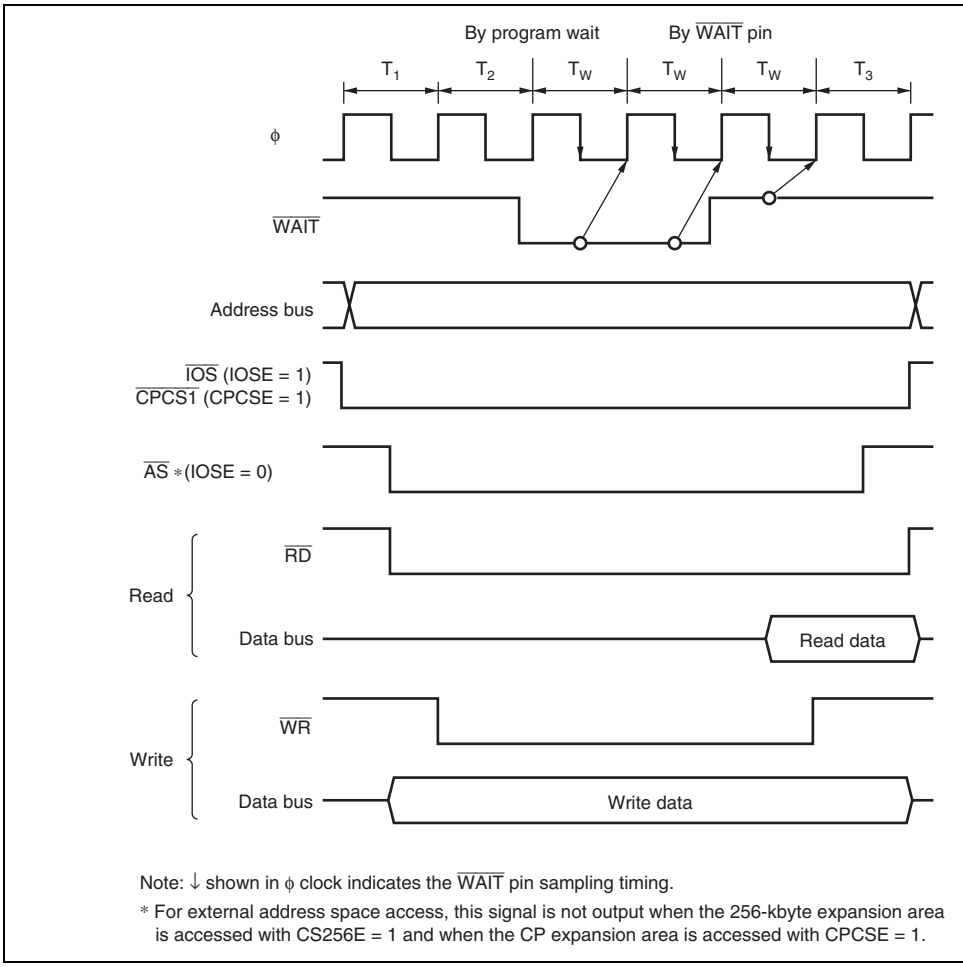
When accessing the external address space, this LSI can extend the bus cycle by insertin more wait states (T_W). There are three ways of inserting wait states: Program wait inserti wait insertion using the $\overline{\text{WAIT}}$ pin, and the combination of program wait and the $\overline{\text{WAIT}}$

(1) In Normal Extended Mode

(a) Program Wait Mode: A specified number of wait states T_W are always inserted betv T_2 state and T_3 state when accessing the external address space. The number of wait state specified by the settings of the WC1 and WC0 bits in WSCR (the WC11 and WC10 bits WSCR2 for the 256-kbyte extended area, and the WC21 and WC20 bits in WSCR2 for t extended area).

(b) Pin Wait Mode: A specified number of wait states T_W are always inserted between t and T_3 state when accessing the external address space. The number of wait states T_W is by the settings of the WC1 and WC0 bits (the WC21 and WC20 bits for the CP extended the $\overline{\text{WAIT}}$ pin is low at the falling edge of ϕ in the last T_2 or T_W state, another T_W state is If the $\overline{\text{WAIT}}$ pin is held low, T_W states are inserted until it goes high.

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



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 expansion area is accessed with CS256E = 1 and when the CP expansion area is accessed with CPCSE = 1.

Figure 6.25 Example of Wait State Insertion Timing (Pin Wait Mode)

RENESAS

(b) Pin Wait Mode: When accessing the external address space, a specified number of T_{DSW} can be inserted between the T_4 state and T_5 state of data state. The number of wait s is specified by the settings of the WC1 and WC0 bits (the WC21 and WC20 bits for the extended area). If the $\overline{\text{WAIT}}$ pin is low at the falling edge of ϕ in the last T_4, T_{DSW}, or T_{DO} another T_{DOW} state is inserted. If the $\overline{\text{WAIT}}$ pin is held low, T_{DOW} states are inserted until high.

Pin wait mode is useful when inserting four or more T_{DOW} states, or when changing the n 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 and T_5 state when accessing the external address space if the $\overline{\text{WAIT}}$ pin is low at the fall of ϕ in the last T_4 state. The number of wait states T_{DOW} is specified by the settings of the WC0 bits (the WC21 and WC20 bits for the CP extended area). Even if the $\overline{\text{WAIT}}$ pin is T_{DOW} 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 signal to the $\overline{\text{WAIT}}$ pin.

Figure 6.26 shows an example of wait state insertion timing in pin wait mode.

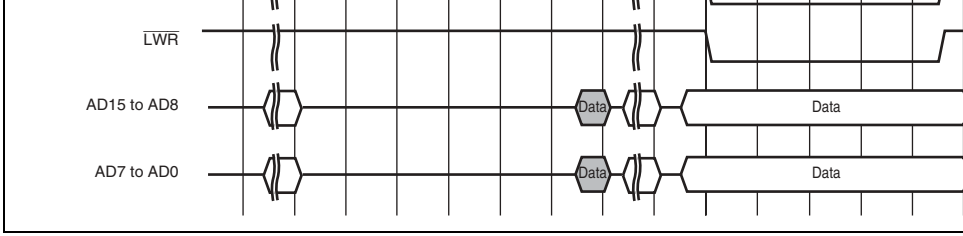


Figure 6.26 Example of Wait State Insertion Timing

Wait states cannot be inserted in a burst cycle. Burst accesses of a maximum four words
performed when the BRSTS0 bit in BCR is cleared to 0, and burst accesses of a maximu
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 6.27 and 6.28.

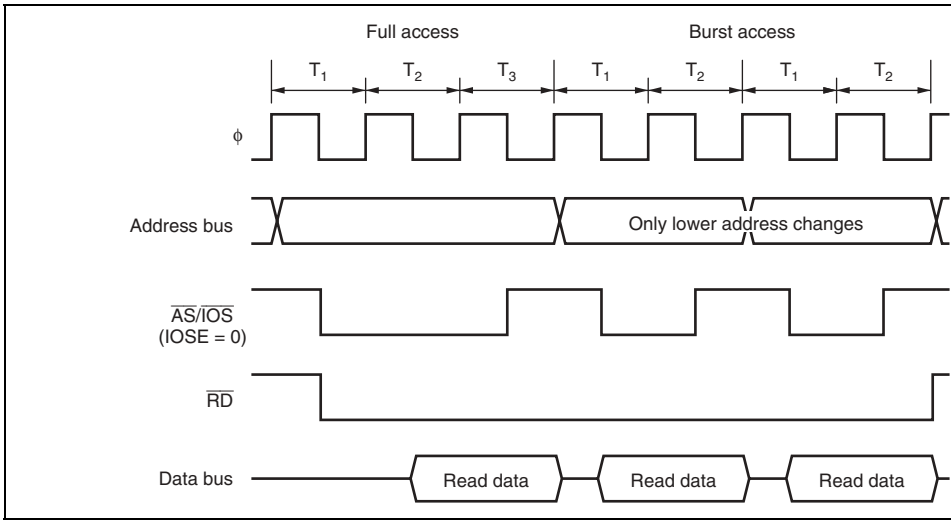


Figure 6.27 Access Timing Example in Burst ROM Space (AST = BRSTS1

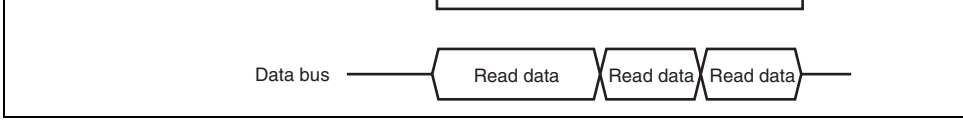


Figure 6.28 Access Timing Example in Burst ROM Space (AST = BRSTS1 =

6.6.2 Wait Control

As with the basic bus interface, program wait insertion or pin wait insertion using the $\overline{WA}$ possible in the initial cycle (full access) of the burst ROM interface. For details, see sectio Wait Control. Wait states cannot be inserted in a burst cycle.

the CPU write data. In figure 6.29 (b), an idle cycle is inserted, thus preventing data coll

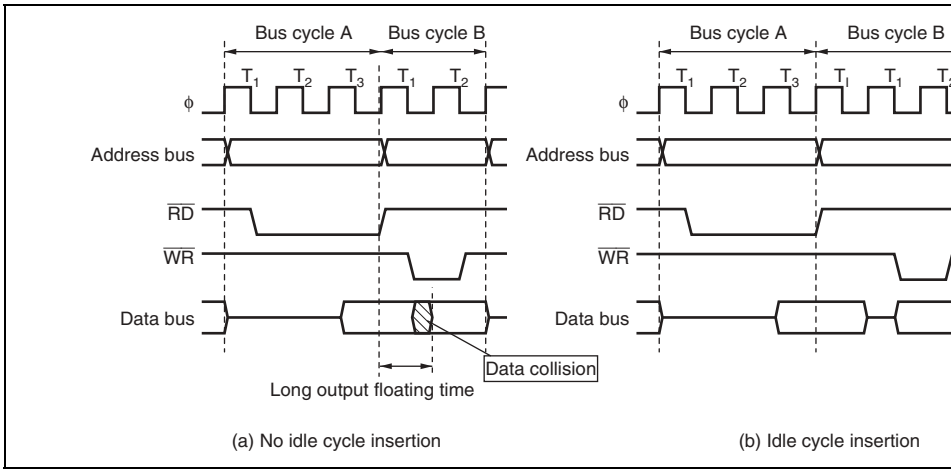


Figure 6.29 Examples of Idle Cycle Operation

Table 6.17 shows the pin states in an idle cycle.

Table 6.17 Pin States in Idle Cycle

Pins	Pin State
A23 to A0	Contents of immediately following bus cyc
D15 to D0	High impedance
$\overline{AS}$, $\overline{IOS}$, $\overline{CS256}$, $\overline{CPCS1}$	High
$\overline{RD}$	High
$\overline{HWR}$, $\overline{LWR}$	High

designated timing. If there are bus requests from more than one bus master, the bus maste
request acknowledge signal is sent to the one with the highest priority. When a bus maste
the bus mastership request acknowledge signal, it takes the bus mastership until that signa
canceled. The order of bus master priority is as follows:

(High) DTC > CPU (Low)

6.8.3 Bus Mastership Transfer Timing

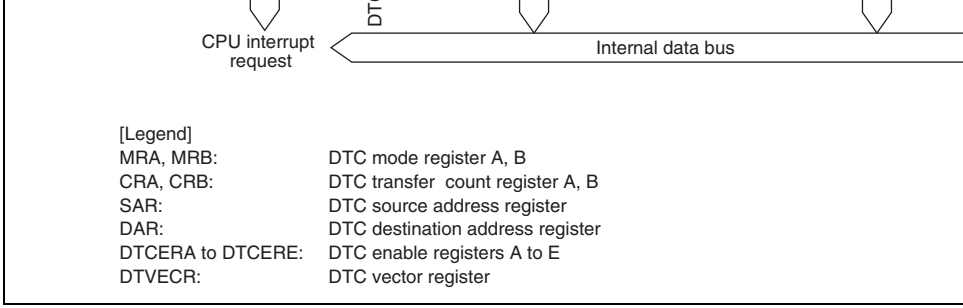
When a bus request is received from a bus master with a higher priority than that of the b
that has acquired the bus mastership and is currently operating, the bus mastership is not
necessarily transferred immediately. Each bus master can relinquish the bus mastership a
timings given below.

CPU: The CPU is the lowest-priority bus master, and if a bus mastership request is receiv
the DTC, the bus arbiter transfers the bus mastership to the DTC. The timing for transfer
bus mastership is as follows:

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

DTC: The DTC sends the bus arbiter a request for the bus mastership when a request for
activation occurs. The DTC releases the bus mastership after a series of processes has co

- 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 DTCERE:	DTC enable registers A to E
DTVECR:	DTC vector register

Figure 7.1 Block Diagram of DTC

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)
5 4	DM1 DM0	Undefined	—	Destination Address Mode 1 and 0 These bits specify a DAR operation after a data 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 2	MD1 MD0	Undefined	—	DTC Mode 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 destination side is set to be a repeat area or block area in repeat 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

[Legend]

*: Don't care

				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 DTCERE. 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 hav These bits are not cleared when the DISEL bit the specified number of transfers have not bee completed

0	DTCEn0	(52)CCIA	(68)CMIAT	(83)IRQ1	(45)CMIBX	(107

[Legend]

n: A to E

( ): Vector number

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

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 to this bit. [Clearing conditions] • When the DISEL bit is 0 and the specified of transfers have not ended • When 0 is written to the DISEL bit after a activated data transfer end interrupt (SWD request has been sent to the CPU. This bit will not be cleared when the DISEL bi data transfer has ended or when the specifie of transfers has ended.

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 modified.
4 to 0	—	All 0	R/W	Reserved The initial value should not be changed.

				modified.
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. of pins are selected starting from EVENT0. W PAnDDR is set to 1, inputting events to EVEN 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

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.

The EVENTI interrupt request activates the DTC and transfers data from RAM to RAM same address. Data is incremented in the DTC. The lower five bits of SAR and DAR are with address code that is generated by the ECS flag status.

									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	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0

7.3.1 Event Counter Handling Priority

EVENT0 to EVENT15 count handling is operated in the priority shown as below.

High Low

EVENT0 > EVENT1 · · · · · · · · · · EVENT14 > EVENT15

7.4 Activation Sources

The DTC is activated by an interrupt request or by a write to DTVECR by software. The request source to activate the DTC is selected by DTCER. At the end of a data transfer consecutive transfer in the case of chain transfer), the interrupt flag that became the activ source or the corresponding DTCER bit is cleared. The activation source flag, in the cas RXI0, for example, is the RDRF flag in SCI_0.

When an interrupt has been designated as a DTC activation source, the existing CPU ma and interrupt controller priorities have no effect. If there is more than one activation sou same time, the DTC operates in accordance with the default priorities. Figure 7.2 shows diagram of DTC activation source control. For details on the interrupt controller, see sec Interrupt Controller.

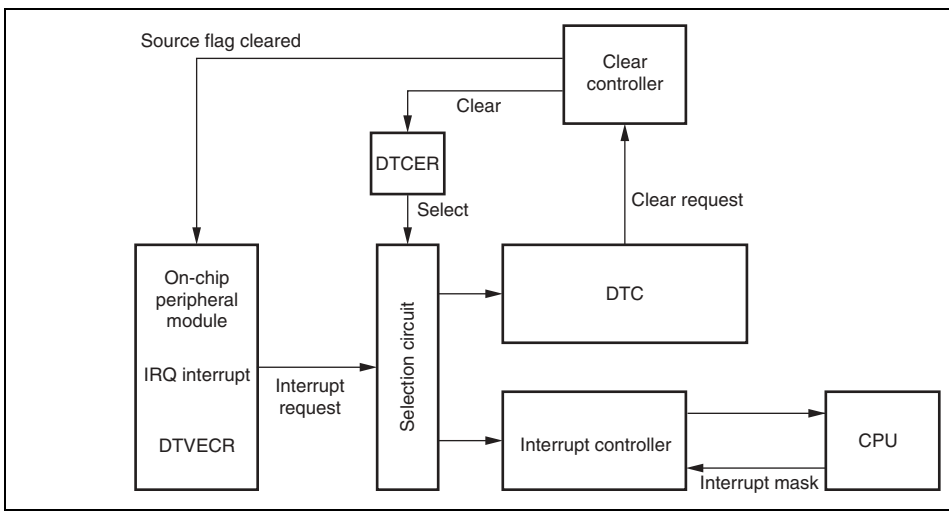


Figure 7.2 Block Diagram of DTC Activation Source Control

(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 th information start address.

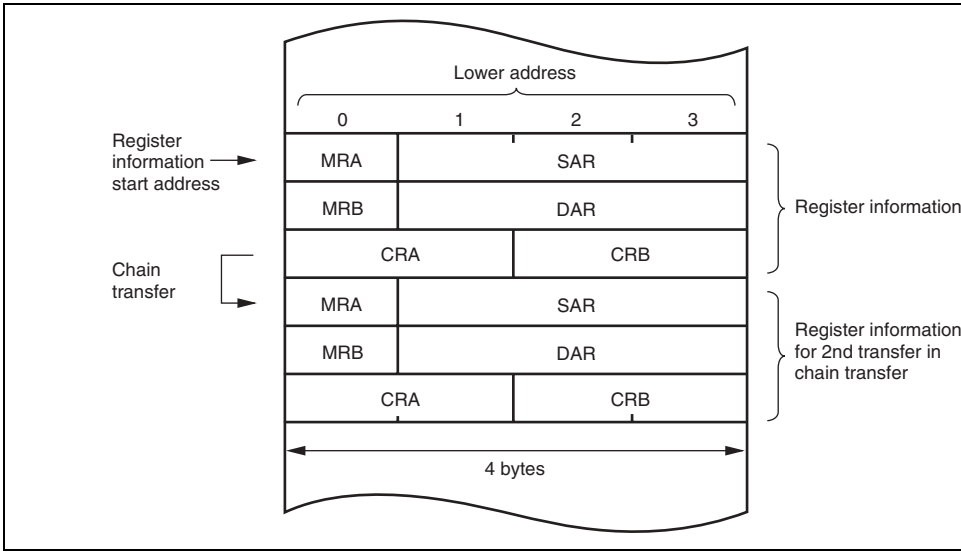


Figure 7.3 DTC Register Information Location in Address Space

	CMIBX	45	H'045A	DTCED0
FRT	ICIA	48	H'0460	DTCEA2
	ICIB	49	H'0462	DTCEA1
	OCIA	52	H'0468	DTCEA0
	OCIB	53	H'046A	DTCEB7
TMR_0	CMIA0	64	H'0480	DTCEB2
	CMIB0	65	H'0482	DTCEB1
TMR_1	CMIA1	68	H'0488	DTCEB0
	CMIB1	69	H'048A	DTCEC7
TMR_Y	CMIAY	72	H'0490	DTCEC6
	CMIBY	73	H'0492	DTCEC5
IIC_2	IICI2	76	H'0498	DTCEB6
IIC_3	IICI3	78	H'049C	DTCED4
SCI_0	RXI0	81	H'04A2	DTCEC2
	TXI0	82	H'04A4	DTCEC1
SCI_1	RXI1	85	H'04AA	DTCEC0
	TXI1	86	H'04AC	DTCED7
SCI_2	RXI2	89	H'04B2	DTCED6
	TXI2	90	H'04B4	DTCED5
IIC_0	IICI0	94	H'04BC	DTCEB5

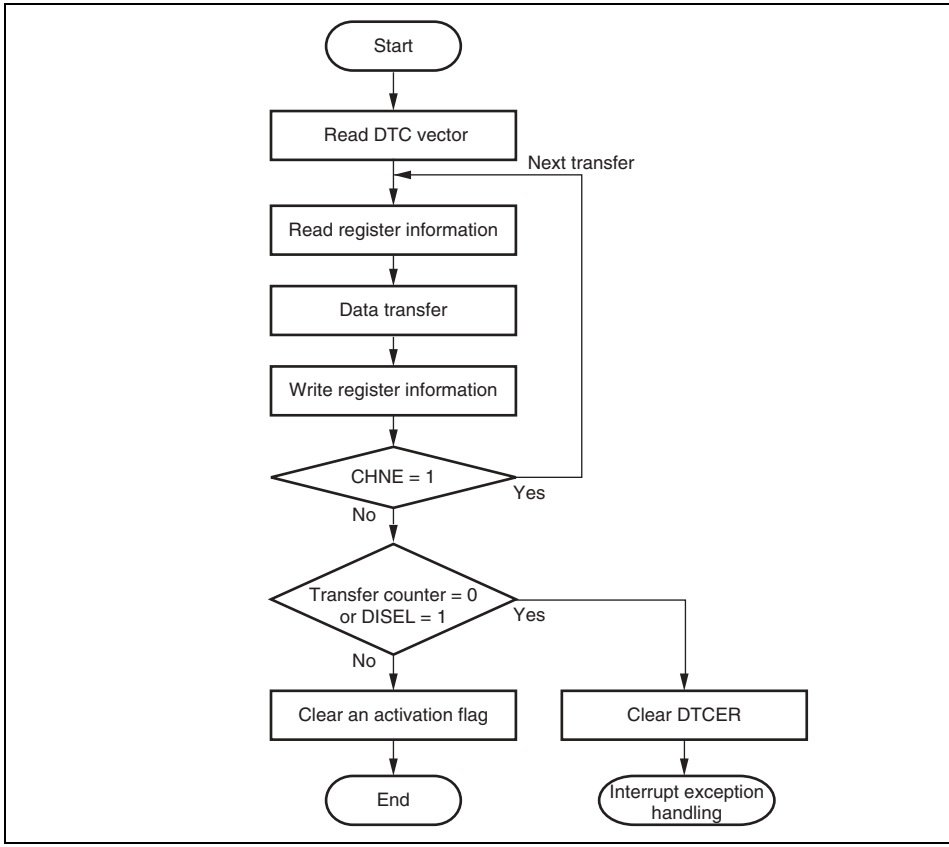


Figure 7.4 DTC Operation Flowchart

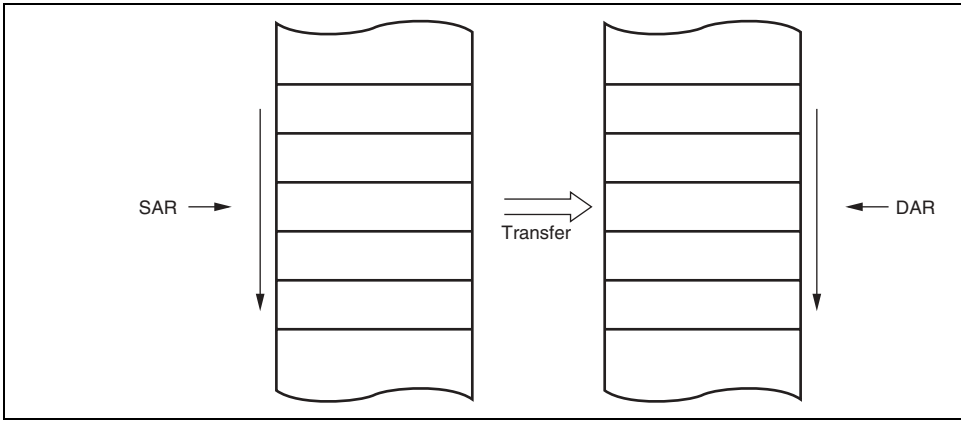


Figure 7.5 Memory Mapping in Normal Mode

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

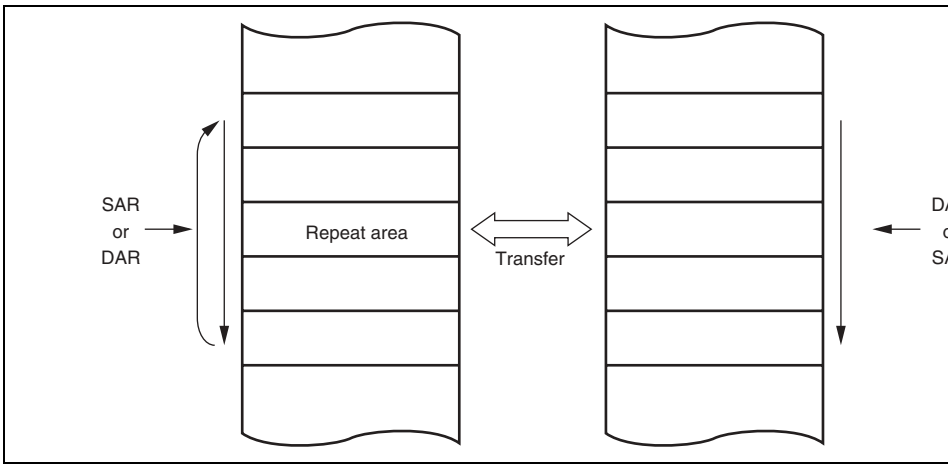


Figure 7.6 Memory Mapping in Repeat Mode

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

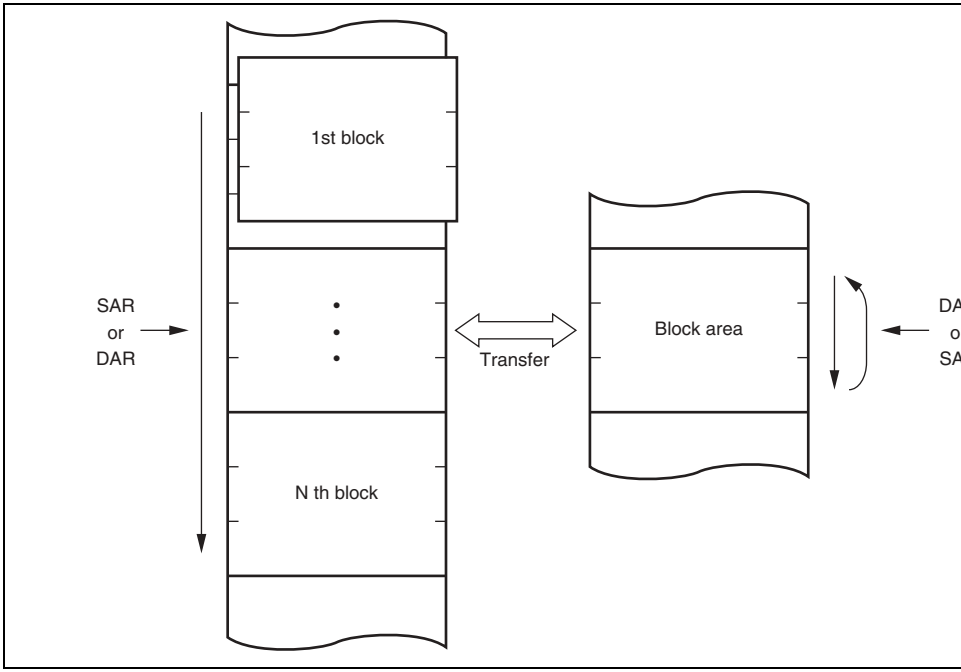


Figure 7.7 Memory Mapping in Block Transfer Mode

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.

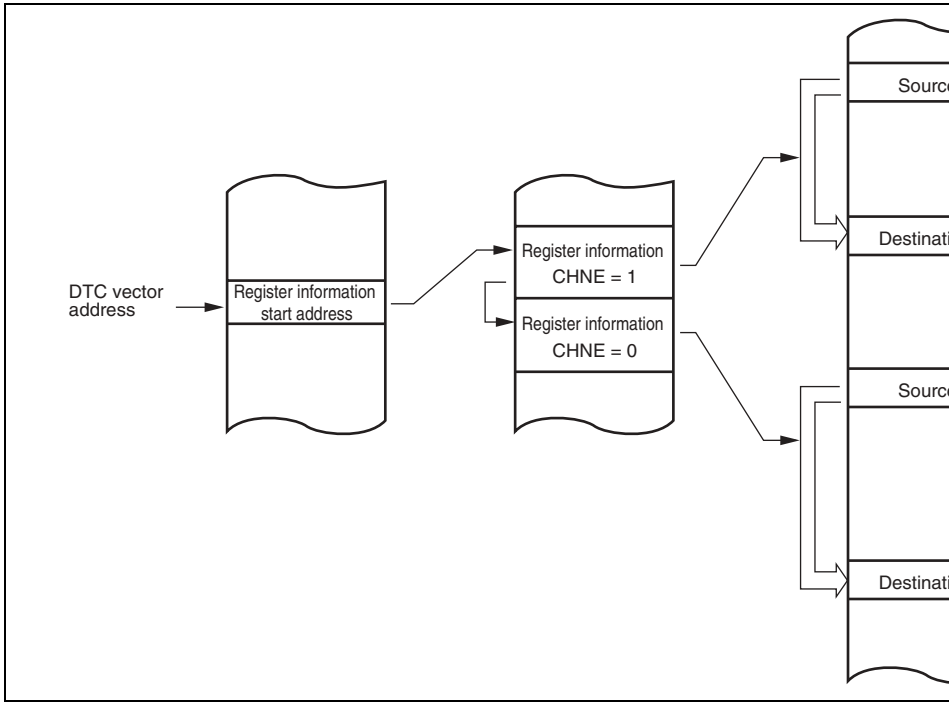


Figure 7.8 Chain Transfer Operation

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

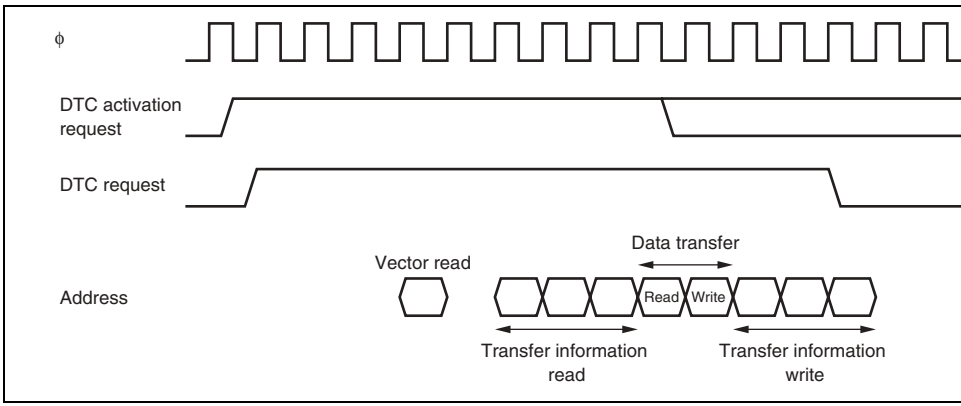


Figure 7.9 DTC Operation Timing (Example in Normal Mode or Repeat Mo

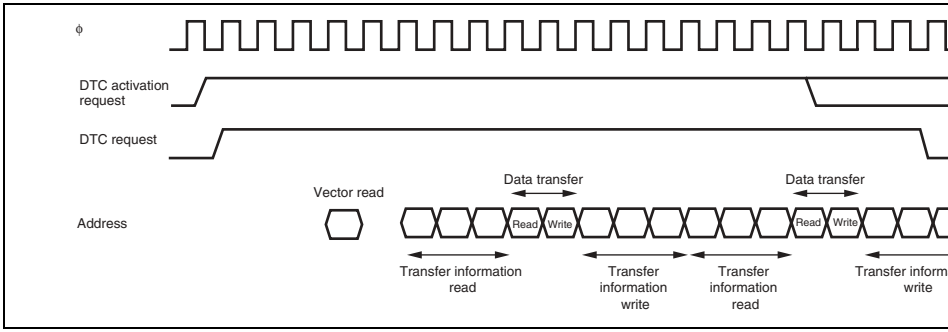


Figure 7.11 DTC Operation Timing (Example of Chain Transfer)

7.6.7 Number of DTC Execution States

Table 7.8 lists the execution status for a single DTC data transfer, and table 7.9 shows th of states required for each execution status.

Table 7.8 DTC Execution Status

Mode	Vector Read I	Register Information Read/Write J	Data Read K	Data Write L	Inte Ope M
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)

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

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

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

For example, when the DTC vector address table is located in on-chip ROM, normal mod
and data is transferred from on-chip ROM to an internal I/O register, then the time requir
DTC operation is 13 states. The time from activation to the end of data write is 10 states.

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.

[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 complete (RXI) interrupt. Since the generation of a receive error during the SCI recep operation will disable subsequent reception, the CPU should be enabled to accept rece 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 decreme 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.

7.8.2 Software Activation

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

[1] Set MRA to incrementing source address (SM1 = 1, SM0 = 0), incrementing destinati address (DM1 = 1, DM0 = 0), block transfer mode (MD1 = 1, MD0 = 0), and byte siz 0). The DTS bit can have any value. Set MRB for one block transfer by one interrupt 0). Set the transfer source address (H'1000) in SAR, the transfer destination address (H in DAR, and 128 (H'8080) in CRA. Set 1 (H'0001) in CRB.

[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 transfer by software.

[4] Write 1 to the SWDTE bit and the vector number (H'60) to DTVECR. The write data

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 Setting Required on Entering Subactive Mode or Watch Mode

Set the MSTP14 bit in MSTPCRH to 1 to make the DTC enter module stop mode, then c
that is set to 1 before making a transition to subactive mode or watch mode.

7.9.5 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
MRB.

can drive a Darlington transistor in output mode. Ports 8, C0 to C5 and D6 to D7 are NM
pull output.

Table 8.1 Port Functions

Port	Description	Extended Mode (EXPE = 1)	Single-Chip Mode (EXPE = 0)	I/O S
Port 1	General I/O port also functioning as PWM output, address output, and address/data multiplex input/output	P17/A7/AD7	P17/PW7	Built pull-
		P16/A6/AD6	P16/PW6	
		P15/A5/AD5	P15/PW5	LED capa
		P14/A4/AD4	P14/PW4	
		P13/A3/AD3	P13/PW3	(sink mA)
		P12/A2/AD2	P12/PW2	
		P11/A1/AD1	P11/PW1	
		P10/A0/AD0	P10/PW0	
Port 2	General I/O port also functioning as PWM output, address output, and address/data multiplex input/output	P27/A15/AD15	P27/PW15	Built pull-
		P26/A14/AD14	P26/PW14	
		P25/A13/AD13	P25/PW13	LED capa
		P24/A12/AD12	P24/PW12	
		P23/A11/AD11	P23/PW11	(sink mA)
		P22/A10/AD10	P22/PW10	
		P21/A9/AD9	P21/PW9	
		P20/A8/AD8	P20/PW8	

Port 4	General I/O port also functioning as interrupt input, and TMR_0, TMR_1, TMR_X, TMR_Y input	P47/$\overline{\text{IRQ7}}$/TMOY
		P46/$\overline{\text{IRQ6}}$/TMOX
		P45/$\overline{\text{IRQ5}}$/TMIY
		P44/$\overline{\text{IRQ4}}$/TMIX
		P43/$\overline{\text{IRQ3}}$/TMO1
		P42/$\overline{\text{IRQ2}}$/TMO0
		P41/$\overline{\text{IRQ1}}$/TMI1
		P40/$\overline{\text{IRQ0}}$/TMI0
Port 5	General I/O port also functioning as interrupt input, PWMX output, and SCI_0, SCI_1, SCI_2 I/O pins	P57/$\overline{\text{IRQ15}}$/PWX1
		P56/$\overline{\text{IRQ14}}$/PWX0
		P55/$\overline{\text{IRQ13}}$/RxD2
		P54/$\overline{\text{IRQ12}}$/TxD2
		P53/$\overline{\text{IRQ11}}$/RxD1/IrRxD
		P52/$\overline{\text{IRQ10}}$/TxD1/IrTxD
		P51/$\overline{\text{IRQ9}}$/RxD0
		P50/$\overline{\text{IRQ8}}$/TxD0

Port 7	General I/O port also functioning as A/D converter analog input, D/A converter analog output, and interrupt input	P77/$\overline{\text{ExIRQ7}}$/AN7/DA1 P76/$\overline{\text{ExIRQ6}}$/AN6/DA0 P75/$\overline{\text{ExIRQ5}}$/AN5 P74/$\overline{\text{ExIRQ4}}$/AN4 P73/$\overline{\text{ExIRQ3}}$/AN3 P72/$\overline{\text{ExIRQ2}}$/AN2 P71/AN1 P70/AN0

	and IIC_1 inputs/outputs		
Port 9	General I/O port also functioning as bus control input/output, system clock output, and external sub-clock input	P97/$\overline{\text{WAIT}}$/$\overline{\text{CS256}}$	P97
		P96/ϕ/EXCL	
		$\overline{\text{AS}}$/$\overline{\text{IOS3}}$	P95
		$\overline{\text{HWR}}$	P94
		$\overline{\text{RD}}$	P93
		P92/$\overline{\text{CPCS1}}$	P92
		P91/$\overline{\text{AH}}$	P91
		P90/$\overline{\text{LWR}}$	P90

		PA2/KIN10/ A18/EVENT2	PA2/KIN10/ EVENT2	
		PA1/KIN9/ A17/SSE2I/ EVENT1	PA1/KIN9/ SSE2I/ EVENT1	
		PA0/KIN8/ A16/SSE0I/ EVENT0	PA0/KIN8/ SSE0I/ EVENT0	
Port B	General I/O port also functioning as DTC event counter input	PB7/EVENT15 PB6/EVENT14 PB5/EVENT13 PB4/EVENT12 PB3/EVENT11 PB2/EVENT10 PB1/EVENT9 PB0/EVENT8		
Port C	General I/O port also functioning as PWMX output and IIC_2, IIC_3, and IIC_4 I/O pins	PC7/PWX3 PC6/PWX2 PC5/SDA4 PC4/SCL4 PC3/SDA3 PC2/SCL3 PC1/SDA2 PC0/SCL2		NM pu (P PC
Port D	General I/O port also functioning as LPC I/O, and IIC_5 I/O pins	PD7/SDA5 PD6/SCL5 PD5/LPCPD PD4/CLKRUN PD3/GA20 PD2/PME PD1/LSMI PD0/LSCI		Bu pu MO (P PD NM pu (P PD

2. 16-bit data bus is selected.

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	In normal extended mode (ADMXE = 0):
6	P16DDR	0	W	The corresponding port 1 pins are address o
5	P15DDR	0	W	when P1DDR bits are set to 1, and input po
4	P14DDR	0	W	cleared to 0.
3	P13DDR	0	W	In address/data multiplex extended mode (A 1):
2	P12DDR	0	W	When the bus width is 16 bits, lower 8 bits o
1	P11DDR	0	W	address/data multiplex bus. When the bus w
0	P10DDR	0	W	bits, this register is used in the same way as chip mode.
				In single-chip mode:
				The corresponding port 1 pins are output po PWM outputs when the P1DDR bits are set input ports when cleared to 0.

1	P11DR	0	R/W	
0	P10DR	0	R/W	

8.1.3 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 input state, the corres
6	P16PCR	0	R/W	input pull-up MOS is turned on when a P1F set to 1.
5	P15PCR	0	R/W	In address-data multiplex extended bus mo
4	P14PCR	0	R/W	used, the initial value should not be change
3	P13PCR	0	R/W	
2	P12PCR	0	R/W	
1	P11PCR	0	R/W	
0	P10PCR	0	R/W	

ABWCP						
Pin function	P1n input pin	AD7 to AD0 input/output pin	P1n input pin	A7 to A0 output pin	Setting prohibited	I

[Legend]
n = 7 to 0

Single-Chip Mode (EXPE = 0):

The function of port 1 pins is switched as shown below according to the combination of bit and P1nDDR bit in PWOERA of PWM and the PWMS bit in PTCNT0.

P1nDDR	0	1	1	
PWMS	—	0	1	
OEn	—	0	—	
Pin function	P1n input pin	P1n output pin		PWn c

[Legend]
n = 7 to 0

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	In normal extended mode (ADMXE = 0): The corresponding port 2 pins are addre ports when the P2DDR bits are set to 1, ports when cleared to 0. Pins function as the address output port on the setting of bits IOSE and CS256E Address/data multiplex extended mode (1): The upper 8-bit of address/data multiplex In single-chip mode: The corresponding port 2 pins are output PWM outputs when the P2DDR bits are and input ports when cleared to 0.
6	P26DDR	0	W	
5	P25DDR	0	W	
4	P24DDR	0	W	
3	P23DDR	0	W	
2	P22DDR	0	W	
1	P21DDR	0	W	
0	P20DDR	0	W	

1	P21DR	0	R/W	
0	P20DR	0	R/W	

8.2.3 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 input state, the corre input pull-up MOS is turned on when a P2 is set to 1.
6	P26PCR	0	R/W	
5	P25PCR	0	R/W	
4	P24PCR	0	R/W	
3	P23PCR	0	R/W	
2	P22PCR	0	R/W	
1	P21PCR	0	R/W	
0	P20PCR	0	R/W	

Address 13 = 1:$\overline{\text{ADFULLE}}$ • $\overline{\text{CS256E}}$ • ($\overline{\text{CPCSE}}$ | IOSE)

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

P2nDDR	0		1		
ADMXE	0	1	0		1
Address 13	—	—	0	1	—
Pin function	P27 to P25 input pins	AD15 to AD13 input/output pins	A15 to A13 output pins	P27 to P25 output pins	AD15 to A input/outpu

[Legend]

n = 7 to 5

P24DDR	0		1		
ADMXE	0	1	0		1
Address 11	—	—	0	1	—
Pin function	P24 input pin	AD12 input/output pin	A12 output pin	P24 output pin	AD12 input/o

P23DDR	0		1		
ADMXE	0	1	0		1
Address 11	—	—	0	1	—
Pin function	P23 input pin	AD11 input/output pin	A11 output pin	P23 output pin	AD11 input/o

OEm	—	0	1
Pin function	P27 to P20 input pins	P27 to P20 output pins	PW15 to PW8 out

[Legend]
n = 7 to 0
m = 15 to 8

8.2.5 Port 2 Input Pull-Up MOS

Port 2 has a built-in input pull-up MOS that can be controlled by software. This input pul MOS can be used regardless of the operating mode. Table 8.3 summarizes the input pull- states.

Table 8.3 Port 2 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 P2DDR = 0 and P2PCR = 1; otherwise off.

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

Bit	Bit Name	Initial Value	R/W	Description
7	P37DDR	0	W	In normal extended mode:
6	P36DDR	0	W	Bidirectional data bus
5	P35DDR	0	W	In other mode:
4	P34DDR	0	W	The corresponding port 3 pins are output port
3	P33DDR	0	W	the P3DDR bits are set to 1, and input ports w
2	P32DDR	0	W	cleared to 0.
1	P31DDR	0	W	
0	P30DDR	0	W	

8.3.2 Port 3 Data Register (P3DR)

P3DR stores output data for the port 3 pins.

Bit	Bit Name	Initial Value	R/W	Description
7	P37DR	0	R/W	In normal extended mode (ADMXE = 0):
6	P36DR	0	R/W	If a port 3 read is performed while the P3DDR
5	P35DR	0	R/W	set to 1, the P3DR values are read. When the
4	P34DR	0	R/W	bits are cleared to 0, 1 is read.
3	P33DR	0	R/W	In other mode:
2	P32DR	0	R/W	If a port 3 read is performed while the P3DDR
1	P31DR	0	R/W	set to 1, the P3DR values are read. If a port 3
0	P30DR	0	R/W	performed while the P3DDR bits are cleared t
				pin states are read.

0	P30PCR	0	R/W

8.3.4 Pin Functions

Normal Extended Mode:

Port 3 pins automatically function as the bidirectional data bus.

Address/Data Multiplex Mode:

Same operation as the single-chip mode.

Single-Chip Mode:

- P37/$\overline{\text{WUE15}}$

  The pin function is switched as shown below according to the P37DDR bit.

  When the WUEM15 bit in WUEMR3 of the interrupt controller is cleared to 0, this pi
  used as the $\overline{\text{WUE15}}$ input pin. To use this pin as the $\overline{\text{WUE15}}$ input pin, clear the P37D
  to 0.

P37DDR	0		1
WUEM15	0	1	—
Pin Function	$\overline{\text{WUEI15}}$ input pin	P37 input pin	P37 output

When the WUEM13 bit in WUEMR3 of the interrupt controller is cleared to 0, this p
used as the $\overline{\text{WUE13}}$ input pin. To use this pin as the $\overline{\text{WUE13}}$ input pin, clear the P35
to 0.

P35DDR	0		1
WUEM13	0	1	—
Pin function	$\overline{\text{WUE13}}$ input pin	P35 input pin	P35 output pin

- P34/$\overline{\text{WUE12}}$

  The pin function is switched as shown below according to the P34DDR bits.

  When the WUEM12 bit in WUEMR3 of the interrupt controller is cleared to 0, this p
  used as the $\overline{\text{WUE12}}$ input pin. To use this pin as the $\overline{\text{WUE12}}$ input pin, clear the P34
  to 0.

P34DDR	0		1
WUEM12	0	1	—
Pin function	$\overline{\text{WUE12}}$ input pin	P34 input pin	P34 output pin

- P33/$\overline{\text{WUE11}}$

  The pin function is switched as shown below according to the P33DDR bits.

  When the WUEM11 bit in WUEMR3 of the interrupt controller is cleared to 0, this p
  used as the $\overline{\text{WUE11}}$ input pin. To use this pin as the $\overline{\text{WUE11}}$ input pin, clear the P33
  to 0.

P33DDR	0		1
WUEM11	0	1	—
Pin function	$\overline{\text{WUE11}}$ input pin	P33 input pin	P33 output pin

When the WUEM9 bit in WUEMR3 of the interrupt controller is cleared to 0, this pin used as the $\overline{\text{WUE9}}$ input pin. To use this pin as the $\overline{\text{WUE9}}$ input pin, clear the P31DD 0.

P31DDR	0		1
WUEM9	0	1	—
Pin function	$\overline{\text{WUE9}}$ input pin	P31 input pin	P31 output pin

- P30/$\overline{\text{WUE8}}$

  The pin function is switched as shown below according to the P30DDR bits.

  When the WUEM8 bit in WUEMR3 of the interrupt controller is cleared to 0, this pin used as the $\overline{\text{WUE8}}$ input pin. To use this pin as the $\overline{\text{WUE8}}$ input pin, clear the P30DD 0.

P30DDR	0		1
WUEM8	0	1	—
Pin function	$\overline{\text{WUE8}}$ input pin	P30 input pin	P30 output pin

(EXPE = 0)	
Address-data multiplex extended mode (EXPE = 1, ADMXE = 1)	

[Legend]

Off : Always off.

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

7	P47DDR	0	W	If port 4 pins are specified for use as the gener
6	P46DDR	0	W	port, the corresponding port 4 pins are output p
5	P45DDR	0	W	when the P4DDR bits are set to 1, and input po
4	P44DDR	0	W	cleared to 0.
3	P43DDR	0	W	
2	P42DDR	0	W	
1	P41DDR	0	W	
0	P40DDR	0	W	

8.4.2 Port 4 Data Register (P4DR)

P4DR stores output data for the port 4 pins.

Bit	Bit Name	Initial Value	R/W	Description
7	P47DR	0	R/W	P4DR stores output data for the port 4 pins tha
6	P46DR	0	R/W	used as the general output port.
5	P45DR	0	R/W	If a port 4 read is performed while the P4DDR
4	P44DR	0	R/W	set to 1, the P4DR values are read. If a port 4 r
3	P43DR	0	R/W	performed while the P4DDR bits are cleared to
2	P42DR	0	R/W	pin states are read.
1	P41DR	0	R/W	
0	P40DR	0	R/W	

Pin function	P47 input pin	P47 output pin	TMOY output
	$\overline{\text{IRQ7}}$ input pin		

- P46/$\overline{\text{IRQ6}}$/TMOX

  The pin function is switched as shown below according to the combination of the OS bits in TCSR of TMR_X and the P46DDR bit.

  When the ISS6 bit in ISSR is cleared to 0 and the IRQ6E bit in IER of the interrupt c is set to 1, this pin can be used as the $\overline{\text{IRQ6}}$ input pin. To use this pin as the $\overline{\text{IRQ6}}$ inp clear the P46DDR bit to 0.

OS3 to OS0	All 0		One bit is set a
P46DDR	0	1	—
Pin function	P46 input pin	P46 output pin	TMOX output
	$\overline{\text{IRQ6}}$ input pin		

	IRQ5 input pin	

- P44/$\overline{\text{IRQ4}}$/TMIX

  The pin function is switched as shown below according to the P44DDR bits.

  When the TMIXS bit in PTCNT0 is cleared to 0 and the external clock is selected by to CKS0 bits in TCR of TMR_X, this bit is used as the TMCIX input pin. When the C and CCLR0 bits in TCR of TMR_X are set to 1, this pin is used as the TMRIX input When the ISS4 bit in ISSR is cleared to 0 and the IRQ4E bit in IER of the interrupt co is set to 1, this pin can be used as the $\overline{\text{IRQ4}}$ input pin. To use this pin as the $\overline{\text{IRQ4}}$ inpu clear the P44DDR bit to 0.

P44DDR	0	1
Pin function	P44 input pin	P44 output pin
	TMIY (TMCIY/TMRIY) input pin	
	$\overline{\text{IRQ4}}$ input pin	

- P43/$\overline{\text{IRQ3}}$/TMO1

  The pin function is switched as shown below according to the OS3 to OS0 bits in TCS TMR_1 and the P43DDR bit. When the ISS3 bit in ISSR is cleared to 0 and the IRQ3 IER of the interrupt controller is set to 1, this pin can be used as the $\overline{\text{IRQ3}}$ input pin. T this pin as the $\overline{\text{IRQ3}}$ input pin, clear the P43DDR bit to 0.

OS3 to OS0	All 0		One bit is set as 1
P43DDR	0	1	—
Pin function	P43 input pin	P43 output pin	TMO1 output pin
	$\overline{\text{IRQ3}}$ input pin		

- P41/$\overline{\text{IRQ1}}$/TMI1

  The pin function is switched as shown below according to the P41DDR bits.

  When the TMI1S bit in PTCNT0 is cleared to 0 and the external clock is selected by to CKS0 bits in TCR of TMR_1, this bit is used as the TMCI1 input pin. When the C and CCLR0 bits in TCR of TMR_1 are set to 1, this pin is used as the TMRI1 input the ISS1 bit in ISSR is cleared to 0 and the IRQ1E bit in IER of the interrupt control 1, this pin can be used as the $\overline{\text{IRQ1}}$ input pin. To use this pin as the $\overline{\text{IRQ1}}$ input pin, c P41DDR bit to 0.

P41DDR	0	1
Pin function	P41 input pin	P41 output pin
	TMI1(TMCI1/TMRI1) input pin	
	$\overline{\text{IRQ1}}$ input pins	

- P40/$\overline{\text{IRQ0}}$/TMI0

  The pin function is switched as shown below according to the P40DDR bits.

  When the TMI0S bit in PTCNT0 is cleared to 0 and the external clock is selected by to CKS0 bits in TCR of TMR_0, this bit is used as the TMCI0 input pin. When the C and CCLR0 bits in TCR of TMR_0 are set to 1, this pin is used as the TMRI0 input the ISS0 bit in ISSR is cleared to 0 and the IRQ0E bit in IER of the interrupt control 1, this pin can be used as the $\overline{\text{IRQ0}}$ input pin. To use this pin as the $\overline{\text{IRQ0}}$ input pin, c P40DDR bit to 0.

P40DDR	0	1
Pin function	P40 input pin	P40 output pin
	TMI0(TMCI0/TMRI0) input pin	
	$\overline{\text{IRQ0}}$ input pin	

7	P57DDR	0	W	If port 5 pins are specified for use as the gen
6	P56DDR	0	W	port, the corresponding port 5 pins are outpu
5	P55DDR	0	W	when the P5DDR bits are set to 1, and input
4	P54DDR	0	W	when cleared to 0.
3	P53DDR	0	W	
2	P52DDR	0	W	
1	P51DDR	0	W	
0	P50DDR	0	W	

8.5.2 Port 5 Data Register (P5DR)

P5DR stores output data for the port 5 pins.

Bit	Bit Name	Initial Value	R/W	Description
7	P57DR	0	R/W	P5DR stores output data for the port 5 pins th
6	P56DR	0	R/W	used as the general output port.
5	P55DR	0	R/W	If a port 5 read is performed while the P5DDR
4	P54DR	0	R/W	set to 1, the P5DR values are read. If a port 5
3	P53DR	0	R/W	performed while the P5DDR bits are cleared
2	P52DR	0	R/W	pin states are read.
1	P51DR	0	R/W	
0	P50DR	0	R/W	

Pin function	P57 input pin	P57 output pin	PWX1 out
	$\overline{\text{IRQ15}}$ input pin		

- P56/$\overline{\text{IRQ14}}$/PWX0

  The pin function is switched as shown below according to the combination of the OE DACR of PWMX and the P56DDR bit.

  When the ISS14 bit in ISSR16 is cleared to 0 and the IRQ14E bit in IER16 of the int controller is set to 1, this pin can be used as the $\overline{\text{IRQ14}}$ input pin. To use this pin as t input pin, clear the P56DDR bit to 0.

OEA	0		1
P56DDR	0	1	—
Pin function	P56 input pin	P56 output pin	PWX0 outp
	$\overline{\text{IRQ14}}$ input pin		

- P55/$\overline{\text{IRQ13}}$/RxD2

  The pin function is switched as shown below according to the combination of the RE SCR of SCI_2 and the P55DDR bit.

  When the ISS13 bit in ISSR16 is cleared to 0 and the IRQ13E bit in IER16 of the int controller is set to 1, this pin can be used as the $\overline{\text{IRQ13}}$ input pin. To use this pin as t input pin, clear the P55DDR bit to 0.

RE	0		1
P55DDR	0	1	—
Pin function	P55 input pin	P55 output pin	RxD2 inp
	$\overline{\text{IRQ13}}$ input pin		

- P53/$\overline{\text{IRQ11}}$/RxD1/IrRxD

  The pin function is switched as shown below according to the combination of the RE
  SCR of SCI_1 and the P53DDR bit.

  When the ISS11 bit in ISSR16 is cleared to 0 and the IRQ11E bit in IER16 of the inte
  controller is set to 1, this pin can be used as the $\overline{\text{IRQ11}}$ input pin. To use this pin as th
  input pin, clear the P53DDR bit to 0.

RE	0		1
P53DDR	0	1	—
Pin function	P53 input pin	P53 output pin	RxD1/IrRxD ir
	$\overline{\text{IRQ11}}$ input pin		

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

  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 this pin as th
  input pin, clear the P52DDR bit to 0.

TE	0		1
P52DDR	0	1	—
Pin function	P52 input pin	P52 output pin	TxD1/IrTxD o
	$\overline{\text{IRQ10}}$ input pin		

- P50/$\overline{\text{IRQ8}}$/TxD0

  The pin function is switched as shown below according to the combination of the TE
  SCR of SCI_0 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 this pin as th
  input pin, clear the P50DDR bit to 0.

TE	0		1
P50DDR	0	1	—
Pin function	P50 input pin	P50 output pin	TxD0 out
	$\overline{\text{IRQ8}}$ input pin		

- Noise cancel cycle setting register (P6NCCS)

8.6.1 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	Normal extended mode (16-bit data bus):
6	P66DDR	0	W	The port functions as the data bus regardless values in these bits.
5	P65DDR	0	W	Other mode:
4	P64DDR	0	W	If port 6 pins are specified for use as the gen port, the corresponding port 6 pins are outpu when the P6DDR bits are set to 1, and input when cleared to 0.
3	P63DDR	0	W	
2	P62DDR	0	W	
1	P61DDR	0	W	
0	P60DDR	0	W	

1	P61DR	0	R/W	
0	P60DR	0	R/W	If a port 6 read is performed while the P6DD set to 1, the P6DR values are read. If a port performed while the P6DDR bits are cleared pin states are read.

8.6.3 Port 6 Pull-Up MOS Control Register (KMPCR6)

KMPCR6 controls the port 6 built-in input pull-up MOSs. This register is accessible wh KINWUE is 1. See section 3.2.2, System Control Register (SYSCR).

Bit	Bit Name	Initial Value	R/W	Description
7	KM7PCR	0	R/W	Normal extended mode (16-bit data bus):
6	KM6PCR	0	R/W	Operation is not affected.
5	KM5PCR	0	R/W	Other mode:
4	KM4PCR	0	R/W	When the pins are in input state, the corresp input pull-up MOS is turned on when a KMP set to 1.
3	KM3PCR	0	R/W	
2	KM2PCR	0	R/W	
1	KM1PCR	0	R/W	
0	KM0PCR	0	R/W	

				1: Current-limit specification is selected
4	—	0	R/W	Reserved The initial value should not be changed.
3	ADMXE	0	R/W	Address data multiplex bus interface enable 0: Normal extended bus interface 1: Address data multiplex extended bus inter
2 to 0	—	All 0	R/W	Reserved The initial value should not be changed.

8.6.5 Noise Canceler Enable Register (P6NCE)

P6NCE enables or disables the noise canceler circuit at port 6.

Bit	Bit Name	Initial Value	R/W	Description
7	P67NCE	0	R/W	In 16 bit bus mode in extended mode:
6	P66NCE	0	R/W	Port 6 operates as the data pin (D7 to D0).
5	P65NCE	0	R/W	In other mode:
4	P64NCE	0	R/W	Noise canceler circuit is enabled and the pin
3	P63NCE	0	R/W	fetched in the P6DR in the sampling cycle se P6NCCS.
2	P62NCE	0	R/W	The operating state changes according to the
1	P61NCE	0	R/W	control bits. Check the pin functions.
0	P60NCE	0	R/W	

1	P61NCMC	1	R/W	
0	P60NCMC	1	R/W	

8.6.7 Noise Canceler Cycle Setting Register (P6NCCS)

P6NCCS controls the sampling cycles of the noise canceler.

Bit	Bit Name	Initial Value	R/W	Description		
7 to 3	—	All undefined	R/W	Reserved. The read data is undefined. The value should not be changed.		
2	NCCK2	0	R/W	These bits set the sampling cycles of the no canceler.		
1	NCCK1	0	R/W			
0	NCCK0	0	R/W	000:	0.06 µs	φ/2
				001:	0.97 µs	φ/32
				010:	15.5 µs	φ/512
				011:	248.2 µs	φ/8192
				100:	993.0 µs	φ/32768
				101:	2.0 ms	φ/65536
				110:	4.0 ms	φ/131072
				111:	7.9 ms	φ/262144

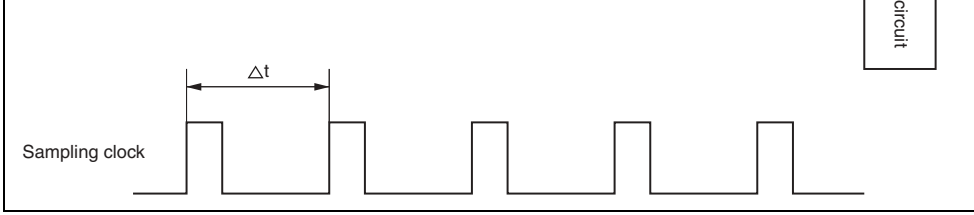


Figure 8.1 Noise Canceler Circuit

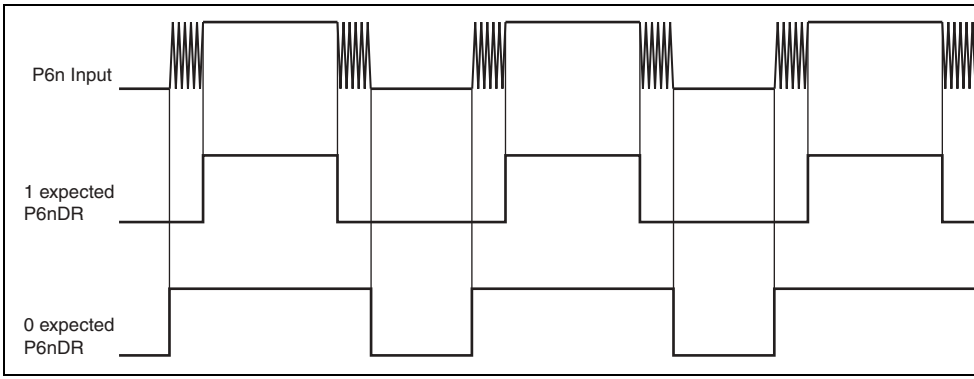


Figure 8.2 Noise Canceler Operation

- P67/ $\overline{\text{KIN7}}$

  The function of port 6 pins is switched as shown below according to the P67DDR bi

  When the KMIM7 bit in KMIMR6 of the interrupt controller is cleared to 0, this pin used as the $\overline{\text{KIN7}}$ input pin. To use this pin as the $\overline{\text{KIN7}}$ input pin, clear the P67DDR

Mode	Port		
P67DDR	0	0	1
P67NCE	0	1	—
Pin function	P67 input pin	P67 input pin (noise canceling)	P67 outp
	$\overline{\text{KIN7}}$ input pin		

- P66/FTOB/$\overline{\text{KIN6}}$

  The function of port 6 pins is switched as shown below according to the combinatio OEB bit in TOCR of FRT and the P66DDR bit.

  When the KMIM6 bit in KMIMR6 of the interrupt controller is cleared to 0, this pin used as the $\overline{\text{KIN6}}$ input pin. To use this pin as the $\overline{\text{KIN6}}$ input pin, clear the P66DDR

Mode	Port			FRT
OEB	0	0	0	1
P66DDR	0	0	1	—
P66NCE	0	1	—	—
Pin function	P66 input pin	P66 input pin (noise canceling)	P66 output pin	FTOB out
	$\overline{\text{KIN6}}$ input pin			

	KIN5 input pin			

- P64/FTIC/$\overline{\text{KIN4}}$

  The function of port 6 pins is switched as shown below according to the P64DDR bit.

  When the ICICE bit in TIER of FRT is set to 1, this pin can be used as the FTIC input

  When the KMIM4 bit in KMIMR6 of the interrupt controller is cleared to 0, this pin c

  used as the $\overline{\text{KIN4}}$ input pin. To use this pin as the $\overline{\text{KIN4}}$ input pin, clear the P64DDR

Mode	Port			FRT
P64DDR	0	0	1	0
P64NCE	0	1	—	0
Pin function	P64 input pin	P64 input pin (noise canceling)	P64 output pin	FTIC input
	$\overline{\text{KIN4}}$ input pin			

	KIN3 input pin		

- P62/FTIA/$\overline{\text{KIN2}}$

  The function of port 6 pins is switched as shown below according to the P62DDR bi

  When the ICIAE bit in TIER of FRT is set to 1, this pin can be used as the FTIA inp

  When the KMIM2 bit in KMIMR6 of the interrupt controller is cleared to 0, this pin used as the $\overline{\text{KIN2}}$ input pin. To use this pin as the $\overline{\text{KIN2}}$ input pin, clear the P62DDR

Mode	Port			FRT
P62DDR	0	0	1	0
P62NCE	0	1	—	0
Pin function	P62 input pin	P62 input pin (noise canceling)	P62 output pin	FTIA inpu
	$\overline{\text{KIN2}}$ input pin			

		canceling)		
	$\overline{\text{KIN1}}$ input pin			

- P60/FTCI/$\overline{\text{KIN0}}$

  The function of port 6 pins is switched as shown below according to the P60DDR bit.

  When the CKS1 and CKS0 bits in TCR of FRT are both set to 1, this pin can be used FTCI input pin. When the KMIM0 bit in KMIMR6 of the interrupt controller is cleare this pin can be used as the $\overline{\text{KIN0}}$ input pin. To use this pin as the $\overline{\text{KIN0}}$ input pin, clea P60DDR bit to 0.

Mode	Port			FRT
P60DDR	0	0	1	0
P60NCE	0	1	—	0
Pin function	P60 input pin	P60 input pin (noise canceling)	P60 output pin	FTCI input
	$\overline{\text{KIN0}}$ input pin			

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

8.7 Port 7

Port 7 is an 8-bit input port. Port 7 pins also function as the A/D converter analog input converter analog output pins, and interrupt input pins. Port 7 has the following register.

- Port 7 input data register (P7PIN)

8.7.1 Port 7 Input Data Register (P7PIN)

P7PIN indicates the pin states.

Bit	Bit Name	Initial Value	R/W	Description
7	P77PIN	Undefined*	R	When a P7PIN read is performed, the pin st always read. This register is assigned to the address as that of PBDDR. When the regist programmed, data is programmed in the PB the setting of port B is changed.
6	P76PIN	Undefined*	R	
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 value is determined in accordance with the pin states of P77 to P70.

values than those shown in the following table.

CH2 to CH0	B'111	Other than B'111		
DAOE1	0	0		1
ISS7	0	0	1	0
Pin function	AN7 input pin	P77 input pin	$\overline{\text{ExIRQ7}}$ input pin	DA1 outp

- P76/$\overline{\text{ExIRQ6}}$/AN6/DA0

  The port 7 function changes as shown in the following table, depending on the combi the SCAN bit and the CH2 to CH0 bits of ADCSR of the A/D converter, the DAOE0 DACR of the D/A converter, and the ISS6 bit of ISSR of the interrupt controller. Do n these bits to other values than those shown in the following table.

SCAN	0				1		
CH2 to CH0	B'110	Other than B'110			B'11*	Other than B'1	
DAOE0	0	0		1	0	0	
ISS6	0	0	1	0	0	0	1
Pin function	AN6 input pin	P76 input pin	$\overline{\text{ExIRQ6}}$ input pin	DA0 output pin	AN6 input pin	P76 input pin	$\overline{\text{ExIRQ6}}$ input pin

[Legend]

*: Don't care

[Legend]

*: Don’t care

- P74/$\overline{\text{ExIRQ4}}$/AN4

  The port 7 function changes as shown in the following table, depending on the comb
  the SCAN bit and the CH2 to CH0 bits of ADCSR of the A/D converter and the ISS
  ISSR of the interrupt controller. Do not set these bits to other values than those show
  following table.

SCAN	0			1		
CH2 to CH0	B'100	Other than B'100		B'1**	Other than	
ISS4	0	0	1	0	0	
Pin function	AN4 input pin	P74 input pin	$\overline{\text{ExIRQ4}}$ input pin	AN4 input pin	P74 input pin	

[Legend]

*: Don’t care

- P73/$\overline{\text{ExIRQ3}}$/AN3

  The port 7 function changes as shown in the following table, depending on the comb
  the CH2 to CH0 bits of ADCSR of the A/D converter and the ISS3 bit of ISSR of th
  controller. Do not set these bits to other values than those shown in the following tab

CH2 to CH0	B'011	Other than B'011	
ISS3	0	0	1
Pin function	AN3 input pin	P73 input pin	$\overline{\text{ExIRQ3}}$ input pin

*: Don't care

- P71/AN1

  The port 7 function changes as shown in the following table, depending on the combi
  the SCAN bit and the CH2 to CH0 bits of ADCSR of the A/D converter. Do not set th
  to other values than those shown in the following table.

SCAN	0		1	
CH2 to CH0	B'001	Other than B'001	B'001, B'01*	Other th and
Pin function	AN1 input pin	P71 input pin	AN1 input pin	P71 in

[Legend]

*: Don't care

- P70/AN0

  The port 7 function changes as shown in the following table, depending on the combi
  the SCAN bit and the CH2 to CH0 bits of ADCSR of the A/D converter. Do not set th
  to other values than those shown in the following table.

SCAN	0		1	
CH2 to CH0	B'000	Other than B'000	B'0**	Other th
Pin function	AN0 input pin	P70 input pin	AN0 input pin	P70 in

[Legend]

*: Don't care

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

Bit	Bit Name	Initial Value	R/W	Description
7	P87DDR	0	W	This register is assigned to the same addres
6	P86DDR	0	W	of PBPIN. When this register is read, the po
5	P85DDR	0	W	are read.
4	P84DDR	0	W	If port 8 pins are specified for use as the ge
3	P83DDR	0	W	port, the corresponding port 8 pins are outp
2	P82DDR	0	W	when the P8DDR bits are set to 1, and inpu
1	P81DDR	0	W	when cleared to 0.
0	P80DDR	0	W	

8.8.2 Port 8 Data Register (P8DR)

P8DR stores output data for the port 8 pins.

Bit	Bit Name	Initial Value	R/W	Description
7	P87DR	0	R/W	P8DR stores output data for the port 8 pins
6	P86DR	0	R/W	used as the general output port.
5	P85DR	0	R/W	If a port 8 read is performed while the P8DD
4	P84DR	0	R/W	set to 1, the P8DR values are read. If a port
3	P83DR	0	R/W	performed while the P8DDR bits are cleared
2	P82DR	0	R/W	pin states are read.
1	P81DR	0	R/W	
0	P80DR	0	R/W	

When this pin is used as the P87 output pin, the output format is NMOS push-pull out

P87DDR	0	1
Pin function	P87 input pin	P87 output pin
	$\overline{\text{ExIRQ15}}$ input pin /$\overline{\text{ADTRG}}$ input pin /ExTMIY input pin	

- P86/$\overline{\text{ExIRQ14}}$/SCK2/ExTMIX

  The pin function is switched as shown below according to the combination of the C/$\overline{\text{A}}$ SMR of SCI_2, the CKE1 and CKE0 bits in SCR, and the P86DDR bit.

  When the ISS14 bit in ISSR16 of the interrupt controller is set to 1, this pin can be use $\overline{\text{ExIRQ14}}$ input pin. When the TMIXS bit in PTCNT0 is set to 1, this pin can be used TMIX (TMCIX/TMRIX) input pin. To use this pin as the $\overline{\text{ExIRQ14}}$ input pin, clear th P86DDR bit to 0.

  When this pin is used as the P86 output pin, the output format is NMOS push-pull out

CKE1	0				
C/$\overline{\text{A}}$	0			1	
CKE0	0		1	—	
P86DDR	0	1	—	—	
Pin function	P86 input pin	P86 output pin	SCK2 output pin	SCK2 output pin	SCK2
	$\overline{\text{ExIRQ14}}$ input pin /ExTMIX input pin				

P85DDR	0	1	—	—	
Pin function	P85 input pin	P85 output pin	SCK1 output pin	SCK1 output pin	SC
	$\overline{\text{ExIRQ13}}$ input pin /ExTMI1 input pin				

- P84/$\overline{\text{ExIRQ12}}$/SCK0/ExTMI0

  The pin function is switched as shown below according to the combination of the C/
  SMR of SCI_0, the CKE1 and CKE0 bits in SCR, and the P84DDR bit.

  When the ISS12 bit in ISSR16 of the interrupt controller is set to 1, this pin can be u
  $\overline{\text{ExIRQ12}}$ input pin. When the TMI0S bit in PTCNT0 is set to 1, this pin can be used
  TMI0 (TMI0/TMRI0) input pin. To use this pin as the $\overline{\text{ExIRQ12}}$ input pin, clear the
  bit to 0.

  When this pin is used as the P84 output pin, the output format is NMOS push-pull ou

CKE1	0				
C/$\overline{\text{A}}$	0			1	
CKE0	0		1	—	
P84DDR	0	1	—	—	
Pin function	P84 input pin	P84 output pin	SCK0 output pin	SCK0 output pin	SC
	$\overline{\text{ExIRQ12}}$ input pin /ExTMI0 input pin				

- P82/$\overline{\text{ExIRQ10}}$/SCL1

  The pin function is switched as shown below according to the combination of the ICE
  ICCR of IIC_1 and the P82DDR bit.

  When the ISS10 bit in ISSR16 of the interrupt controller is set to 1, this pin can be us
  $\overline{\text{ExIRQ10}}$ input pin. To use this pin as the $\overline{\text{ExIRQ10}}$ input pin, clear the P82DDR bit t

  When this pin is used as the P82 output pin, the output format is NMOS push-pull out
  output format for SCL1 is NMOS open-drain output, and direct bus drive is possible.

ICE	0		1
P82DDR	0	1	—
Pin function	P82 input pin	P82 output pin	SCL1 input/ou
	$\overline{\text{ExIRQ10}}$ input pin		

- P81/$\overline{\text{ExIRQ9}}$/SDA0

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

  When the ISS9 bit in ISSR16 of the interrupt controller is set to 1, this pin can be used
  $\overline{\text{ExIRQ9}}$ input pin. To use this pin as the $\overline{\text{ExIRQ9}}$ input pin, clear the P81DDR bit to 0

  When this pin is used as the P81 output pin, the output format is NMOS push-pull out
  output format for SDA0 is NMOS open-drain output, and direct bus drive is possible.

ICE	0		1
P81DDR	0	1	—
Pin function	P81 input pin	P81 output pin	SDA0 input/ou
	$\overline{\text{ExIRQ9}}$ input pin		

Bit	Bit Name	Initial Value	R/W	Description
7	P97DDR	0	W	If port 9 pins are specified for use as the gen port, the corresponding port 9 pins are outpu when the P9DDR bits are set to 1, and input when cleared to 0.
6	P96DDR	0	W	When this bit is set to 1, the corresponding p is the system clock output pin (φ), and as a g input port when cleared to 0.
5	P95DDR	0	W	If port 9 pins are specified for use as the gen port, the corresponding port 9 pins are outpu when the P9DDR bits are set to 1, and input when cleared to 0.
4	P94DDR	0	W	
3	P93DDR	0	W	
2	P92DDR	0	W	
1	P91DDR	0	W	
0	P90DDR	0	W	

1	P91DR	0	R/W
0	P90DR	0	R/W

Note: The initial value of bit 6 is determined in accordance with the P96 pin state.

8.9.3 Pin Functions

The relationship between the operating mode, register setting values, and pin functions a
follows.

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

  The pin function is switched as shown below according to the combination of the op
  mode, the CS256E bit in SYSCR, the WMS1 bit in WSCR, the WMS21 bit in WSC
  the P97DDR bit.

Operating Mode	Extended Mode				Single-Chi	
WMS1, WMS21	All 0			One bit is set as 1	—	
CS256E	0		1	—	—	
P97DDR	0	1	—	—	0	
Pin function	P97 input pin	P97 output pin	$\overline{\text{CS256}}$ output pin	$\overline{\text{WAIT}}$ input pin	P97 input pin	

Operating Mode	Extended Mode		Single-Chip Mode	
P95DDR	—		0	
IOSE	0	1	—	
Pin function	$\overline{AS}$ output pin	$\overline{IOS}$ output pin	P95 input pin	P95 ou

- P94/$\overline{HWR}$

  The pin function is switched as shown below according to the combination of the ope mode and the P94DDR bit.

Operating Mode	Extended Mode	Single-Chip Mode	
P94DDR	—	0	1
Pin function	$\overline{HWR}$ output pin	P94 input pin	P94 output

- P93/$\overline{RD}$

  The pin function is switched as shown below according to the combination of the ope mode and the P93DDR bit.

Operating Mode	Extended Mode	Single-Chip Mode	
P93DDR	—	0	1
Pin function	$\overline{RD}$ output pin	P93 input pin	P93 outp

mode, the ADMXE bit of SYSCR2, and the P91DDR bit.

Operating Mode	Extended Mode			Single-Chip Mo	
ADMXE	0		1	—	
P91DDR	0	1	—	0	
Pin function	P91 input pin	P91 output pin	$\overline{AH}$ output pin	P91 input pin	P91

- P90/$\overline{LWR}$

  The pin function is switched as shown below according to the combination of the op
  mode, the ABW and ABW256 bits in WSCR, the ABWCP bit in BCR2, and the P90

Operating Mode	Extended Mode			Single-Chip Mo	
ABW, ABW256, ABWCP	All 1		One bit is set as 0	—	
P90DDR	0	1	—	0	
Pin function	P90 input pin	P90 output pin	$\overline{LWR}$ output pin	P90 input pin	P90

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

Bit	Bit Name	Initial Value	R/W	Description
7	PA7DDR	0	W	In normal extended mode:
6	PA6DDR	0	W	The corresponding port A pins are address o
5	PA5DDR	0	W	ports when the PADDR bits are set to 1, and
4	PA4DDR	0	W	ports when cleared to 0. Pins function as the
3	PA3DDR	0	W	output port depending on the setting of bits I
2	PA2DDR	0	W	CS256E, CPCSE, ADFULLE in bus controlle
1	PA1DDR	0	W	In other mode:
0	PA0DDR	0	W	The corresponding port A pins are output por the PADDR bits are set to 1, and input ports cleared to 0.

1	PA1ODR	0	R/W	
0	PA0ODR	0	R/W	

8.10.3 Port A Input Data Register (PAPIN)

PAPIN indicates the pin states.

Bit	Bit Name	Initial Value	R/W	Description
7	PA7PIN	Undefined*	R	When a PAPIN read is performed, the pin s
6	PA6PIN	Undefined*	R	always read.
5	PA5PIN	Undefined*	R	
4	PA4PIN	Undefined*	R	
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

PA4/KIN12/EVENT4/A20, PA3/KIN11/EVENT3/A19, PA2/KIN10/EVENT2/A18

The function of port A pins is switched according to the combination of address 18 se the PAnDDR bit. When the KMIM bit in KMIMRA of the interrupt controller is clear this pin can be used as the $\overline{\text{KIN}}$ input pin. To use this pin as the $\overline{\text{KIN}}$ input pin, clear t PAnDDR bit to 0. When this pin is used as EVENT input pin according to bits ECSB ECSB0 in ECCR of the data transfer controller settings, clear the PAnDDR bit to 0. T this pin has been set to the EVENT input pin, to use as the PAn or A1 output pin, set t PAnDDR bit to 1.

PAnDDR	0	1	1
Address 18	1	1	0
Pin function	PAn input pins	PAn output pin	AI output
	$\overline{\text{KINm}}$ input pin EVENTn input pin		

[Legend]
n = 7 to 2
m = 15 to 10
l = 23 to 18

- PA1/$\overline{\text{KIN9}}$/EVENT1/A17/SSE2I

  The function of port A pins is switched as shown below according to the combination SSE bit in SEMR of SCI_2, the C/$\overline{\text{A}}$ bit in SMR, the CKE1 bit in SCR, address 13 set the PA1DDR bit.

  When the KMIM9 bit in KMIMRA of the interrupt controller is cleared to 0, this pin used as the $\overline{\text{KIN9}}$ input pin. To use this pin as the $\overline{\text{KIN9}}$ input pin, clear the PA1DDR When this pin is used as EVENT1 input pin according to bits ECSB3 to ECSB0 in EC the data transfer controller settings, clear the PA1DDR bit to 0. Though this pin has b the EVENT1 input pin, to use as the PA1 or A17 output pin, set the PA1DDR bit to 1

SSE bit in SEMR of SCI_0, the C/A bit in SMR, the CKE1 bit in SCR, address 13 se
the PA0DDR bit.

When the KMIM8 bit in KMIMRA of the interrupt controller is cleared to 0, this pin
used as the $\overline{\text{KIN8}}$ input pin. To use this pin as the $\overline{\text{KIN8}}$ input pin, clear the PA0DDR
When this pin is used as EVENT0 input pin according to bits ECSB3 to ECSB0 in E
the data transfer controller settings, clear the PA0DDR bit to 0. Though this pin has
the EVENT0 input pin, to use as the PA0 or A16 output pin, set the PA0DDR bit to

SSE	0			
C/$\overline{\text{A}}$	—			
CKE1	—			
PA0DDR	0	1	1	
Address 13	1		0	
Pin function	PA0 input pin	PA0 output pin	A16 output pin	SSE0
	$\overline{\text{KIN8}}$ input pin /EVENT0 input pin			

Single-Chip Mode and Address-Data Multiplex Extended Mode: Port A functions as
input, external control input of SCI_0 and SCI_2, and also as an I/O port, and input or o
be specified in bit units.

- PA7/$\overline{\text{KIN15}}$/EVENT7, PA6/$\overline{\text{KIN14}}$/EVENT6, PA5/$\overline{\text{KIN13}}$/EVENT5, PA4/$\overline{\text{KIN12}}$/E
  PA3/$\overline{\text{KIN11}}$/EVENT3, PA2/$\overline{\text{KIN10}}$/EVENT2

  When the KMIM bit in KMIMRA of the interrupt controller is cleared to 0, this pin
  used as the $\overline{\text{KIN}}$ input pin. To use this pin as the $\overline{\text{KIN}}$ input pin, clear the PAnDDR b
  When this pin is used as the EVENT input pin according to bits ECSB3 to ECSB0 in
  the data transfer controller settings, clear the PAnDDR bit to 0. Though this pin has
  the EVENT input pin, to use as the PAn output pins, set the PAnDDR bit to 1.

When this pin is used as the EVENT1 input pin according to bits ECSB3 to ECSB0 in
of the data transfer controller settings, clear the PA1DDR bit to 0. Though this pin has
to the EVENT1 input pin, to use as the PA1 output pin, set the PA1DDR bit to1.

SSE	0		1
C/$\overline{A}$	—		1
CKE1	—		1
PA1DDR	0	1	—
Pin function	PA1 input pin	PA1 output pin	SSE2I input
	$\overline{KIN9}$ input pin /EVENT1 input pin		

- PA0/$\overline{KIN8}$/EVENT0/SSE0I

  The function of port A pins is switched as shown below according to the combination
  SSE bit in SEMR of SCI_0, the C/$\overline{A}$ bit in SMR, the CKE1 bit in SCR, and the PA0D

  When the KMIM8 bit in KMIMRA of the interrupt controller is cleared to 0, this pin
  used as the $\overline{KIN8}$ input pin. To use this pin as the $\overline{KIN8}$ input pin, clear the PA0DDR
  When this pin is used as the EVENT0 input pin according to bits ECSB3 to ECSB0 in
  of the data transfer controller settings, clear the PA0DDR bit to 0. Though this pin has
  to the EVENT0 input pin, to use as the PA0 output pin, set the PA0DDR bit to1.

MOS can be used in any operating mode, and can be specified as on or off on a bit-by-b

PAnDDR	0		1
PAnODR	1	0	—
PAn pull-up MOS	ON	OFF	OFF

[Legend]
n = 7 to 0

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 8.6 summarizes the input pull-up MOS states.

Table 8.6 Port A 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.

Bit	Bit Name	Initial Value	R/W	Description
7	PB7DDR	0	W	The corresponding port B pins are output por
6	PB6DDR	0	W	the PBDDR bits are set to 1, and input ports
5	PB5DDR	0	W	cleared to 0.
4	PB4DDR	0	W	
3	PB3DDR	0	W	
2	PB2DDR	0	W	
1	PB1DDR	0	W	
0	PB0DDR	0	W	

8.11.2 Port B Output Data Register (PBODR)

PBODR stores output data for the port B pins.

Bit	Bit Name	Initial Value	R/W	Description
7	PB7ODR	0	R/W	The PBODR register stores the output data f
6	PB6ODR	0	R/W	pins that are used a general output port.
5	PB5ODR	0	R/W	
4	PB4ODR	0	R/W	
3	PB3ODR	0	R/W	
2	PB2ODR	0	R/W	
1	PB1ODR	0	R/W	
0	PB0ODR	0	R/W	

0	PB0PIN	Undefined*	R

Note: The initial value of these pins is determined in accordance with the state of pins P PB0.

8.11.4 Pin Functions

Port B is a multi-function port that can function as an event counter input pin. The relati between the operating mode setup and pin functions is described below.

When this pin is used as the EVENT input pin according to bits ECSB3 to ECSB0 in EC data transfer controller settings, clear the PBnDDR bit to 0. (n = 7 to 0)

- PB7/EVENT15

PB7DDR	0		1
Event counter*1	Disable	Enable	—
Pin function	PB7 input pin	EVENT15 input pin	PB7 output pin

- PB6/EVENT14

PB6DDR	0		1
Event counter*1	Disable	Enable	—
Pin function	PB6 input pin	EVENT14 input pin	PB6 output pin

- PB3/EVENT11

PB3DDR	0		1
Event counter*[1]	Disable	Enable	—
Pin function	PB3 input pin	EVENT11 input pin	PB3 output pin

- PB2/EVENT10

PB2DDR	0		1
Event counter*[1]	Disable	Enable	—
Pin function	PB2 input pin	EVENT10 input pin	PB2 output pin

- PB1/EVENT9

PB1DDR	0		1
Event counter*[1]	Disable	Enable	—
Pin function	PB1 input pin	EVENT9 input pin	PB1 output pin

- PB0/EVENT8

PB0DDR	0		1
Event counter*[1]	Disable	Enable	—
Pin function	PB0 input pin	EVENT8 input pin	PB0 output pin

Note: For event counter setting, refer to section 7, Data Transfer Controller (DTC).

PCDDR is used to specify the input/output attribute of each pin of port C.

Bit	Bit Name	Initial Value	R/W	Description
7	PC7DDR	0	W	When a given bit is set to 1, the correspond function as an output port, and when cleare functions as an input port. This register is assigned to the same addres of PCPIN. When this address is read, the p states are returned.
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	

8.12.2 Port C Output Data Register (PCODR)

PCODR stores output data for port C.

Bit	Bit Name	Initial Value	R/W	Description
7	PC7ODR	0	R/W	The PCODR register stores the output data pins that are used as a general output port.
6	PC6ODR	0	R/W	
5	PC5ODR	0	R/W	
4	PC4ODR	0	R/W	
3	PC3ODR	0	R/W	
2	PC2ODR	0	R/W	
1	PC1ODR	0	R/W	
0	PC0ODR	0	R/W	

0	PC0 PIN	Undefined*[1]	R

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

8.12.4 Pin Functions

Port C is capable of functioning as the input and output of IIC_2, IIC_3, and IIC_4, and t
PWMX output. The relationship between the register settings and pin function is describe

- PC7/PWX3

  The pin function is switched as shown below according to the combination of the OE
  the 14-bit PWMX DACR and the PC7DDR.

OEB	0		1
PC7DDR	0	1	—
Pin Function	PC7 input pin	PC7 output pin	PWX3 outpu

- PC6/PWX2

  The pin function is switched as shown below according to the combination of the OE
  the 14-bit PWMX DACR and the PC6DDR.

OEA	0		1
PC6DDR	0	1	—
Pin Function	PC6 input pin	PC6 output pin	PWX2 outpu

ICE	0		1
PC4DDR	0	1	—
Pin Function	PC4 input pin	PC4 output pin	SCL4 input/ou

- PC3/SDA3

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

ICE	0		1
PC3DDR	0	1	—
Pin Function	PC3 input pin	PC3 output pin	SDA3 input/ou

- PC2/SCL3

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

ICE	0		1
PC2DDR	0	1	—
Pin Function	PC2 input pin	PC2 output pin	SCL3 input/ou

- PC1/SDA2

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

ICE	0		1
PC1DDR	0	1	—
Pin Function	PC1 input pin	PC1 output pin	SDA2 input/ou

- Port D data direction register (PDDDR)
- Port D output data register (PDODR)
- Port D input data register (PDPIN)

8.13.1 Port D Data Direction Register (PDDDR)

PDDDR is used to specify the input/output attribute of each pin of port D.

Bit	Bit Name	Initial Value	R/W	Description
7	PD7DDR	0	W	When the general input/output port function is selected, and the given bit is set to 1, the corresponding pin will function as an output p when the bit is cleared to 0, the pin will functi input port. This register is assigned to the same address of PDPIN. When this address is read, the por states are returned.
6	PD6DDR	0	W	
5	PD5DDR	0	W	
4	PD4DDR	0	W	
3	PD3DDR	0	W	
2	PD2DDR	0	W	
1	PD1DDR	0	W	
0	PD0DDR	0	W	

1	PD1ODR	0	R/W	
0	PD0ODR	0	R/W	

8.13.3 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	Pin states can be read by performing a read
6	PD6PIN	Undefined*	R	this register.
5	PD5PIN	Undefined*	R	This register is assigned to the same addres
4	PD4PIN	Undefined*	R	of PDDDR. When this register is written to,
3	PD3PIN	Undefined*	R	written to PDDDR and the port D setting is t
2	PD2PIN	Undefined*	R	changed.
1	PD1PIN	Undefined*	R	
0	PD0PIN	Undefined*	R	

Note: The initial value of these pins is determined in accordance with the state of pins P
PD0.

PD7DDR	0	1	—
Pin Function	PD7 input pin	PD7 output pin	SDA5 input/ou

- PD6/SCL5

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

ICE	0		1
PD6DDR	0	1	—
Pin Function	PD6 input pin	PD6 output pin	SCL5 input/ou

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

  The pin function is switched as shown below according to the combination of LPC enabled/disabled and the PD5DDR.

LPC	Disabled		Enabled
PD5DDR	0	1	0
Pin Function	PD5 input pin	PD5 output pin	$\overline{\text{LPCPD}}$ inpu

- PD4/$\overline{\text{CLKRUN}}$

  The pin function is switched as shown below according to the combination of LPC enabled/disabled and the PD4DDR.

LPC	Disabled		Enabled
PD4DDR	0	1	0
Pin Function	PD4 input pin	PD4 output pin	$\overline{\text{CLKRUN}}$ input/o

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

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

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

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

- PD0/LSCI

  The pin function is switched as shown below according to the combination of the LS
  LPC HICR0 and the PD0DDR.

LSCIE	0		1
PD0DDR	0	1	0
Pin Function	PD0 input pin	PD0 output pin	LSCI outp

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

Table 8.7 summarizes the input pull-up MOS states.

Table 8.7 Port D Input Pull-Up MOS States

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

[Legend]
Off: Always off.
On/Off: On when PDDDR = 0 and PDODR = 1; otherwise off.

Bit	Bit Name	Initial Value	R/W	Description
7	PE7DDR	0	W	When a given bit of PEDDR is set to 1, the corresponding pin will function as an output when cleared to 0, it will function as an inpu This register is assigned to the same addres of PEPIN. When this address is read, the po states are returned.
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	

8.14.2 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 pins that are used a 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	

0	PE0PIN	Undefined*	R

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

8.14.4 Pin Functions

Port E also functions as an LPC input/output. The pin function is switched with LPC enal
disabled. The LPC module is disabled when the LPC1E, LPC2E, and LPC3E bits in HIC
LPC are all 0.

- PE7/SERIRQ

  The pin function is switched as shown below according to the LPC enabled/disabled a
  PE7DDR.

LPC	Disabled		Enabled
PE7DDR	0	1	—
Pin Function	PE7 input pin	PE7 output pin	SERIRQ input/o

- PE6/LCLK

  The pin function is switched as shown below according to the LPC enabled/disabled a
  PE6DDR.

LPC	Disabled		Enabled
PE6DDR	0	1	—
Pin Function	PE6 input pin	PE6 output pin	LCLK input

LPC	Disabled		Enable
PE4DDR	0	1	—
Pin Function	PE4 input pin	PE4 output pin	$\overline{\text{LFRAME}}$ in

- PE3/LAD3

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

LPC	Disabled		Enable
PE3DDR	0	1	—
Pin Function	PE3 input pin	PE3 output pin	LAD3 input/o

- PE2/LAD2

  The pin function is switched as shown below according to the LPC enabled/disabled
  PE2DDR.

LPC	Disabled		Enable
PE2DDR	0	1	—
Pin Function	PE2 input pin	PE2 output pin	LAD2 input/o

- PE1/LAD1

  The pin function is switched as shown below according to the LPC enabled/disabled
  PE1DDR.

LPC	Disabled		Enable
PE1DDR	0	1	—
Pin Function	PE1 input pin	PE1 output pin	LAD1 input/o

- Port F data direction register (PFDDR)
- Port F output data register (PFODR)
- Port F input data register (PFPIN)

8.15.1 Port F Data Direction Register (PFDDR)

PFDDR is used to specify the input/output attribute of each pin of port F.

Bit	Bit Name	Initial Value	R/W	Description
7 to 3	—	—	—	Reserved
2	PF2DDR	0	W	When the given bit of PFDDR is set to 1, the corresponding pin of port F will function as an port, and when the bit is cleared to 0, the por function as an input port. This register is assigned to the same address of PFPIN. When this address is read, the por are returned.
1	PF1DDR	0	W	
0	PF0DDR	0	W	

8.15.2 Port F Output Data Register (PFODR)

PFODR stores output data for the port F pins.

Bit	Bit Name	Initial Value	R/W	Description
7 to 3	—	—	—	Reserved. When this bit is read, an undefine returned.
2	PF2ODR	0	R/W	The PFODR register stores the output data f pins that are used a general output port.
1	PF1ODR	0	R/W	
0	PF0ODR	0	R/W	

PF0.

8.15.4 Pin Functions

Port F is a 3-bit input/output port that functions as a PWM output. The relationship betw
register settings and pin functions is depicted below.

- PF2/ExPW2, PF1/ExPW1, PF0/ExPW0

  The pin function is switched as shown below according to the combination of the OE
  PWOERA of PWM, the PWMS bit in PTCNT0, and PFnDDR bit.

PFnDDR	0		1		
PWMS	0	1	0	1	
OEn	—	0	—	0	
Pin Function	PFn input pin		PFn output pin		PWn ou

[Legend]
n = 2 to 0

ISSR16 and ISSR select ports that also function as $\overline{\text{IRQ15}}$ to $\overline{\text{IRQ0}}$ input pins.

- ISSR16

Bit	Bit Name	Initial Value	R/W	Description
15	ISS15	0	R/W	0: P57/$\overline{\text{IRQ15}}$ is selected 1: P87/$\overline{\text{ExIRQ15}}$ is selected
14	ISS14	0	R/W	0: P56/$\overline{\text{IRQ14}}$ 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: P83/$\overline{\text{ExIRQ11}}$ is selected
10	ISS10	0	R/W	0: P52/$\overline{\text{IRQ10}}$ is selected 1: P82/$\overline{\text{ExIRQ10}}$ is selected
9	ISS9	0	R/W	0: P51/$\overline{\text{IRQ9}}$ is selected 1: P81/$\overline{\text{ExIRQ9}}$ is selected
8	ISS8	0	R/W	0: P50/$\overline{\text{IRQ8}}$ is selected 1: P80/$\overline{\text{ExIRQ8}}$ is selected

				1: P73/$\overline{\text{ExIRQ3}}$ is selected
2	ISS2	0	R/W	0: P42/$\overline{\text{IRQ2}}$ is selected 1: P72/$\overline{\text{ExIRQ2}}$ is selected
1	ISS1	0	R/W	P41/$\overline{\text{IRQ1}}$ is always selected
0	ISS0	0	R/W	P40/$\overline{\text{IRQ0}}$ is always selected

4	TMIYS	0	R/W	0: P45/TMIY is selected 1: P87/ExTMIY is selected
3	—	0	R/W	Reserved The initial values should not be changed.
2	PWMS	0	R/W	0: P10/PW0, P11/PW1, P12/PW2 are sele 1: PF0/ExPW0, PF1/ExPW1, and PF2/Ex selected
1, 0	—	All 0	R/W	Reserved The initial values should not be changed.

Figure 9.1 shows a block diagram of the PWM timer.

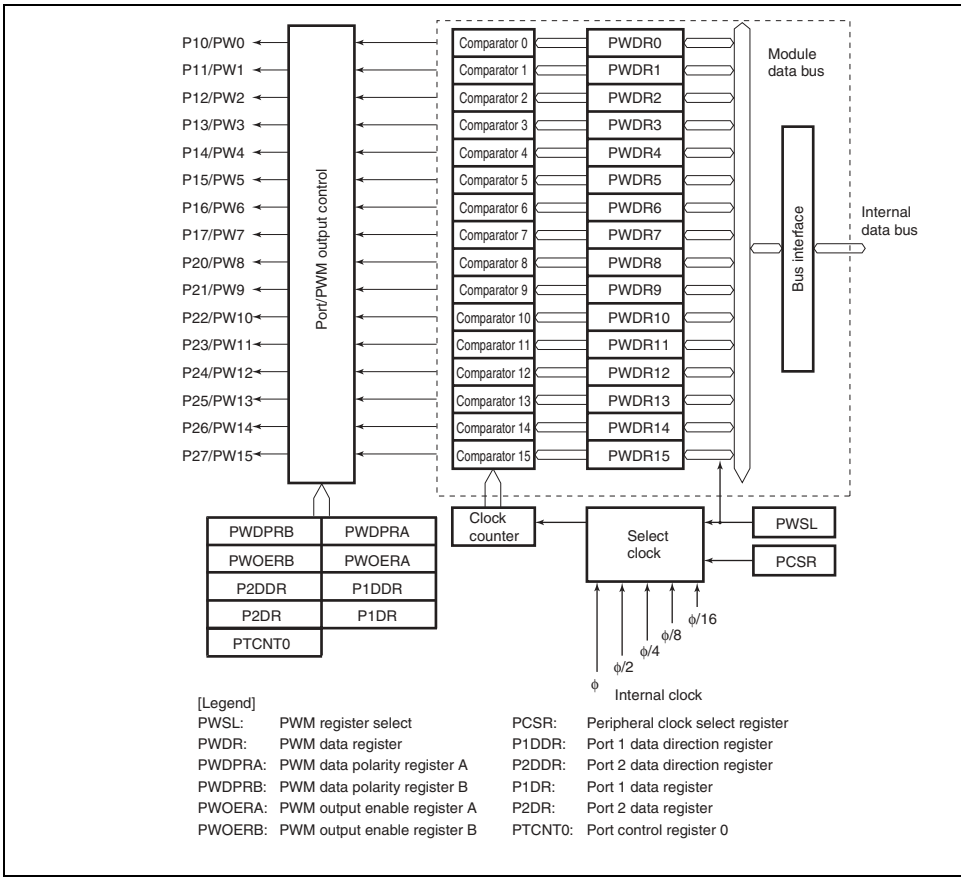


Figure 9.1 Block Diagram of PWM Timer

register (STCR) must be cleared to 0. For details on the serial timer control register (STCR), see section 3.2.3, Serial Timer Control Register (STCR).

- PWM register select (PWSL)
- PWM data registers 15 to 0 (PWDR15 to PWDR0)
- PWM data polarity register A (PWDPRA)
- PWM data polarity register B (PWDPRB)
- PWM output enable register A (PWOERA)
- PWM output enable register B (PWOERB)
- Peripheral clock select register (PCSR)

				from the following equations. Resolution (minimum pulse width) = 1/internal clock PWM conversion period = resolution × 256 Carrier frequency = 16/PWM conversion period With a 33 MHz system clock (ϕ), the resolution, PWM conversion period, and carrier frequency are as shov 9.3.
5	—	1	R	Reserved This bit is always read as 1 and cannot be modified.
4	—	0	R	Reserved This bit is always read as 0 and cannot be modified.

0111: PWDR7 selected
1000: PWDR8 selected
1001: PWDR9 selected
1010: PWDR10 selected
1011: PWDR11 selected
1100: PWDR12 selected
1101: PWDR13 selected
1110: PWDR14 selected
1111: PWDR15 selected

Table 9.2 Internal Clock Selection

PWSL		PCSR			
PWCKE	**PWCKS**	**PWCKB**	**PWCKA**	**Description**	
0	—	—	—	Clock input is disabled	(Initi
1	0	—	—	ϕ (system clock) is selected	
	1	0	0	ϕ/2 is selected	
			1	ϕ/4 is selected	
		1	0	ϕ/8 is selected	
			1	ϕ/16 is selected	

PWDR are 8-bit readable/writable registers. The PWM has sixteen PWM data registers. PWDR specifies the duty cycle of the basic pulse to be output, and the number of additi pulses. The value set in PWDR corresponds to a 0 or 1 ratio in the conversion period. Th four bits specify the duty cycle of the basic pulse as 0/16 to 15/16 with a resolution of 1/ lower four bits specify how many extra pulses are to be added within the conversion per comprising 16 basic pulses. Thus, a specification of 0/256 to 255/256 is possible for 0/1 within the conversion period. For 256/256 (100%) output, port output should be used.

9.3.3 PWM Data Polarity Registers A and B (PWDPRA and PWDPRB)

Each PWDPR selects the PWM output phase.

- PWDPRA

Bit	Bit Name	Initial Value	R/W	Description
7 to 0	OS7 to OS0	All 0	R/W	Output Select 7 to 0 These bits select the PWM output phase. Bits OS7 t correspond to outputs PW7 to PW0. 0: PWM direct output (PWDR value corresponds to h of output) 1: PWM inverted output (PWDR value corresponds t width of output)

Each PWOER switches between PWM output and port output.

- PWOERA

Bit	Bit Name	Initial Value	R/W	Description
7 to 0	OE7 to OE0	All 0	R/W	Output Enable 7 to 0 These bits, together with P1DDR, specify the P1n/PW state. Bits OE7 to OE0 correspond to outputs PW7 to P1nDDR OEn: Pin state 0*: Port input 10: Port output or PWM 256/256 output 11: PWM output (0 to 255/256 output)

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

m = 8 to 15
*: Don't care

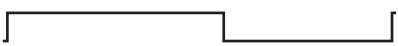
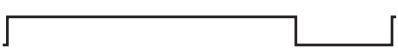
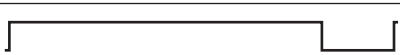
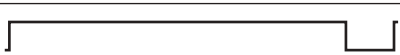
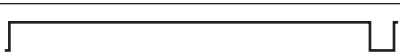
To perform PWM 256/256 output when DDR = 1 and OE = 0, the corresponding pin sh
to port output. The corresponding pin can be set as port output in single-chip mode or w
= 1 and CS256E = 0 in SYSCR in extended mode with on-chip ROM. Otherwise, it sho
noted that an address bus is output to the corresponding pin.

DR data is output when the corresponding pin is used as port output. A value correspond
PWM 256/256 output is determined by the OS bit, so the value should have been set to
beforehand.

9.3.5 Peripheral Clock Select Register (PCSR)

PCSR selects the PWM input clock.

Bit	Bit Name	Initial Value	R/W	Description
7	PWCKX1B	0	R/W	See section 10.3.4, Peripheral Clock Select Regis
6	PWCKX1A	0	R/W	(PCSR).
5	PWCKX0B	0	R/W	
4	PWCKX0A	0	R/W	
3	PWCKX1C	0	R/W	
2	PWCKB	0	R/W	PWM Clock Select B and A
1	PWCKA	0	R/W	Together with bits PWCKE and PWCKS in PWSL bits select the internal clock input to TCNT in the details, see table 9.2.
0	PWCKX0C	0	R/W	See section 10.3.4, Peripheral Clock Select Regis (PCSR).

B'0011	
B'0100	
B'0101	
B'0110	
B'0111	
B'1000	
B'1001	
B'1010	
B'1011	
B'1100	
B'1101	
B'1110	
B'1111	

B'0010							Yes						
B'0011							Yes				Yes		
B'0100			Yes				Yes				Yes		
B'0101			Yes				Yes				Yes		Yes
B'0110			Yes		Yes		Yes				Yes		Yes
B'0111			Yes		Yes		Yes		Yes		Yes		Yes
B'1000		Yes		Yes		Yes		Yes		Yes		Yes	
B'1001		Yes		Yes		Yes		Yes		Yes		Yes	
B'1010		Yes		Yes		Yes	Yes	Yes		Yes		Yes	
B'1011		Yes		Yes		Yes	Yes	Yes		Yes	Yes	Yes	
B'1100		Yes	Yes	Yes		Yes	Yes	Yes		Yes	Yes	Yes	
B'1101		Yes	Yes	Yes		Yes	Yes	Yes		Yes	Yes	Yes	Yes
B'1110		Yes	Yes	Yes	Yes	Yes	Yes	Yes		Yes	Yes	Yes	Yes
B'1111		Yes	Yes	Yes	Yes	Yes	Yes	Yes	Yes	Yes	Yes	Yes	Yes

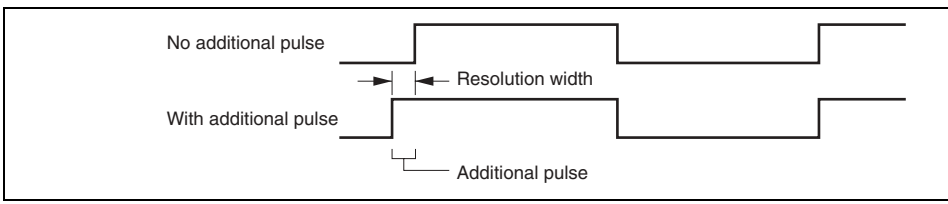


Figure 9.2 Example of Additional Pulse Timing (When Upper 4 Bits of PWDR =

| : Pulse added

Combination of the basic pulse and added pulse outputs 0/256 to 255/256 of duty cycle as low ripple

Figure 9.3 Example of PWM Setting

9.4.2 Diagram of PWM Used as D/A Converter

Figure 9.4 shows the diagram example when using the PWM pulse as the D/A converter. signal with low ripple can be generated by connecting the low pass filter.

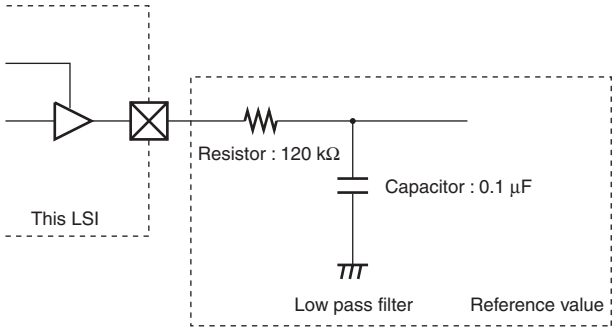


Figure 9.4 Example when PWM is Used as D/A Converter

- 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 10.1 shows a block diagram of the PWM (D/A) module.

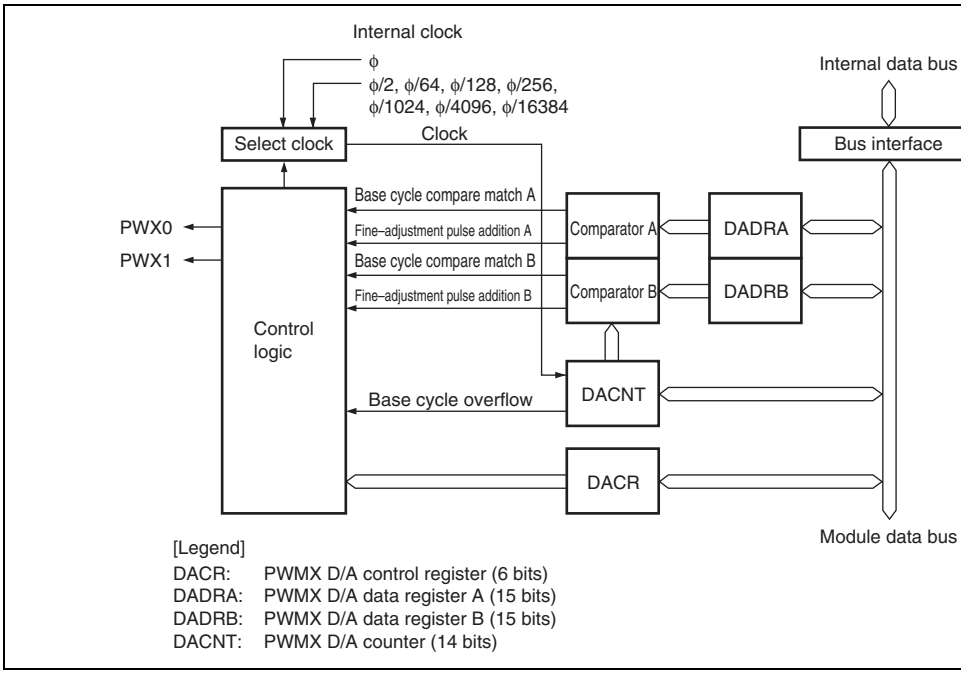


Figure 10.1 PWMX (D/A) Block Diagram

10.3 Register Descriptions

The PWMX (D/A) module has the following registers. The PWMX (D/A) registers are as
the same addresses with other registers. The registers are selected by the IICE bit in the s
timer control register (STCR). For details on the module stop control register, see section
Module Stop Control Register H, L, and A (MSTPCRH, MSTPCRL, 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.

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 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

				In each base cycle, the DACNT value is contin 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 regist be set within a range that depends on the CFS DADR value is outside this range, the PWM ou held constant. A channel can be operated with 12-bit precisio fixing DA0 and DA1 to 0. The two data bits are compared 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'3FF 1: Base cycle = resolution (T) × 256 The range of DA13 to DA0: H'0040 to H'3FF
0	—	1	R	Reserved This bit is always read as 1 and cannot be mod

				hold 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

5, 4		All 1	R	Reserved These bits are always read as 1 and cannot be modified.
3	OEB	0	R/W	Output Enable B Enables or disables output on PWMX (D/A) ch 0: PWMX (D/A) channel B output (at the PWX1 pins) is disabled 1: PWMX (D/A) channel B output (at the PWX1 pins) is enabled
2	OEA	0	R/W	Output Enable A Enables or disables output on PWMX (D/A) ch 0: PWMX (D/A) channel A output (at the PWX0 pin) is disabled 1: PWMX (D/A) channel A output (at the PWX0 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 kind resolution can be selected. 0: Operates at resolution (T) = system clock cy (t_{cyc}) 1: Operates at resolution (T) = system clock cy (t_{cyc}) × 2, × 64, × 128, × 256, × 1024, × 4096, 16384.

				See table 10.2.
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 10.2.
2 1	PWCKB PWCKA	0 0	R/W R/W	PWM Clock Select B and A See section 9.3.5, Peripheral Clock Select Re (PCSR).
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 10.2.

Table 10.2 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

- 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
```

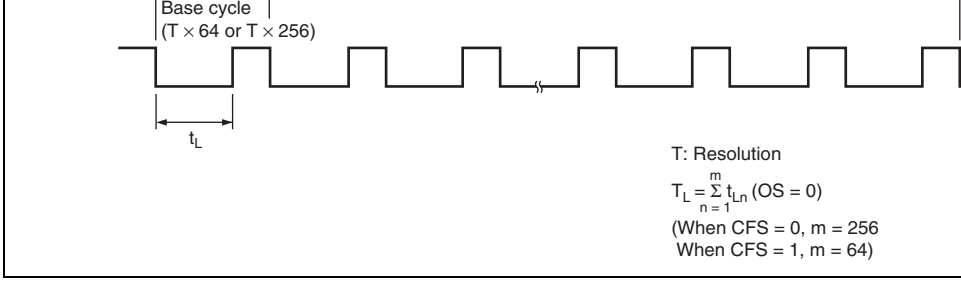


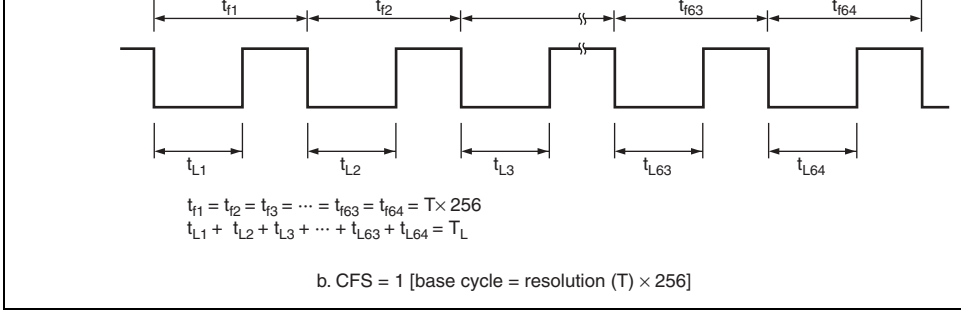
Figure 10.2 PWMX (D/A) Operation

Table 10.3 summarizes the relationships between the CKS and CFS bit settings and the
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 o
waveform is shown in figures 10.3 and 10.4.

				(ϕ)		/128.9kHz		DA13 to 0 = H'0040 to H'3FFF	10		0	0	0	0	
0	0	0	1	0.06	0	3.88	0.99	Always low/high output	14						
						(μs)	(ms)	DA13 to 0 = H'0000 to H'00FF	12				0	0	
								(Data value) × T							
						/257.8kHz		DA13 to 0 = H'0100 to H'3FFF	10		0	0	0	0	
					1	15.52	0.99	Always low/high output	14						
						(μs)	(ms)	DA13 to 0 = H'0000 to H'003F	12				0	0	
								(Data value) × T							
				(ϕ/2)		/64.5kHz		DA13 to 0 = H'0040 to H'3FFF	10		0	0	0	0	
0	0	1	1	1.94	0	124.12	31.78	Always low/high output	14						
						(μs)	(ms)	DA13 to 0 = H'0000 to H'00FF	12				0	0	
								(Data value) × T							
						/8.1kHz		DA13 to 0 = H'0100 to H'3FFF	10		0	0	0	0	
					1	496.48	31.78	Always low/high output	14						
						(μs)	(ms)	DA13 to 0 = H'0000 to H'003F	12				0	0	
								(Data value) × T							
				(ϕ/64)		/2.0kHz		DA13 to 0 = H'0040 to H'3FFF	10		0	0	0	0	
0	1	0	1	3.88	0	248.24	63.55	Always low/high output	14						
						(μs)	(ms)	DA13 to 0 = H'0000 to H'00FF	12				0	0	
								(Data value) × T							
						/4.0kHz		DA13 to 0 = H'0100 to H'3FFF	10		0	0	0	0	
					1	992.97	63.55	Always low/high output	14						
						(μs)	(ms)	DA13 to 0 = H'0000 to H'003F	12				0	0	
								(Data value) × T							
				(ϕ/128)		/1.0kHz		DA13 to 0 = H'0040 to H'3FFF	10		0	0	0	0	

						(ms)	(ms)	DA13 to 0 = H'0000 to H'00FF	12			0	0
								(Data value) × T					
						/503.5Hz		DA13 to 0 = H'0100 to H'3FFF	10	0	0	0	0
					1	7.94	508.40	Always low/high output	14				
						(ms)	(ms)	DA13 to 0 = H'0000 to H'003F	12			0	0
								(Data value) × T					
				(ϕ/1024)		/125.9Hz		DA13 to 0 = H'0040 to H'3FFF	10	0	0	0	0
1	0	1	1	124.12	0	7.94	2.03	Always low/high output	14				
						(ms)	(s)	DA13 to 0 = H'0000 to H'00FF	12			0	0
								(Data value) × T					
						/125.9Hz		DA13 to 0 = H'0100 to H'3FFF	10	0	0	0	0
					1	31.78	2.03	Always low/high output	14				
						(ms)	(s)	DA13 to 0 = H'0000 to H'003F	12			0	0
								(Data value) × T					
				(ϕ/4096)		/31.5Hz		DA13 to 0 = H'0040 to H'3FFF	10	0	0	0	0
1	1	0	1	496.48	0	31.78	8.13	Always low/high output	14				
						(ms)	(s)	DA13 to 0 = H'0000 to H'00FF	12			0	0
								(Data value) × T					
						/31.5Hz		DA13 to 0 = H'0100 to H'3FFF	10	0	0	0	0
					1	127.10	8.13	Always low/high output	14				
						(ms)	(s)	DA13 to 0 = H'0000 to H'003F	12			0	0
								(Data value) × T					
				(ϕ/16384)		/7.9Hz		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.



b. CFS = 1 [base cycle = resolution (T) × 256]

Figure 10.3 Output Waveform (OS = 0, DADR corresponds to T_L)

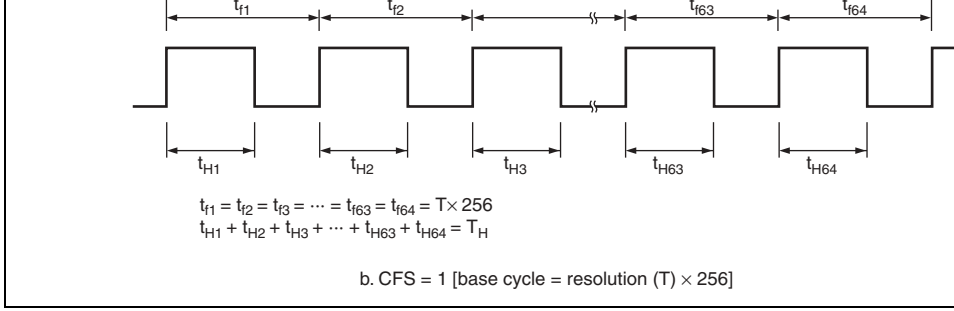


Figure 10.4 Output Waveform (OS = 1, DADR corresponds to T_H)

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 10.5.

Table 10.4 lists the locations of the additional pulses.

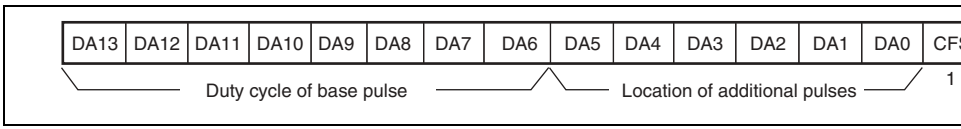


Figure 10.5 D/A Data Register Configuration when CFS = 1

In this example, DADR = H'0207 (B'0000 0010 0000 0111). The output waveform is sh
figure 10.6. 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).

Since the value of the subsequent six bits is B'0000 01, an additional pulse is output onl
location of base pulse No. 63 according to table 10.4. Thus, an additional pulse of 1/256
be added to the base pulse.

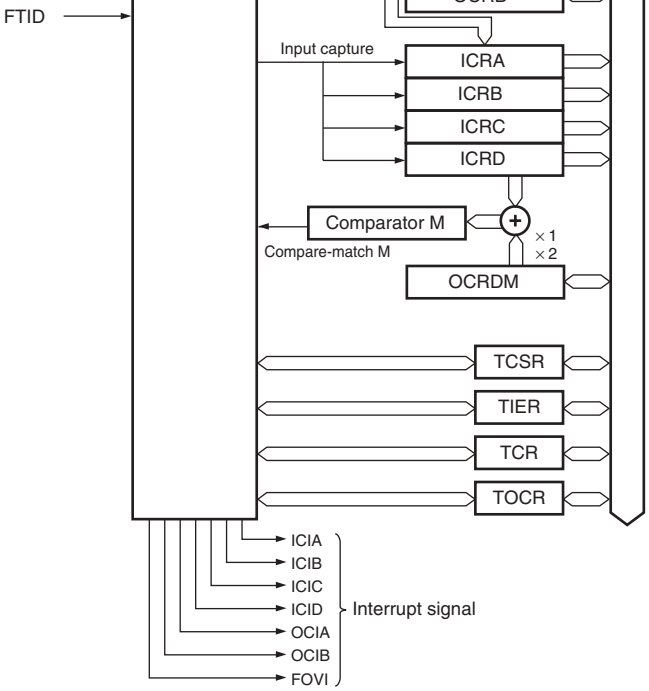
However, when CFS = 0 (base cycle = resolution (T) × 64), the duty cycle of the base pul
determined by the upper six bits and the locations of the additional pulses by the subsequ
bits with a method similar to as above.

Base pulse No.
Lower 6 bits

— Two independent waveforms can be output.

- Four independent input capture channels
  — The rising or falling edge can be selected.
  — Buffer modes can be specified.
- Counter clearing
  — The free-running counters can be cleared on compare-match A.
- Seven independent interrupts
  — Two compare-match interrupts, four input capture interrupts, and one overflow i
  can be requested independently.
- 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
  The contents of ICRD can be added automatically to the contents of OCRDM × 2
  input capture operations in this interval to be restricted.

Figure 11.1 shows a block diagram of the FRT.



[Legend]

OCRA, OCRB:	Output compare register A, B (16-bit)
OCRAR,OCRAF:	Output compare register AR, AF (16-bit)
OCRDM:	Output compare register DM (16-bit)
FRC:	Free-running counter (16-bit)
ICRA to D:	Input capture registers A to D (16-bit)
TCSR:	Timer control/status register (8-bit)
TIER:	Timer interrupt enable register (8-bit)
TCR:	Timer control register (8-bit)
TOCR:	Timer output compare control register (8-bit)

Figure 11.1 Block Diagram of 16-Bit Free-Running Timer

Input capture C input pin	FTIC	Input	Input capture C input
Input capture D input pin	FTID	Input	Input capture D input

11.3 Register Descriptions

The FRT has the following registers.

- Free-running counter (FRC)
- Output compare register A (OCRA)
- Output compare register B (OCRB)
- Input capture register A (ICRA)
- Input capture register B (ICRB)
- Input capture register C (ICRC)
- Input capture register D (ICRD)
- Output compare register AR (OCRAR)
- Output compare register AF (OCRAF)
- Output compare register DM (OCRDM)
- Timer interrupt enable register (TIER)
- 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. ICRA, ICRB, and ICRC share the same addresses with OCRAR, O
and OCRDM. Register selection is controlled by the ICRS bit in TOCR.

match, the output level selected by the OLVLA or OLVLB bit in TOCR is output at the c
compare output pin (FTOA or FTOB). Following a reset, the FTOA and FTOB output lev
until the first compare-match. OCR should always be accessed in 16-bit units; cannot be a
in 8-bit units. OCR is initialized to H'FFFF.

11.3.3 Input Capture Registers A to D (ICRA to ICRD)

The FRT has four input capture registers, ICRA to ICRD, each of which is a 16-bit read-o
register. When the rising or falling edge of the signal at an input capture input pin (FTIA
is detected, the current FRC value is transferred to the corresponding input capture registe
to ICRD). At the same time, the corresponding input capture flag (ICFA to ICFD) in TCS
to 1. The FRC contents are transferred to ICR regardless of the value of ICF. The input ca
edge is selected by the input edge select bits (IEDGA to IEDGD) in TCR.

ICRC and ICRD can be used as ICRA and ICRB buffer registers, respectively, by means
enable bits A and B (BUFEA and BUFEB) in TCR. For example, if an input capture occu
ICRC is specified as the ICRA buffer register, the FRC contents are transferred to ICRA,
transferred to the buffer register ICRC. When IEDGA and IEDGC bits in TCR are set to
values, both rising and falling edges can be specified as the change of the external input s

To ensure input capture, the input capture pulse width should be at least 1.5 system clock
a single edge. When triggering is enabled on both edges, the input capture pulse width sh
least 2.5 system clocks (ϕ).

ICRA to ICRD should always be accessed in 16-bit units; cannot be accessed in 8-bit uni
initialized to H'0000.

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.

11.3.5 Output Compare Register DM (OCRDM)

OCRDM is a 16-bit readable/writable register in which the upper eight bits are fixed at l
When the ICRDMS bit in TOCR is set to 1 and the contents of OCRDM are other than l
the operation of ICRD is changed to include the use of OCRDM. The point at which inp
D occurs is taken as the start of a mask interval. Next, twice the contents of OCRDM is
the contents of ICRD, and the result is compared with the FRC value. The point at whic
values match is taken as the end of the mask interval. New input capture D events are di
during the mask interval. A mask interval is not generated when the contents of OCRDM
H'0000 while the ICRDMS bit is set to 1.

OCRDM should always be accessed in 16-bit units; cannot be accessed in 8-bit units. O
initialized to H'0000.

				Selects whether to enable input capture interru request (ICIB) when input capture flag B (ICFB TCSR is set to 1. 0: ICIB requested by ICFB is disabled 1: ICIB requested by ICFB is enabled
5	ICICE	0	R/W	Input Capture Interrupt C Enable Selects whether to enable input capture interru request (ICIC) when input capture flag C (ICFC TCSR is set to 1. 0: ICIC requested by ICFC is disabled 1: ICIC requested by ICFC is enabled
4	ICIDE	0	R/W	Input Capture Interrupt D Enable Selects whether to enable input capture interru request (ICID) when input capture flag D (ICFD TCSR is set to 1. 0: ICID requested by ICFD is disabled 1: ICID requested by ICFD is enabled
3	OCIAE	0	R/W	Output Compare Interrupt A Enable Selects whether to enable output compare inte request (OCIA) when output compare flag A (O TCSR is set to 1. 0: OCIA requested by OCFA is disabled 1: OCIA requested by OCFA is enabled

				1: FOVI requested by OVF is enabled
0	—	0	R	Reserved This bit is always read as 1 and cannot be m

11.3.7 Timer Control/Status Register (TCSR)

TCSR is used for counter clear selection and control of interrupt request signals.

Bit	Bit Name	Initial Value	R/W	Description
7	ICFA	0	R/(W)*	Input Capture Flag A This status flag indicates that the FRC value transferred to ICRA by means of an input cap signal. When BUFEA = 1, ICFA indicates tha ICRA value has been moved into ICRC and t FRC value has been transferred to ICRA. [Setting condition] When an input capture signal causes the FR be transferred to ICRA [Clearing condition] Read ICFA when ICFA = 1, then write 0 to IC

5	ICFC	0	R/(W)*	Input Capture Flag C This status flag indicates that the FRC value h transferred to ICRC by means of an input capt signal. When BUFEA = 1, on occurrence of an capture signal specified by the IEDGC bit at th input pin, ICFC is set but data is not transferre ICRC. In buffer operation, ICFC can be used a external interrupt signal by setting the ICICE b [Setting condition] When an input capture signal is received [Clearing condition] Read ICFC when ICFC = 1, then write 0 to ICF
4	ICFD	0	R/(W)*	Input Capture Flag D This status flag indicates that the FRC value h transferred to ICRD by means of an input capt signal. When BUFEB = 1, on occurrence of an capture signal specified by the IEDGD bit at th input pin, ICFD is set but data is not transferre ICRD. In buffer operation, ICFD can be used a external interrupt signal by setting the ICIDE b [Setting condition] When an input capture signal is received [Clearing condition] Read ICFD when ICFD = 1, then write 0 to ICF

				When FRC = OCRB
				[Clearing condition]
				Read OCFB when OCFB = 1, then write 0 to
1	OVF	0	R/(W)*	Overflow Flag
				This status flag indicates that the FRC has ov
				[Setting condition]
				When FRC overflows (changes from H'FFFF H'0000)
				[Clearing condition]
				Read OVF when OVF = 1, then write 0 to OV
0	CCLRA	0	R/W	Counter Clear A
				This bit selects whether the FRC is to be clea compare-match A (when the FRC and OCRA match).
				0: FRC clearing is disabled
				1: FRC is cleared at compare-match A

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

				Selects the rising or falling edge of the input ca signal (FTIB). 0: Capture on the falling edge of FTIB 1: Capture on the rising edge of FTIB
5	IEDGC	0	R/W	Input Edge Select C Selects the rising or falling edge of the input ca signal (FTIC). 0: Capture on the falling edge of FTIC 1: Capture on the rising edge of FTIC
4	IEDGD	0	R/W	Input Edge Select D Selects the rising or falling edge of the input ca signal (FTID). 0: Capture on the falling edge of FTID 1: Capture on the rising edge of FTID
3	BUFEA	0	R/W	Buffer Enable A Selects whether ICRC is to be used as a buffe for ICRA. 0: ICRC is not used as a buffer register for ICR 1: ICRC is used as a buffer register for ICRA
2	BUFEB	0	R/W	Buffer Enable B Selects whether ICRD is to be used as a buffe for ICRB. 0: ICRD is not used as a buffer register for ICR 1: ICRD is used as a buffer register for ICRB

between output compare registers A and B, controls the ICRD and OCRA operating mo
switches access to input capture registers A, B, and C.

Bit	Bit Name	Initial Value	R/W	Description
7	ICRDMS	0	R/W	Input Capture D Mode Select Specifies whether ICRD is used in the normal mode or in the operating mode using OCRDM 0: The normal operating mode is specified for 1: The operating mode using OCRDM is spec ICRD
6	OCRAMS	0	R/W	Output Compare A Mode Select Specifies whether OCRA is used in the norma operating mode or in the operating mode usin and OCRAF. 0: The normal operating mode is specified for 1: The operating mode using OCRAR and OC specified for OCRA
5	ICRS	0	R/W	Input Capture Register Select The same addresses are shared by ICRA and by ICRB and OCRAF, and by ICRC and OCR ICRS bit determines which registers are selec the shared addresses are read from or written operation of ICRA, ICRB, and ICRC is not aff 0: ICRA, ICRB, and ICRC are selected 1: OCRAR, OCRAF, and OCRDM are selecte

				1: Output compare A output is enabled
2	OEB	0	R/W	Output Enable B Enables or disables output of the output compa output pin (FTOB). 0: Output compare B output is disabled 1: Output compare B output is enabled
1	OLVLA	0	R/W	Output Level A Selects the level to be output at the output com output pin (FTOA) in response to compare-mat (signal indicating a match between the FRC an values). When the OCRAMS bit is 1, this bit is 0: 0 is output at compare-match A 1: 1 is output at compare-match A
0	OLVLB	0	R/W	Output Level B Selects the level to be output at the output com output pin (FTOB) in response to compare-mat (signal indicating a match between the FRC an values). 0: 0 is output at compare-match B 1: 1 is output at compare-match B

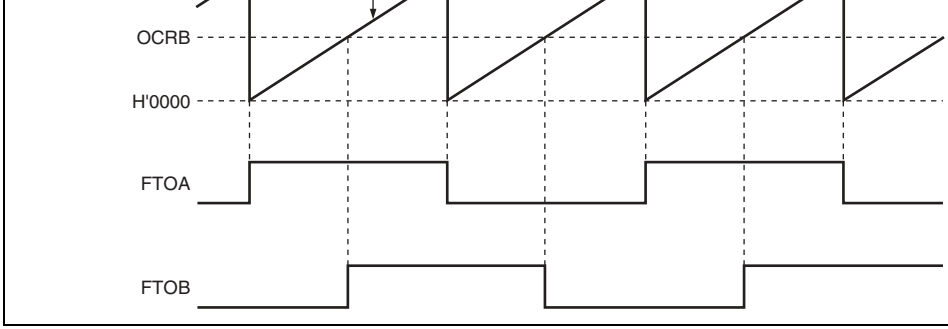


Figure 11.2 Example of Pulse Output

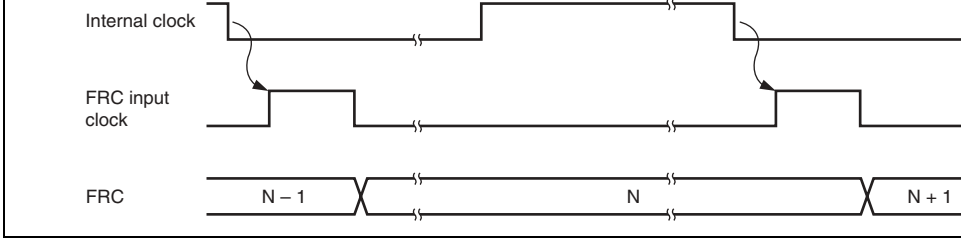


Figure 11.3 Increment Timing with Internal Clock Source

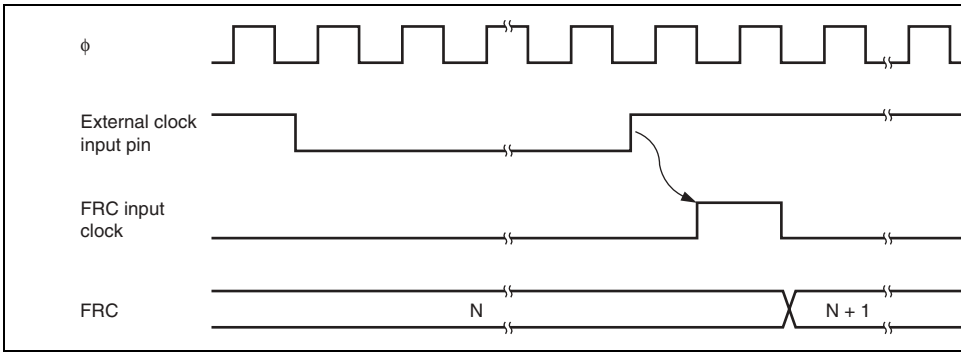


Figure 11.4 Increment Timing with External Clock Source

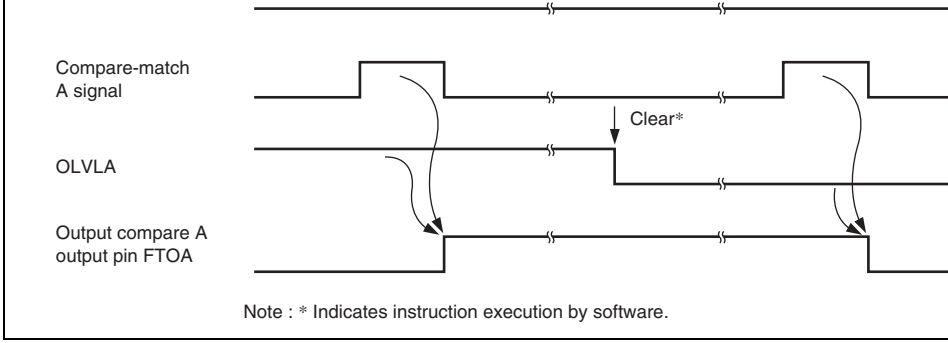


Figure 11.5 Timing of Output Compare A Output

11.5.3 FRC Clear Timing

FRC can be cleared when compare-match A occurs. Figure 11.6 shows the timing of thi
operation.

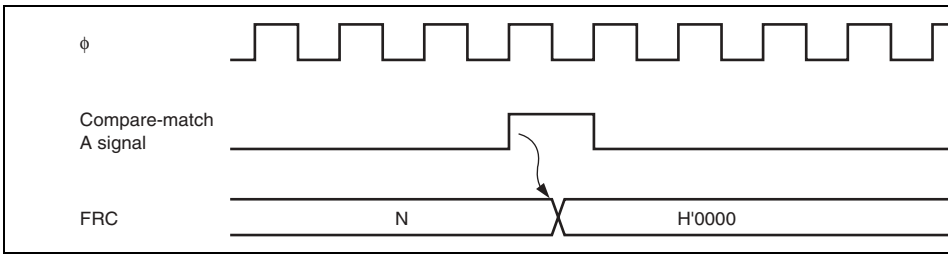


Figure 11.6 Clearing of FRC by Compare-Match A Signal

Figure 11.7 Input Capture Input Signal Timing (Usual Case)

If ICRA to ICRD are read when the corresponding input capture signal arrives, the intern
capture signal is delayed by one system clock (ϕ). Figure 11.8 shows the timing for this c

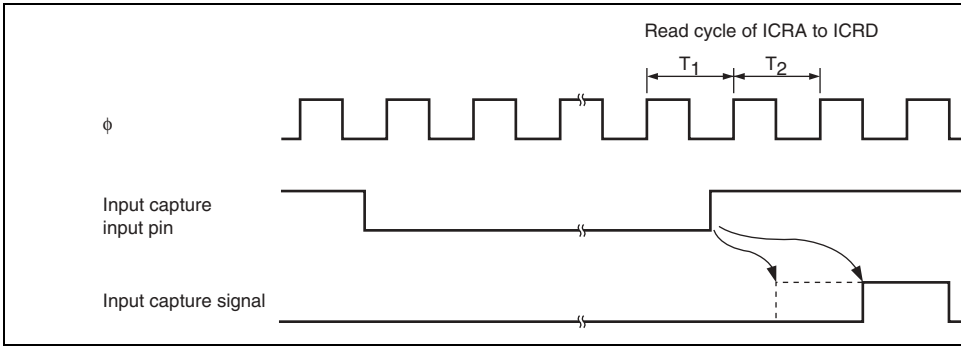


Figure 11.8 Input Capture Input Signal Timing (When ICRA to ICRD is Rea

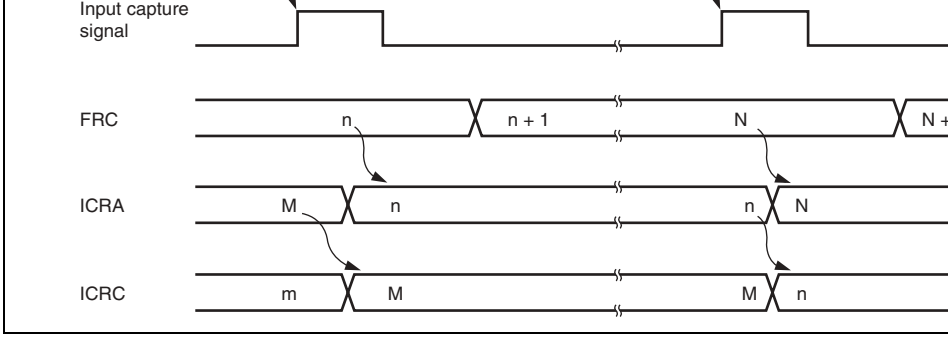


Figure 11.9 Buffered Input Capture Timing

Even when ICRC or ICRD is used as a buffer register, its input capture flag is set by the transition of its input capture signal. For example, if ICRC is used to buffer ICRA, when transition selected by the IEDGC bit occurs on the FTIC input capture line, ICFC will be if the ICICE bit is set at this time, an interrupt will be requested. The FRC value will not transferred to ICRC, however. In buffered input capture, if either set of two registers to will be transferred (ICRA and ICRC, or ICRB and ICRD) is being read when the input c input signal arrives, input capture is delayed by one system clock (ϕ). Figure 11.10 show timing when BUFEA = 1.

11.5.6 Timing of Input Capture Flag (ICF) Setting

The input capture flag, ICFA to ICFD, is set to 1 by the input capture signal. The FRC va simultaneously transferred to the corresponding input capture register (ICRA to ICRD). F 11.11 shows the timing of setting the ICFA to ICFD flag.

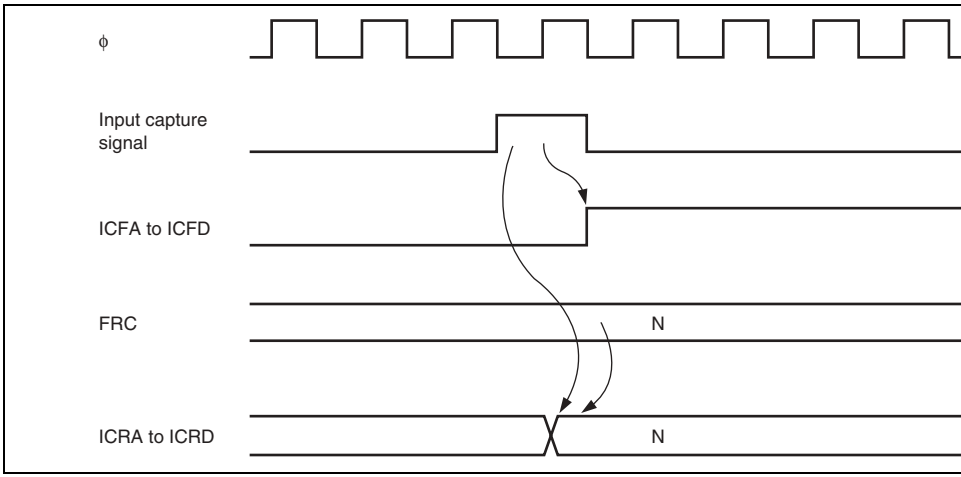


Figure 11.11 Timing of Input Capture Flag (ICFA to ICFD) Setting

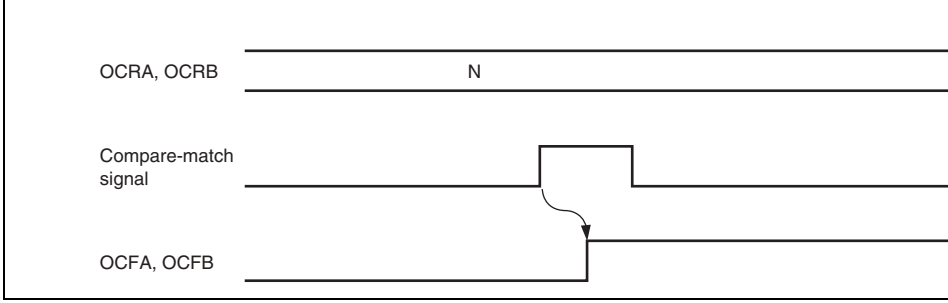


Figure 11.12 Timing of Output Compare Flag (OCFA or OCFB) Setting

11.5.8 Timing of FRC Overflow Flag (OVF) Setting

The FRC overflow flag (OVF) is set to 1 when FRC overflows (changes from H'FFFF to
Figure 11.13 shows the timing of setting the OVF flag.

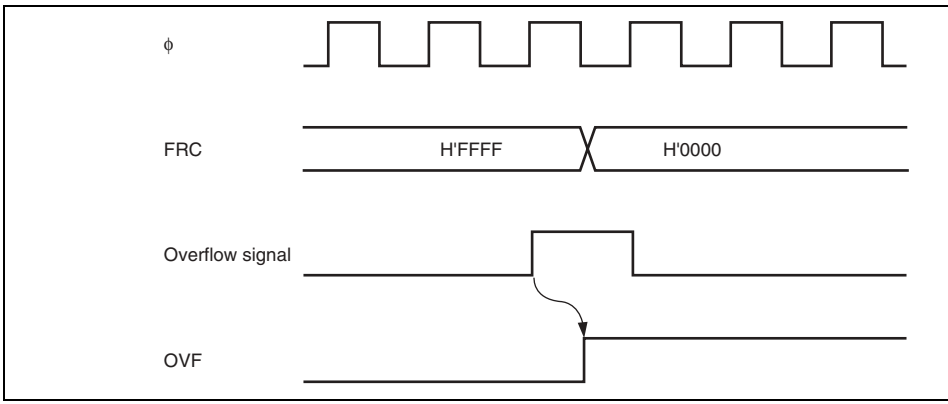


Figure 11.13 Timing of Overflow Flag (OVF) Setting

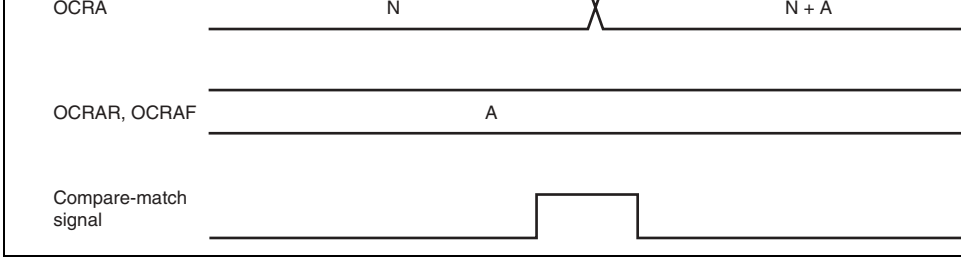


Figure 11.14 OCRA Automatic Addition Timing

11.5.10 Mask Signal Generation Timing

When the ICRDMS bit in TOCR is set to 1 and the contents of OCRDM are other than H signal that masks the ICRD input capture signal is generated. The mask signal is set by th capture signal. The mask signal is cleared by the sum of the ICRD contents and twice the OCRDM contents, and an FRC compare-match. Figure 11.15 shows the timing of setting signal. Figure 11.16 shows the timing of clearing the mask signal.

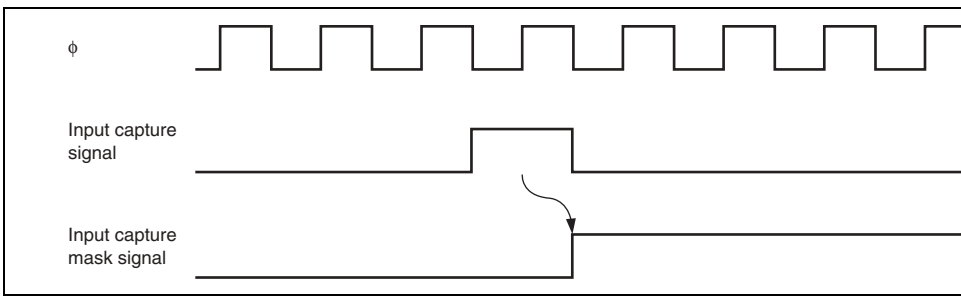


Figure 11.15 Timing of Input Capture Mask Signal Setting

Figure 11.16 Timing of Input Capture Mask Signal Clearing

ICIB	Input capture of ICRB	ICFB	Possible	
ICIC	Input capture of ICRC	ICFC	Not possible	
ICID	Input capture of ICRD	ICFD	Not possible	
OCIA	Compare match of OCRA	OCFA	Possible	
OCIB	Compare match of OCRB	OCFB	Possible	
FOVI	Overflow of FRC	OVF	Not possible	Low

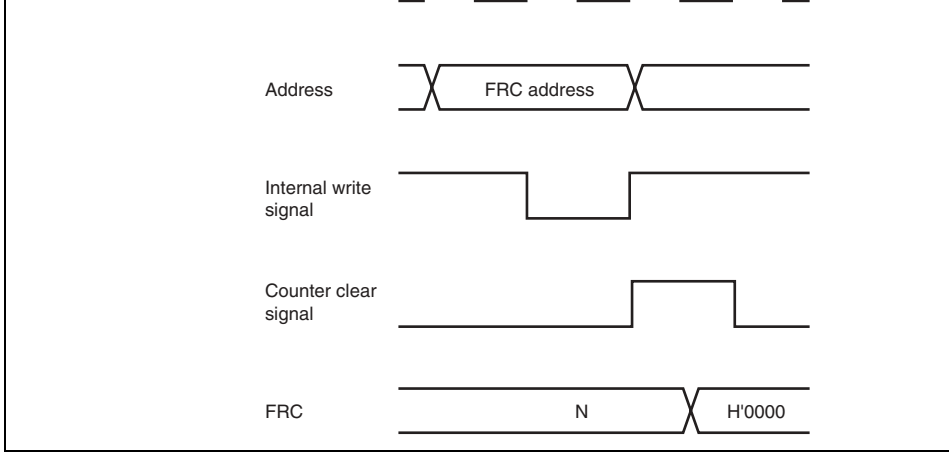


Figure 11.17 Conflict between FRC Write and Clear

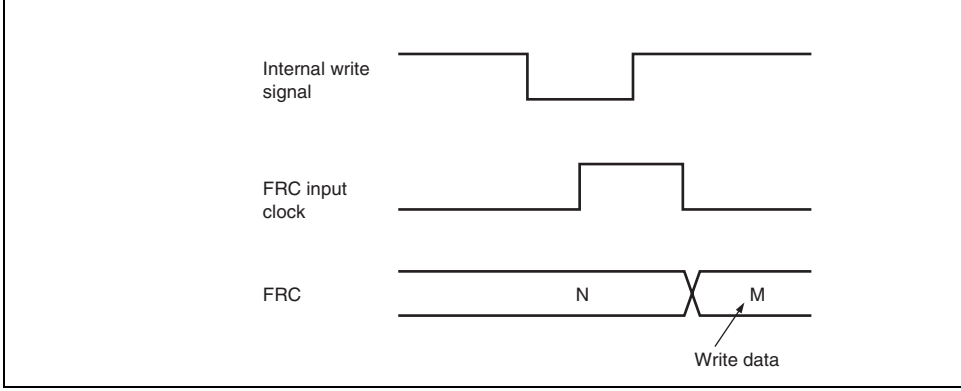


Figure 11.18 Conflict between FRC Write and Increment

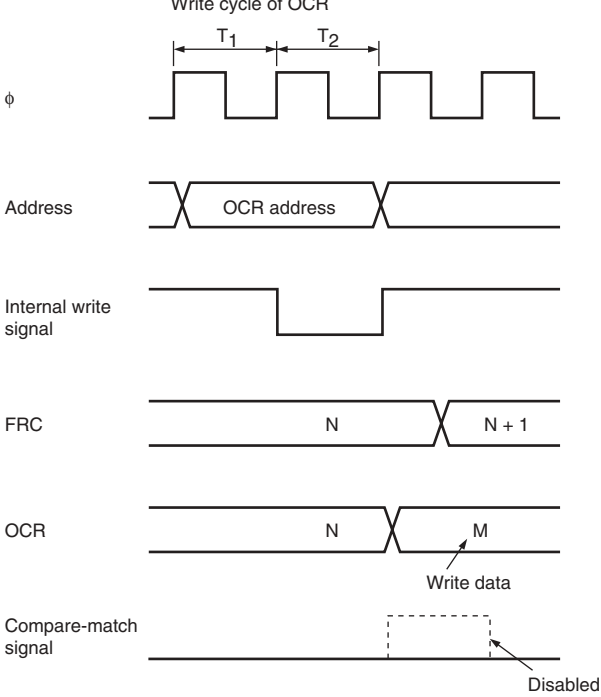


Figure 11.19 Conflict between OCR Write and Compare-Match (When Automatic Addition Function is Not Used)

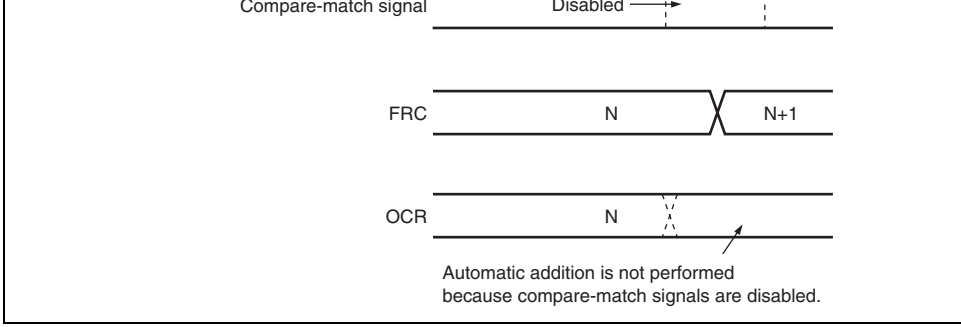
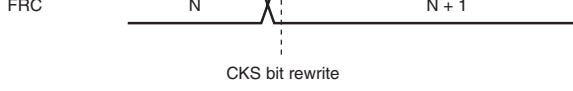
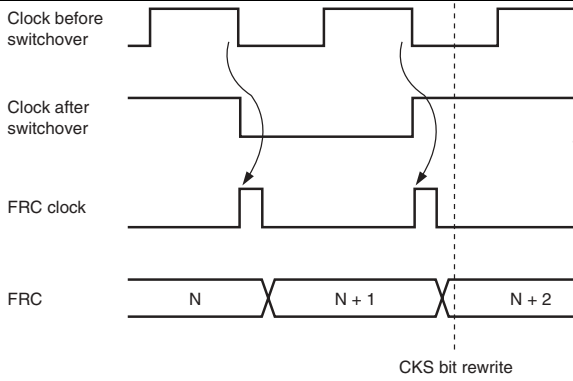
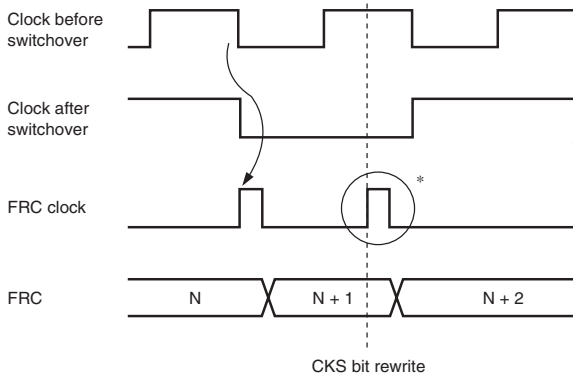


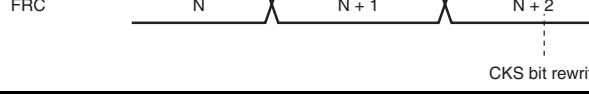
Figure 11.20 Conflict between OCR Write and Compare-Match (When Automatic Addition Function is Used)

11.7.4 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
11.3.

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 11.3, 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	



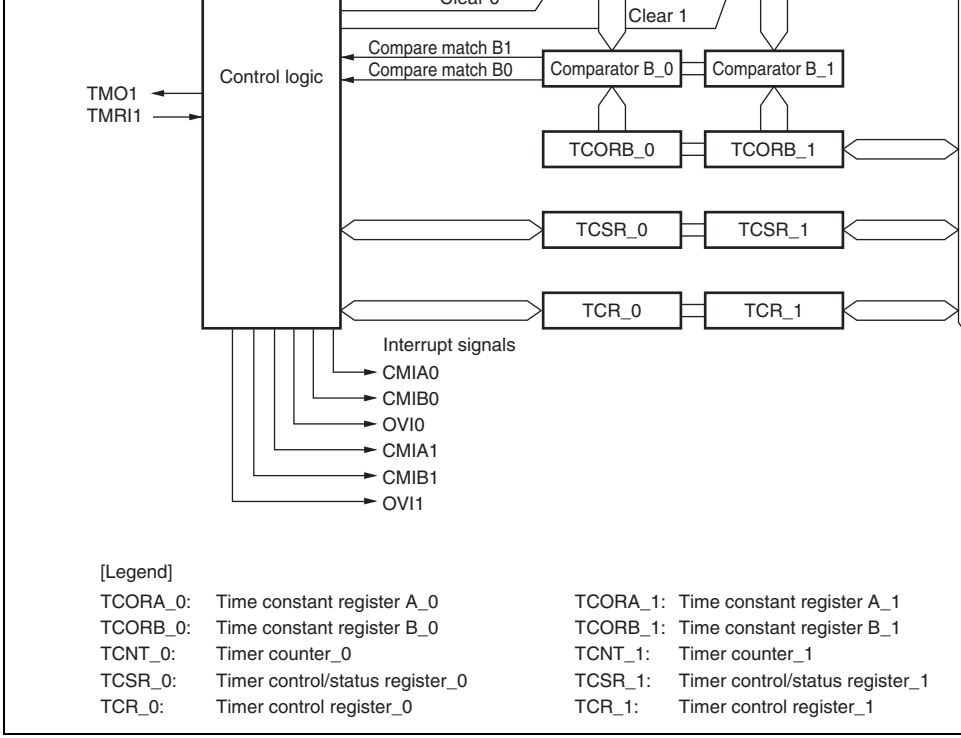
Note: * Generated on the assumption that the switchover is a falling edge; FRC is incr

— TMR_0, TMR_1: The counter input clock can be selected from six internal cloc
external clock
— TMR_Y, TMR_X: The counter input clock can be selected from three internal cl
an external clock

- Selection of three ways to clear the counters
  — The counters can be cleared on compare-match A, compare-match B, or by an ex
  reset signal.
- Timer output controlled by two compare-match signals
  — The timer output signal in each channel is controlled by two independent compar
  signals, enabling the timer to be used for various applications, such as the genera
  pulse output or PWM output with an arbitrary duty cycle.
- 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, and TMR_Y: Three types of interrupts: Compare-match A, compare-match B, and overflow
  — TMR_X: Four types of interrupts: Compare-match A, compare-match B, overflow, and input capture

Figures 12.1 and 12.2 show block diagrams of 8-bit timers.

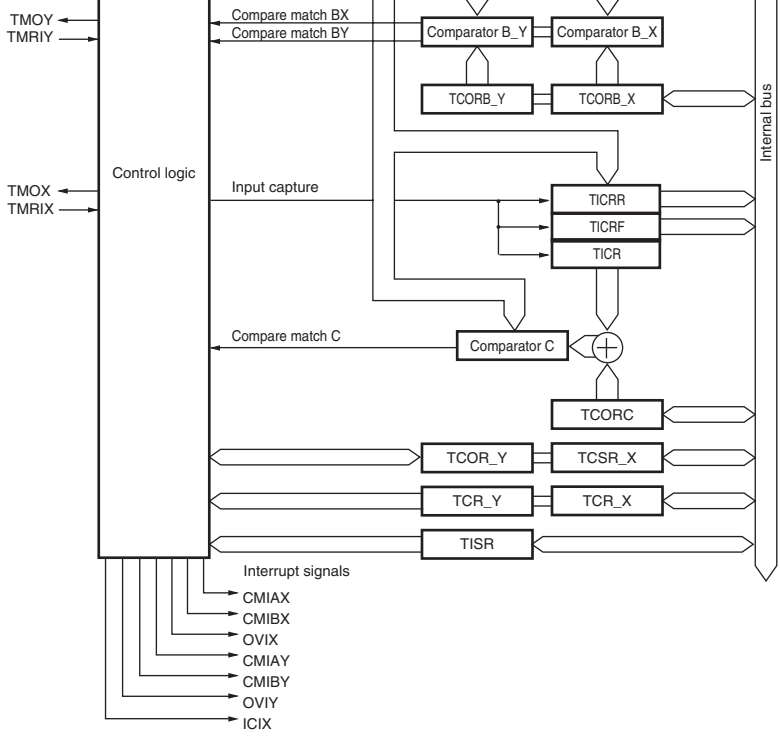
An input capture function is added to TMR_X.



[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 12.1 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
TISR:	Timer input select register	TICR:	Input capture register
		TCORC:	Tme constant registerC
		TICRR:	Input capture register R
		TICRF:	Input capture register F

Figure 12.2 Block Diagram of 8-Bit Timer (TMR_Y and TMR_X)

TMR_Y	Timer output	TMOY	Output	Output controlled by compar
	Timer clock/reset input	TMIY/ExTMIY	Input	External clock input (TMCIY reset input (TMRIY) for the c
TMR_X	Timer output	TMOX	Output	Output controlled by compar
	Timer clock/reset input	TMIX/ExTMIX	Input	External clock input (TMCIX reset input (TMRIX) for the c

- Input capture register R (TICRR)*[1]
- Input capture register F (TICRF)*[1]
- Timer input select register (TISR)*[2]
- Timer connection register I (TCONRI)*[1]
- Timer connection register S (TCONRS)*[1]

Notes: Some of the registers of TMR_X and TMR_Y use the same address. The reg
be switched by the TMRX/Y bit in TCONRS.

1. Only for the TMR_X
2. Only for the TMR_Y

12.3.1 Timer Counter (TCNT)

Each TCNT is an 8-bit readable/writable up-counter. TCNT_0 and TCNT_1 comprise a
bit register, so they can be accessed together by word access. The clock source is selecte
CKS2 to CKS0 bits in TCR. TCNT can be cleared by an external reset input signal, cor
match A signal or compare-match B signal. The method of clearing can be selected by t
and CCLR0 bits in TCR. When TCNT overflows (changes from H'FF to H'00), the OVF
TCSR is set to 1. TCNT is initialized to H'00.

TCNT_Y can be accessed when the KINWUE bit in SYSCR is 0 and the TMRX/Y bit i
TCONRS is 1. TCNT_X can be accessed when the KINWUE bit in SYSCR is 0 and the
bit in TCONRS is 0. See section 3.2.2, System Control Register (SYSCR), and section 1
Timer Connection Register S (TCONRS).

12.3.11, Timer Connection Register S (TCONRS).

12.3.3 Time Constant Register B (TCORB)

TCORB is an 8-bit readable/writable register. TCORB_0 and TCORB_ comprise a singl register, so they can be accessed together by word access. TCORB is continually compar the value in TCNT. When a match is detected, the corresponding compare-match flag B (in TCSR is set to 1. Note however that comparison is disabled during the T2 state of a TC write cycle. The timer output from the TMO pin can be freely controlled by these compar B signals and the settings of output select bits OS3 and OS2 in TCSR. TCORB is initializ H'FF.

TCORB_Y can be accessed when the KINWUE bit in SYSCR is 0 and the TMRX/Y bit TCONRS is 1. TCORB_X can be accessed when the KINWUE bit in SYSCR is 0 and th TMRX/Y bit in TCONRS is 0. See section 3.2.2, System Control Register (SYSCR), and 12.3.11, Timer Connection Register S (TCONRS).

				enabled or disabled when the CMFB flag in T set to 1. 0: CMFB interrupt request (CMIB) is disablec 1: CMFB interrupt request (CMIB) is enablec
6	CMIEA	0	R/W	Compare-Match Interrupt Enable A Selects whether the CMFA interrupt request enabled or disabled when the CMFA flag in T set to 1. 0: CMFA interrupt request (CMIA) is disablec 1: CMFA interrupt request (CMIA) is enablec
5	OVIE	0	R/W	Timer Overflow Interrupt Enable Selects whether the OVF interrupt request (C enabled or disabled when the OVF flag in TC to 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	These bits select the method by which the ti counter is cleared. 00: Clearing is disabled 01: Cleared on compare-match A 10: Cleared on compare-match B 11: Cleared on rising edge of external reset i
2 to 0	CKS2 to CKS0	All 0	R/W	Clock Select 2 to 0 These bits select the clock input to TCNT an condition, together with the ICKS1 and ICKS STCR. For details, see table 12.2.

	0	1	1	—	0	Increments at falling edge of in clock φ/1024
	0	1	1	—	1	Increments at falling edge of in clock φ/256
	1	0	0	—	—	Increments at overflow signal f TCNT_1*
TMR_1	0	0	0	—	—	Disables clock input
	0	0	1	0	—	Increments at falling edge of in clock φ/8
	0	0	1	1	—	Increments at falling edge of in clock φ/2
	0	1	0	0	—	Increments at falling edge of in clock φ/64
	0	1	0	1	—	Increments at falling edge of in clock φ/128
	0	1	1	0	—	Increments at falling edge of in clock φ/1024
	0	1	1	1	—	Increments at falling edge of in clock φ/2048
	1	0	0	—	—	Increments at compare-match TCNT_0*
TMR_Y	0	0	0	—	—	Disables clock input
	0	0	1	—	—	Increments at falling edge of in clock φ/4
	0	1	0	—	—	Increments at falling edge of in clock φ/256

	0	1	1	—	—	Increments at falling edge of i clock $\phi/4$
	1	0	0	—	—	Setting prohibited
Common	1	0	1	—	—	Increments at rising edge of e clock
	1	1	0	—	—	Increments at falling edge of e clock
	1	1	1	—	—	Increments at both rising and edges of external clock.

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 therefo avoided.

6	CMFA	0	R/(W)*	Compare-Match Flag A [Setting condition] When the values of TCNT_0 and TCORA_ [Clearing condition] Read CMFA when CMFA = 1, then write 0
5	OVF	0	R/(W)*	Timer Overflow Flag [Setting condition] When TCNT_0 overflows from H′FF to H′0 [Clearing condition] Read OVF when OVF = 1, then write 0 in C
4	ADTE	0	R/W	A/D Trigger Enable Enables or disables A/D converter start req compare-match A. 0: A/D converter start requests by compare are disabled 1: A/D converter start requests by compare are enabled
3	OS3	0	R/W	Output Select 3 and 2 These bits specify how the TMO0 pin outpu to be changed by compare-match B of TC and TCNT_0. 00: No change 01: 0 is output 10: 1 is output 11: Output is inverted (toggle output)
2	OS2	0	R/W	

Bit	Bit Name	Initial Value	R/W	Description
7	CMFB	0	R/(W)*	Compare-Match Flag B [Setting condition] When the values of TCNT_1 and TCORB_ [Clearing condition] Read CMFB when CMFB = 1, then write 0
6	CMFA	0	R/(W)*	Compare-Match Flag A [Setting condition] When the values of TCNT_1 and TCORA_ [Clearing condition] Read CMFA when CMFA = 1, then write 0
5	OVF	0	R/(W)*	Timer Overflow Flag [Setting condition] When TCNT_1 overflows from H′FF to H′0 [Clearing condition] Read OVF when OVF = 1, then write 0 in
4	—	1	R	Reserved This bit is always read as 1 and cannot be

				and TCNT_1. 00: No change 01: 0 is output 10: 1 is output 11: Output is inverted (toggle output)

Note: * Only 0 can be written, for flag clearing.

- TCSR_Y

  This register can be accessed when the KINWUE bit in SYSCR is 0 and the TMRX/Y
  TCONRS is 1.

Bit	Bit Name	Initial Value	R/W	Description
7	CMFB	0	R/(W)*	Compare-Match Flag B [Setting condition] When the values of TCNT_Y and TCORB_ [Clearing condition] Read CMFB when CMFB = 1, then write 0
6	CMFA	0	R/(W)*	Compare-Match Flag A [Setting condition] When the values of TCNT_Y and TCORA_ [Clearing condition] Read CMFA when CMFA = 1, then write 0

				1: ICF interrupt request (ICIX) is enabled
3 2	OS3 OS2	0 0	R/W R/W	Output Select 3 and 2 These bits specify how the TMOY pin outp to be changed by compare-match B of TC and TCNT_Y. 00: No change 01: 0 is output 10: 1 is output 11: Output is inverted (toggle output)
1 0	OS1 OS0	0 0	R/W R/W	Output Select 1 and 0 These bits specify how the TMOY pin outp to be changed by compare-match A of TC and TCNT_Y. 00: No change 01: 0 is output 10: 1 is output 11: Output is inverted (toggle output)

Table: * Only 0 can be written, for flag clearing.

				When the values of TCNT_X and TCORA_
				[Clearing condition]
				Read CMFA when CMFA = 1, then write 0
5	OVF	0	R/(W)*	Timer Overflow Flag
				[Setting condition]
				When TCNT_X overflows from H'FF to H'0
				[Clearing condition]
				Read OVF when OVF = 1, then write 0 in C
4	ICF	0	R/(W)*	Input Capture Flag
				[Setting condition]
				When a rising edge and falling edge is dete the external reset signal in that order.
				[Clearing condition]
				Read ICF when ICF = 1, then write 0 in ICF
3	OS3	0	R/W	Output Select 3 and 2
2	OS2	0	R/W	These bits specify how the TMOX pin outpu to be changed by compare-match B of TCO and TCNT_X.
				00: No change
				01: 0 is output
				10: 1 is output
				11: Output is inverted (toggle output)

12.3.6 Input Capture Register (TICR)

TICR is an 8-bit register. The contents of TCNT are transferred to TICR at the rising ed external reset input. TICR cannot be directly accessed by the CPU.

12.3.7 Time Constant Register C (TCORC)

TCORC is an 8-bit readable/writable register. The sum of contents of TCORC and TICF compared with TCNT. When a match is detected, a compare-match C signal is generate However, comparison at the T2 state in the write cycle to TCORC and at the input captu TICR is disabled. TCORC is initialized to H'FF.

12.3.8 Input Capture Registers R and F (TICRR and TICRF)

TICRR and TICRF are 8-bit read-only registers. While the ICST bit in TCONRI is set to contents of TCNT are transferred at the rising edge and falling edge of the external reset that order. The ICST bit is cleared to 0 when one capture operation ends. TICRR and TI initialized to H'00.

TICRR and TICRF can be accessed when the KINWUE bit in SYSCR is 0 and the TMR TCONRS is 0. See section 3.2.2, System Control Register (SYSCR).

0: TMIY or ExTMIY (TMCIY/TMRIY) is not s

1: TMIY or ExTMIY (TMCIY/TMRIY) is sele

12.3.10 Timer Connection Register I (TCONRI)

TCONRI controls TMR_X input capture.

Bit	Bit Name	Initial Value	R/W	Description
7 to 5	—	All 0	R/W	Reserved The initial values should not be modifiec
4	ICST	0	R/W	Input Capture Start Bit TMR_X has input capture registers (TIC TICRR, and TICRF). TICRR and TICRF measure the width of a pulse by means single capture operation under the contr ICST bit. When a rising edge followed b edge is detected on TMRIX after the ICS set to 1, the contents of TCNT at those captured into TICRR and TICRF, respec and the ICST bit is cleared to 0. [Clearing condition] When a rising edge followed by a falling detected on TMRIX [Setting condition] When 1 is written in ICST after reading
3 to 0	—	All 0	R/W	Reserved The initial values should not be modifiec

Table 12.3 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	TICRR	TICRF	TCNT_X	TCORC	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	TISR	

can be output without the intervention of software.

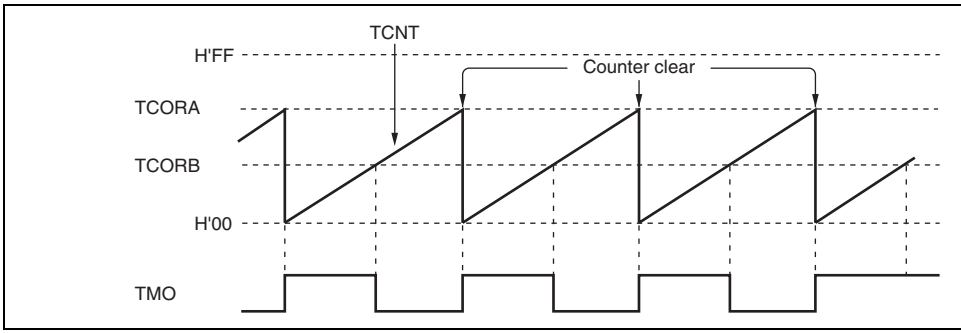


Figure 12.3 Pulse Output Example

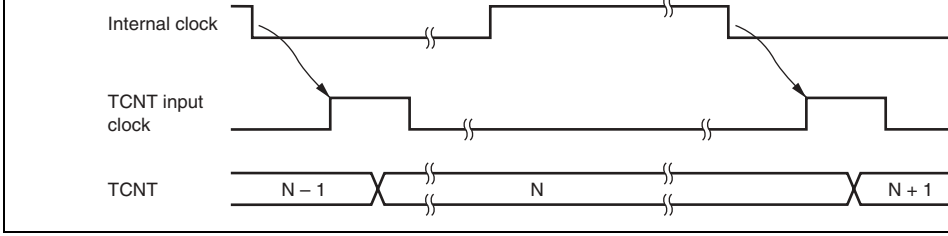


Figure 12.4 Count Timing for Internal Clock Input

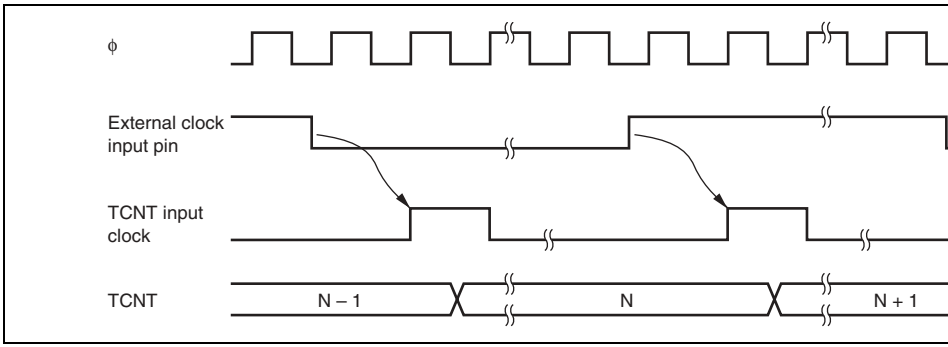


Figure 12.5 Count Timing for External Clock Input

12.5.2 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
TCNT and TCOR values match. The compare-match signal is generated at the last state
the match is true, just when the timer counter is updated. Therefore, when TCNT and T
match, the compare-match signal is not generated until the next TCNT input clock. Figu
shows the timing of CMF flag setting.

12.5.3 Timing of Timer Output at Compare-Match

When a compare-match signal occurs, the timer output changes as specified by the OS3 t
bits in TCSR. Figure 12.7 shows the timing of timer output when the output is set to togg
compare-match A signal.

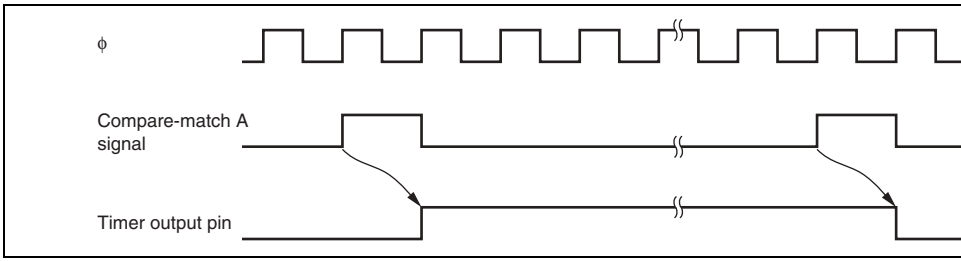


Figure 12.7 Timing of Toggled Timer Output by Compare-Match A Signa

12.5.4 Timing of Counter Clear at Compare-Match

TCNT is cleared when compare-match A or compare-match B occurs, depending on the s
the CCLR1 and CCLR0 bits in TCR. Figure 12.8 shows the timing of clearing the counte
compare-match.

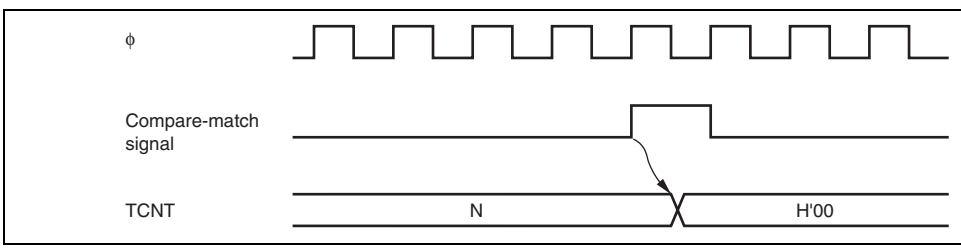


Figure 12.8 Timing of Counter Clear by Compare-Match

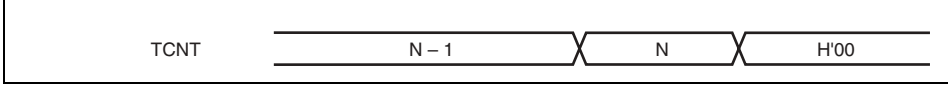


Figure 12.9 Timing of Counter Clear by External Reset Input

12.5.6 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
12.10 shows the timing of OVF flag setting.

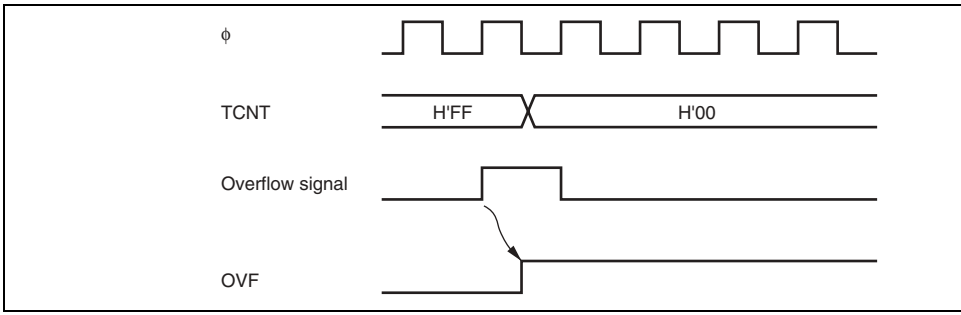


Figure 12.10 Timing of OVF Flag Setting

— 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.
- Pin output
  — Control of output from the TMO0 pin by bits OS3 to OS0 in TCSR_0 is in accord the 16-bit compare-match conditions.
  — Control of output from the TMO1 pin by bits OS3 to OS0 in TCSR_1 is in accord the lower 8-bit compare-match conditions.

12.6.2 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, output from the TMO pin, and counter clearing are in accordance with the settings for each channel.

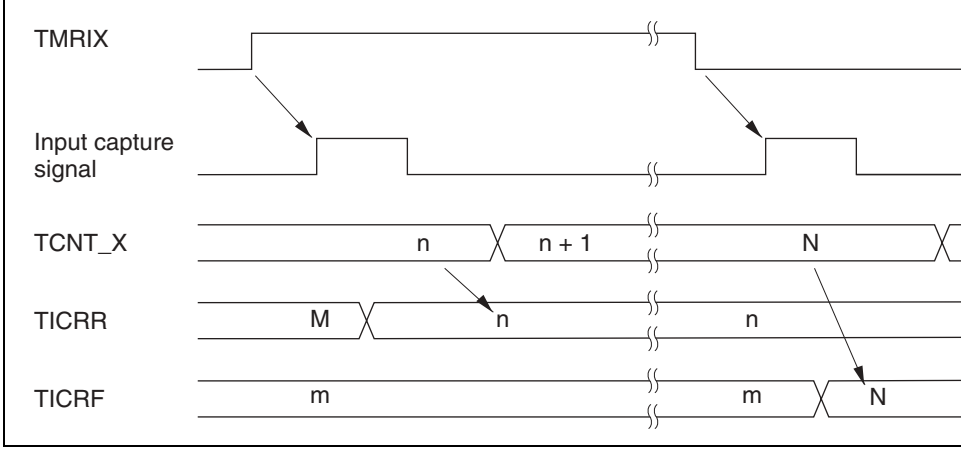


Figure 12.11 Timing of Input Capture Operation

If the input capture signal is input while TICRR and TICRF are being read, the input cap
signal is delayed by one system clock (ϕ) cycle. Figure 12.12 shows the timing of this op

(Input capture signal is input during TICRR and TICRF read)

Selection of Input Capture Signal Input: TMRIX (input capture input signal of TMR_X switched according to the setting of the ICST bits in TCONR1. Input capture signal selec shown in table 12.4.

Table 12.4 Input Capture Signal Selection

TCONRI Bit 4 ICST	Description
0	Input capture function not used
1	TMIX pin input selection

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	
	ICIX	Input capture	ICF	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	Low

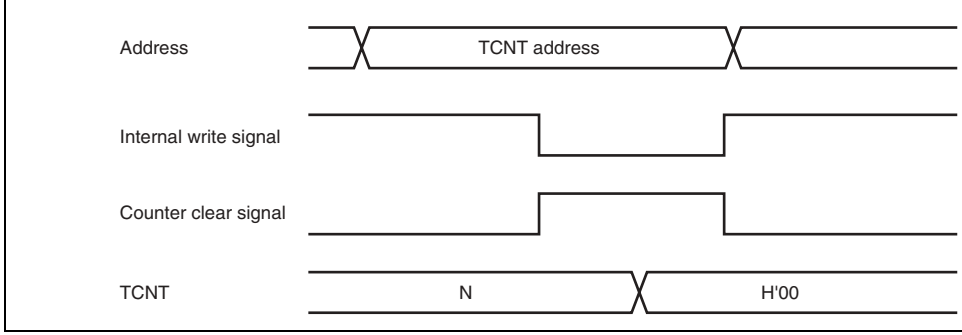


Figure 12.13 Conflict between TCNT Write and Counter Clear

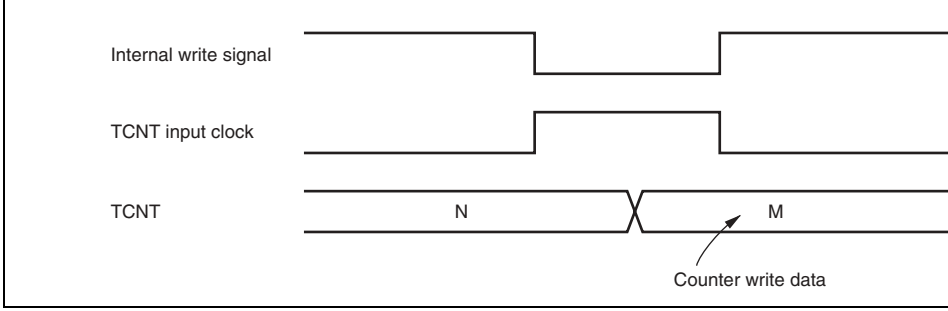


Figure 12.14 Conflict between TCNT Write and Increment

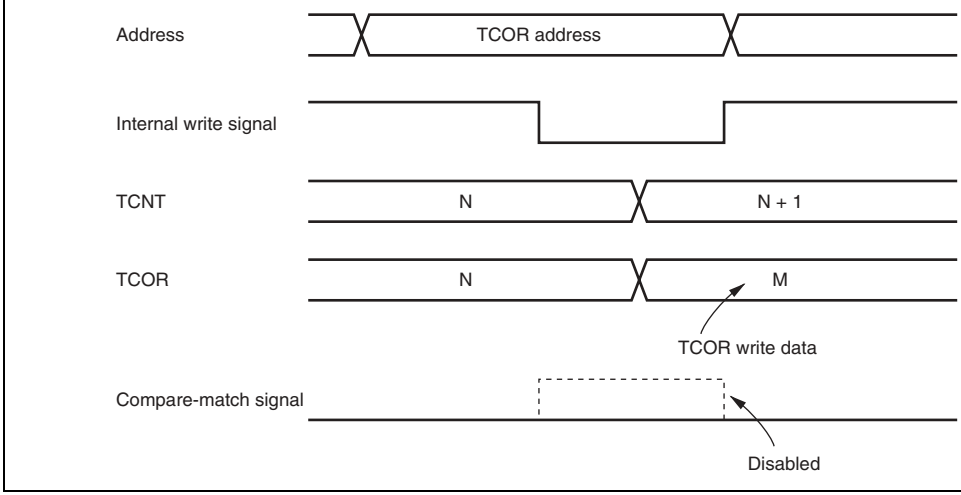


Figure 12.15 Conflict between TCOR Write and Compare-Match

No change	Low

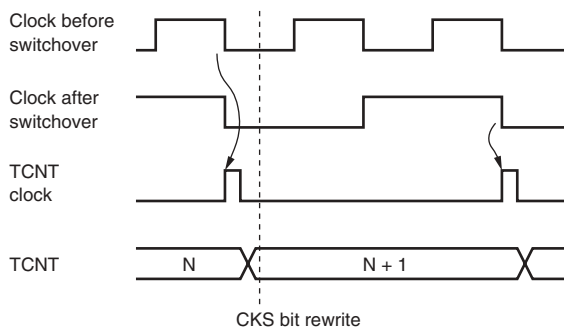
12.9.5 Switching of Internal Clocks and TCNT Operation

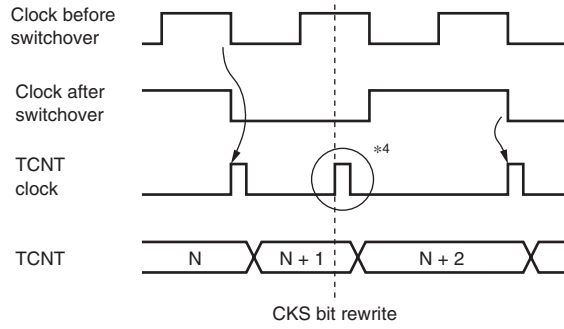
TCNT may increment erroneously when the internal clock is switched over. Table 12.7 relationship between the timing at which the internal clock is switched (by writing to the and CKS0 bits) and the TCNT operation.

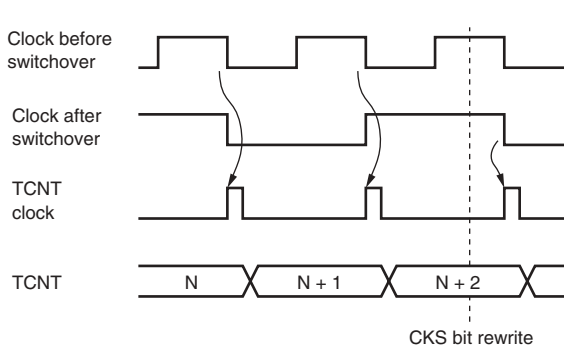
When the TCNT clock is generated from an internal clock, the falling edge of the intern pulse is detected. If clock switching causes a change from high to low level, as shown in table 12.7, a TCNT clock pulse is generated on the assumption that the switchover is a f edge, and TCNT is incremented.

Erroneous incrementation can also happen when switching between internal and externa

Table 12.7 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	

3	Clock switching from high to low level*3	

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.

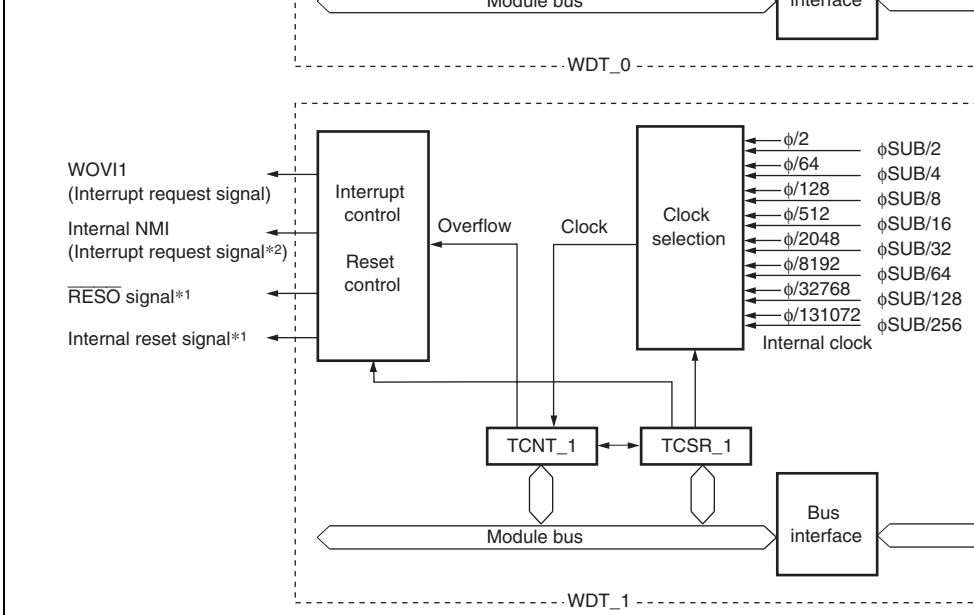
- Selectable from eight (WDT_0) or 16 (WDT_1) counter input clocks.
- Switchable between watchdog timer mode and interval timer mode

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 $\overline{\text{RESO}}$ 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 sign 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 W The interrupt controller does not distinguish the NMI interrupt request from WDT_0 fro that from WDT_1.

Figure 13.1 Block Diagram of WDT

TCNT is an 8-bit readable/writable up-counter.

TCNT is initialized to H'00 when the TME bit in timer control/status register (TCSR) is c
to 0.

13.3.2 Timer Control/Status Register (TCSR)

TCSR selects the clock source to be input to TCNT, and the timer mode.

- TCSR_0

Bit	Bit Name	Initial Value	R/W	Description
7	OVF	0	R/(W)*	Overflow Flag Indicates that TCNT has overflowed (changes fr to H'00). [Setting conditions] • When TCNT overflows (changes from H'FF t • When internal reset request generation is se watchdog timer mode, OVF is cleared autom by the internal reset. [Clearing conditions] • When TCSR is read when OVF = 1, then 0 is 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

2 to 0	CKS2 to CKS0	All 0	R/W	Clock Select 2 to 0 Select the clock source to be input to TCNT. Tl overflow frequency for ϕ = 33 MHz is enclosed parentheses. 000: ϕ/2 (frequency: 15.5 µs) 001: ϕ/64 (frequency: 496.5 µs) 010: ϕ/128 (frequency: 993.0 µs) 011: ϕ/512 (frequency: 4.0 ms) 100: ϕ/2048 (frequency: 15.9 ms) 101: ϕ/8192 (frequency: 63.6 ms) 110: ϕ/32768 (frequency: 254.2 ms) 111: ϕ/131072 (frequency: 1.02 s)

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

				written 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 watchdo or 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 a initialized to H'00.
4	PSS	0	R/W	Prescaler Select Selects the clock source to be input to TCNT. 0: Counts the divided cycle of ϕ–based prescal 1: Counts the divided cycle of ϕSUB–based pre (PSS)
3	RST/$\overline{NMI}$	0	R/W	Reset or NMI Selects to request an internal reset or an NMI i when TCNT has overflowed. 0: An NMI interrupt is requested 1: An internal reset is requested

110: ϕ/32768 (frequency: 254.2 ms)

111: ϕ/131072 (frequency: 1.02 s)

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.

RESO pin for 132 states, as shown in figure 13.2. If the RST/NMI bit is cleared to 0, whe
TCNT overflows, an NMI interrupt request is generated. Here, the output from the RESO
remains high.

An internal reset request from the watchdog timer and a reset input from the 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 RES pin occurs at the same time as a reset caused
WDT overflow, the 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.

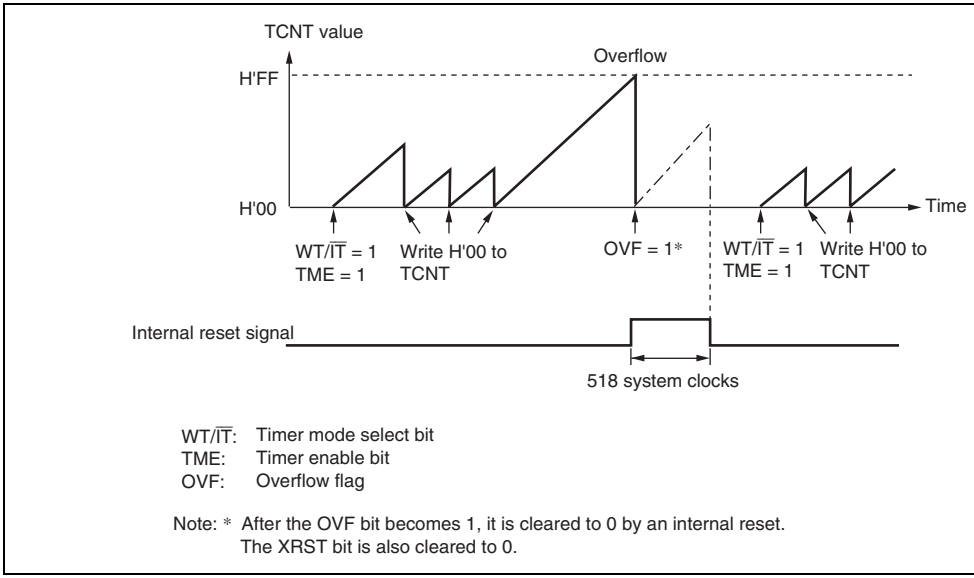
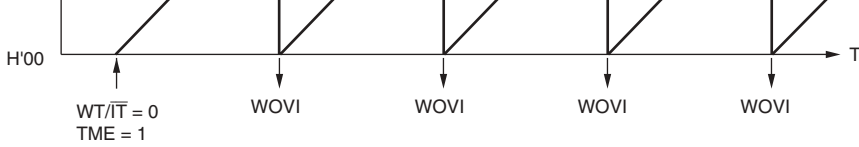


Figure 13.2 Watchdog Timer Mode (RST/NMI = 1) Operation



WOVI : Interval timer interrupt request occurrence

Figure 13.3 Interval Timer Mode Operation

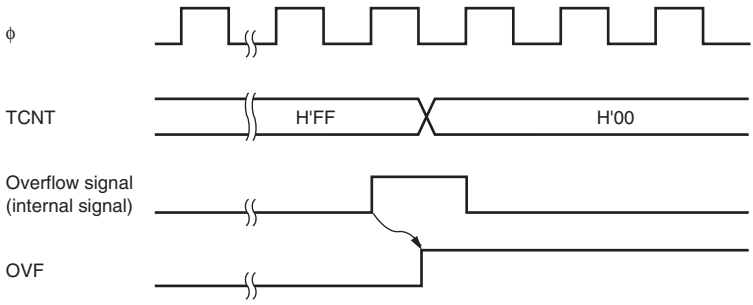


Figure 13.4 OVF Flag Set Timing

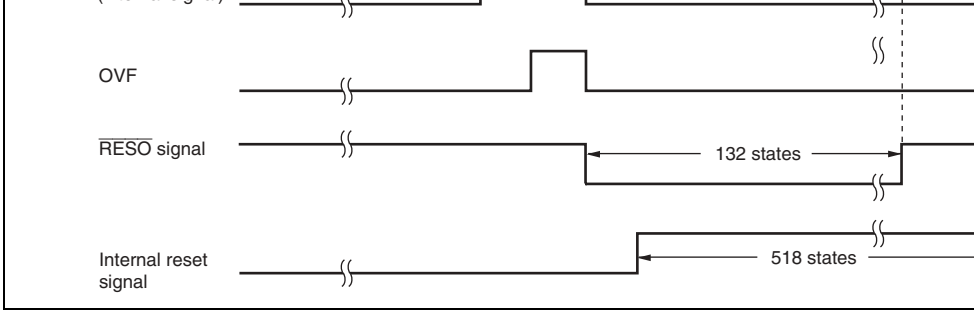


Figure 13.5 Output Timing of $\overline{\text{RESO}}$ signal

TCNT and TCSR both have the same write address. Therefore, satisfy the relative conditi
shown in figure 13.6 to write to TCNT or TCSR. To write to TCNT, the higher bytes mus
the value H'5A and the lower bytes must contain the write data. To write to TCSR, the hi
bytes must contain the value H'A5 and the lower bytes must contain the write data.

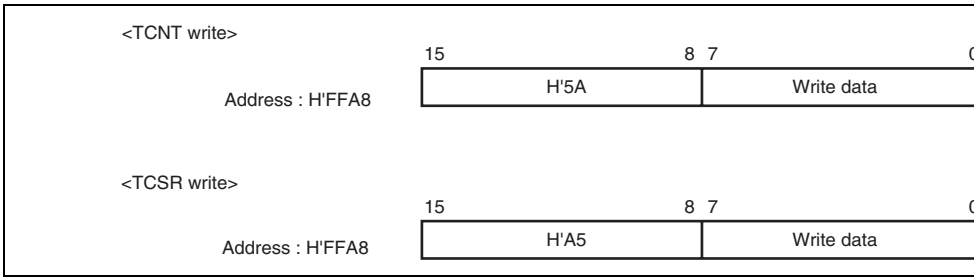


Figure 13.6 Writing to TCNT and TCSR (WDT_0)

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 fo
and H'FFA9 for TCNT.

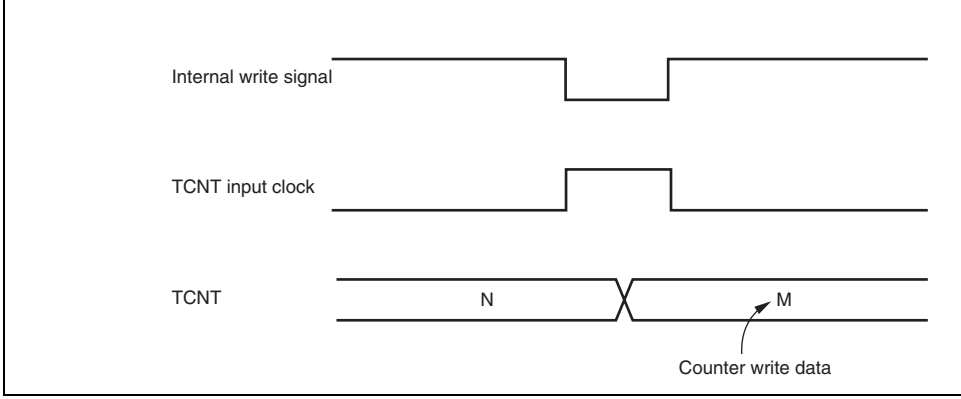


Figure 13.7 Conflict between TCNT Write and Increment

13.6.3 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.

13.6.4 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 v PSS bit.

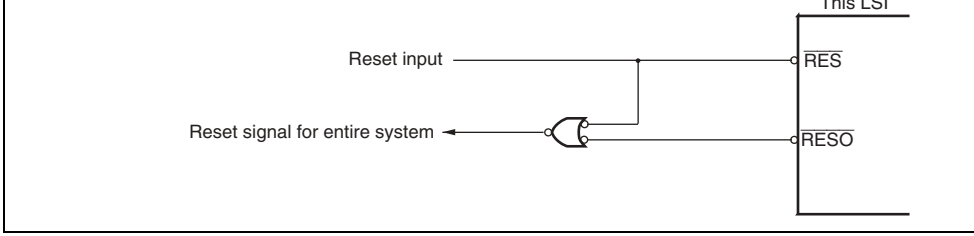


Figure 13.8 Sample Circuit for Resetting the System by the $\overline{\text{RESO}}$ Signal

SCI_1 can handle communication using the waveform based on the Infrared Data Assoc
(IrDA) standard version 1.0. SCI_0 and SCI_2 provide high-speed communication at an
transfer rate of a specific system clock frequency. Reliable fast data transfers are secured
internal cyclic redundancy check (CRC) operation circuit. Since the CRC operation circ
connected to the SCI, data is transferred to the circuit using the MOV instruction to be o
there.

14.1 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

Clock Synchronous Mode:

- Data length: 8 bits
- Receive error detection: Overrun errors
- SCI channel selectable (SCI_0 and SCI_2): When SSE0I = 1, TxD0 = high-impedanc and SCK0 = fixed to high input; when SSE2I = 1, TxD2 = high-impedance state and S fixed to high input

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 14.1 shows a block diagram of SCI_1, and figure 14.2 shows a block diagram of S SCI_2.

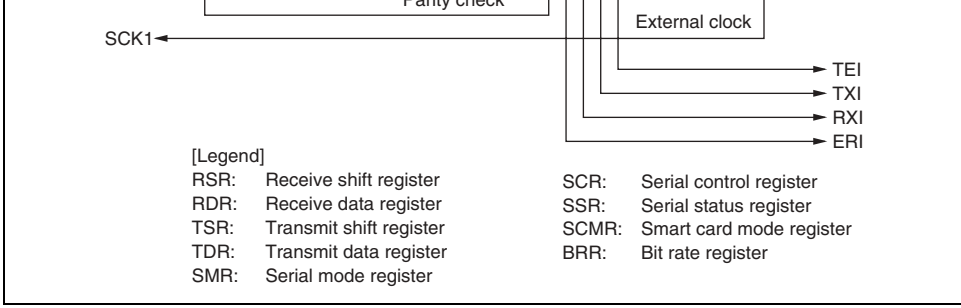
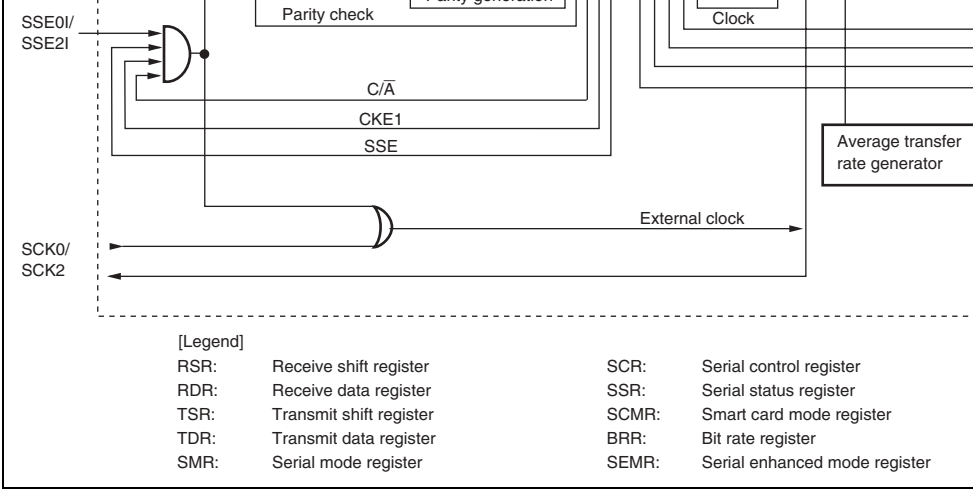


Figure 14.1 Block Diagram of SCI_1



[Legend]

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

Figure 14.2 Block Diagram of SCI_0 and SCI_2

	RxD1/IrRxD	Input	Channel 1 receive data input (normal/IrDA)
	TxD1/IrTxD	Output	Channel 1 transmit data output (normal/IrD
2	SCK2	Input/Output	Channel 2 clock input/output
	RxD2	Input	Channel 2 receive data input
	TxD2	Output	Channel 2 transmit data output
	SSE2I	Input	Channel 2 stop input

Note: * Pin names SCK, RxD, and TxD are used in the text for all channels, omitting channel designation.

- 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)
- Serial interface control register (SCICR)*[1]
- Serial enhanced mode register (SEMR)*[2]

Notes: 1. SCICR is not available in SCI_0 or SCI_2.
2. SEMR is not available in SCI_1.

14.3.1 Receive Shift Register (RSR)

RSR is a shift register used to receive serial data that converts it into parallel data. When
frame of data has been received, it is transferred to RDR automatically. RSR cannot be di
accessed by the CPU.

14.3.2 Receive Data Register (RDR)

RDR is an 8-bit register that stores receive data. When the SCI has received one frame of
data, it transfers the received serial data from RSR to RDR where it is stored. After this, I
receive the next data. Since RSR and RDR function as a double buffer in this way, contin
receive operations be performed. After confirming that the RDRF bit in SSR is set to 1, re
for only once. RDR cannot be written to by the CPU.

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.

14.3.5 Serial Mode Register (SMR)

SMR is used to set the SCI's serial transfer format and select the baud rate generator clo
Some bits in SMR have different functions in normal mode and smart card interface mo

				bits is used.
5	PE	0	R/W	Parity Enable (enabled only in asynchronous When this bit is set to 1, the parity bit is adde transmit data before transmission, and the pa checked in reception. For a multiprocessor fo parity bit addition and checking are not perfo regardless of the PE bit setting.
4	O/$\overline{E}$	0	R/W	Parity Mode (enabled only when the PE bit is asynchronous mode) 0: Selects even parity. 1: Selects odd parity.
3	STOP	0	R/W	Stop Bit Length (enabled only in asynchronou Selects the stop bit length in transmission. 0: 1 stop bit 1: 2 stop bits In reception, only the first stop bit is checked second stop bit is 0, it is treated as the start b next transmit frame.
2	MP	0	R/W	Multiprocessor Mode (enabled only in asynch mode) When this bit is set to 1, the multiprocessor communication function is enabled. The PE b O/$\overline{E}$ bit settings are invalid in multiprocessor

(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 opera GSM mode, the TEND set timing is put forw 11.0 etu* from the start and the clock outpu function is appended. For details, see sectio Clock Output Control.
6	BLK	0	R/W	Setting this bit to 1 allows block transfer mo operation. For details, see section 14.7.3, B Transfer Mode.
5	PE	0	R/W	Parity Enable (valid only in asynchronous m When this bit is set to 1, the parity bit is add transmit data before transmission, and the p checked in reception. Set this bit to 1 in sma interface mode.
4	O/$\overline{E}$	0	R/W	Parity Mode (valid only when the PE bit is 1 asynchronous mode) 0: Selects even parity 1: Selects odd parity For details on the usage of this bit in smart interface mode, see section 14.7.2, Data Fo (Except in Block Transfer Mode).

1	CKS1	0	R/W	Clock Select 1 and 0
0	CKS0	0	R/W	These bits select the clock source for the bau 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 and the baud rate, see section 14.3.9, Bit Ra Register (BRR). n is the decimal display of th of n in BRR (see section 14.3.9, Bit Rate Reg (BRR)).

Note: * etu: Element Time Unit (time taken to transfer one bit)

6	RIE	0	R/W	Receive Interrupt Enable When this bit is set to 1, RXI and ERI interru requests are enabled.
5	TE	0	R/W	Transmit Enable When this bit is set to 1, transmission is ena
4	RE	0	R/W	Receive Enable When this bit is set to 1, reception is enable
3	MPIE	0	R/W	Multiprocessor Interrupt Enable (enabled or the MP bit in SMR is 1 in asynchronous mo When this bit is set to 1, receive data in whi multiprocessor bit is 0 is skipped, and settin RDRF, FER, and ORER status flags in SSR disabled. On receiving data in which the multiprocessor bit is 1, this bit is automatica and normal reception is resumed. For detail section 14.5, Multiprocessor Communicatio Function.
2	TEIE	0	R/W	Transmit End Interrupt Enable When this bit is set to 1, a TEI interrupt requ enabled.

(Inputs a clock with a frequency 16 times
rate from the SCK pin.)

Clock synchronous mode:

0*: Internal clock (SCK pin functions as clock

1*: External clock (SCK pin functions as cloc

[Legend]

*: Don't care

				When this bit is set to 1, reception is enable
3	MPIE	0	R/W	Multiprocessor Interrupt Enable (enabled or MP bit in SMR is 1 in asynchronous mode) Write 0 to this bit in smart card interface mo
2	TEIE	0	R/W	Transmit End Interrupt Enable Write 0 to this bit in smart card interface mo
1	CKE1	0	R/W	Clock Enable 1 and 0
0	CKE0	0	R/W	These bits control the clock output from the GSM mode, clock output can be dynamicall For details, see section 14.7.8, Clock Outpu When GM in SMR = 0 00: Output disabled (SCK pin functions as I/ 01: Clock output 1*: Reserved When GM in SMR = 1 00: Output fixed to low 01: Clock output 10: Output fixed to high 11: Clock output

[Legend]

*: Don't care.

				• When data is transferred from TDR to TDR is ready for data write [Clearing conditions] • When 0 is written to TDRE after readi 1 • When a TXI interrupt request is issue DTC to write data to TDR
6	RDRF	0	R/(W)*	Receive Data Register Full Indicates that receive data is stored in RD [Setting condition] • When serial reception ends normally a receive data is transferred from RSR [Clearing conditions] • When 0 is written to RDRF after read = 1 • When an RXI interrupt request is issu DTC to read data from RDR The RDRF flag is not affected and retains previous value when the RE bit in SCR is 0.

				When 0 is written to FER after reading FE In 2-stop-bit mode, only the first stop bit is
3	PER	0	R/(W)*	Parity Error [Setting condition] When a parity error is detected during rec [Clearing condition] When 0 is written to PER after reading PE
2	TEND	1	R	Transmit End [Setting conditions] • When the TE bit in SCR is 0 • When TDRE = 1 at transmission of the a 1-byte serial transmit character [Clearing conditions] • When 0 is written to TDRE after readin 1 • When a TXI interrupt request is issued DTC to write data to TDR
1	MPB	0	R	Multiprocessor Bit MPB stores the multiprocessor bit in the r frame. When the RE bit in SCR is cleared previous state is retained.
0	MPBT	0	R/W	Multiprocessor Bit Transfer MPBT stores the multiprocessor bit to be the transmit frame.

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

				• When a TXI interrupt request is issued DTC to write data to TDR
6	RDRF	0	R/(W)*[1]	Receive Data Register Full Indicates that receive data is stored in RD [Setting condition] • When serial reception ends normally a receive data is transferred from RSR to [Clearing conditions] • When 0 is written to RDRF after readi = 1 • When an RXI interrupt request is issue allowing DTC to read data from RDR The RDRF flag is not affected and retains previous value when the RE bit in SCR is to 0.
5	ORER	0	R/(W)*[1]	Overrun Error [Setting condition] When the next serial reception is complete RDRF = 1 [Clearing condition] When 0 is written to ORER after reading O
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 ER

				• When ERS = 0 and TDRE = 1 after a time passed after the start of 1-byte d transfer. The set timing depends on th setting as follows. When GM = 0 and BLK = 0, 2.5 etu*2 transmission start When GM = 0 and BLK = 1, 1.5 etu*2 transmission start When GM = 1 and BLK = 0, 1.0 etu*2 transmission start When GM = 1 and BLK = 1, 1.0 etu*2 transmission start [Clearing conditions] • When 0 is written to TDRE after readi TDRE = 1 • When a TXI interrupt request is issue DTC 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 m

Notes: 1. Only 0 can be written, to clear the flag.
2. etu: Element Time Unit (time taken to transfer one bit)

				1: TDR contents are transmitted with MSB-
				Stores receive data as MSB first in RDR.
				The SDIR bit is valid only when the 8-bit da is used for transmission/reception; when th data format is used, data is always transmitted/received with LSB-first.
2	SINV	0	R/W	Smart Card Data Invert
				Specifies inversion of the data logic level. T bit does not affect the logic level of the pari When the parity bit is inverted, invert the O/ SMR.
				0: TDR contents are transmitted as they ar Receive data is stored as it is in RDR.
				1: TDR contents are inverted before being transmitted. Receive data is stored in inv form in RDR.
1	—	1	R	Reserved
				This bit is always read as 1 and cannot be
0	SMIF	0	R/W	Smart Card Interface Mode Select
				When this bit is set to 1, smart card interfac is selected.
				0: Normal asynchronous or clock synchron mode
				1: Smart card interface mode

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**	
0	0	3
0	1	6
1	0	3
1	1	2

Table 14.3 shows sample N settings in BRR in normal asynchronous mode. Table 14.4 s maximum bit rate settable for each frequency. Table 14.6 and 14.8 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 14.7.4, Receive Data Sampling Timing and Reception Margin. T and 14.7 show the maximum bit rates with external clock input.

4800	0	32	–1.36	0	38	0.16	0	39	0.00	0	47	0.00	0	5
9600	0	15	1.73	0	19	–2.34	0	19	0.00	0	23	0.00	0	2
19200	0	7	1.73	0	9	–2.34	0	9	0.00	0	11	0.00	0	1
31250	0	4	0.00	0	5	0.00	0	5	2.40	—	—	—	0	7
38400	0	3	1.73	0	4	–2.34	0	4	0.00	0	5	0.00	—	–

	Operating Frequency ϕ (MHz)										
	9.8304			10			12			12.	
Bit Rate (bit/s)	n	N	Error (%)	n	N	Error (%)	n	N	Error (%)	n	N
110	2	174	–0.26	2	177	–0.25	2	212	0.03	2	217
150	2	127	0.00	2	129	0.16	2	155	0.16	2	159
300	1	255	0.00	2	64	0.16	2	77	0.16	2	79
600	1	127	0.00	1	129	0.16	1	155	0.16	1	159
1200	0	255	0.00	1	64	0.16	1	77	0.16	1	79
2400	0	127	0.00	0	129	0.16	0	155	0.16	0	159
4800	0	63	0.00	0	64	0.16	0	77	0.16	0	79
9600	0	31	0.00	0	32	–1.36	0	38	0.16	0	39
19200	0	15	0.00	0	15	1.73	0	19	–2.34	0	19
31250	0	9	–1.70	0	9	0.00	0	11	0.00	0	11
38400	0	7	0.00	0	7	1.73	0	9	–2.34	0	9

[Legend]

—: Can be set, but there will be a degree of error.

Note: * Make the settings so that the error does not exceed 1%.

4800	0	90	0.16	0	95	0.00	0	103	0.16	0	11
9600	0	45	–0.93	0	47	0.00	0	51	0.16	0	55
19200	0	22	–0.93	0	23	0.00	0	25	0.16	0	27
31250	0	13	0.00	0	14	–1.70	0	15	0.00	0	16
38400	—	—	—	0	11	0.00	0	12	0.16	0	16

	Operating Frequency ϕ (MHz)											
	18			19.6608			20			25		
Bit Rate (bit/s)	n	N	Error (%)	n	N	Error (%)	n	N	Error (%)	n	N	Error (%)	n	N
110	3	79	–0.12	3	86	0.31	3	88	–0.25	3	110	–0.02	3	1
150	2	233	0.16	2	255	0.00	3	64	0.16	3	80	–0.47	3	1
300	2	116	0.16	2	127	0.00	2	129	0.16	2	162	0.15	2	2
600	1	233	0.16	1	255	0.00	2	64	0.16	2	80	–0.47	2	1
1200	1	116	0.16	1	127	0.00	1	129	0.16	1	162	0.15	1	2
2400	0	233	0.16	0	255	0.00	1	64	0.16	1	80	–0.47	1	1
4800	0	116	0.16	0	127	0.00	0	129	0.16	0	162	0.15	0	2
9600	0	58	–0.69	0	63	0.00	0	64	0.16	0	80	–0.47	0	1
19200	0	28	1.02	0	31	0.00	0	32	–1.36	0	40	–0.76	0	5
31250	0	17	0.00	0	19	–1.70	0	19	0.00	0	24	0.00	0	3
38400	0	14	–2.34	0	15	0.00	0	15	1.73	0	19	1.73	0	2

[Legend]

—: Can be set, but there will be a degree of error.

Note: * Make the settings so that the error does not exceed 1%.

12	375000	0	0	25	781250	0	
12.288	384000	0	0	33	1031250	0	

Table 14.5 Maximum Bit Rate with External Clock Input (Asynchronous Mode)

ϕ (MHz)	External Input Clock (MHz)	Maximum Bit Rate (bit/s)	ϕ (MHz)	External Input Clock (MHz)	Maxim Rate (
5	1.2500	78125	14	3.5000	21875
6	15.000	93750	14.7456	3.6864	23040
6.144	1.5360	96000	16	4.0000	25000
7.3728	1.8432	115200	17.2032	4.3008	26880
8	2.0000	125000	18	4.5000	28125
9.8304	2.4576	153600	19.6608	4.9152	30720
10	2.5000	156250	20	5.0000	31250
12	3.0000	187500	25	6.2500	39062
12.288	3.0720	192000	33	8.2500	51562

10k	0	199	0	249	1	99	1	124	1	149	1
25k	0	79	0	99	0	159	0	199	0	239	0
50k	0	39	0	49	0	79	0	99	0	119	0
100k	0	19	0	24	0	39	0	49	0	59	0
250k	0	7	0	9	0	15	0	19	0	23	0
500k	0	3	0	4	0	7	0	9	0	11	0
1M	0	1			0	3	0	4	0	5	0
2.5M			0	0*			0	1			
5M							0	0*			

[Legend]

: Setting prohibited.

— : Can be set, but there will be a degree of error.

*: Continuous transfer or reception is not possible.

Table 14.7 Maximum Bit Rate with External Clock Input (Clock Synchronous M

ϕ (MHz)	External Input Clock (MHz)	Maximum Bit Rate (bit/s)	ϕ (MHz)	External Input Clock (MHz)	Max Rate
6	1.0000	1000000.0	16	2.6667	2666
8	1.3333	1333333.3	18	3.0000	3000
10	1.6667	1666666.7	20	3.3333	3333
12	2.0000	2000000.0	25	4.1667	4166
14	2.3333	2333333.3	33	5.5000	5500

Table 14.9 Maximum Bit Rate for Each Frequency (Smart Card Interface Mode, S

ϕ (MHz)	Maximum Bit Rate (bit/s)	n	N	ϕ (MHz)	Maximum Bit Rate (bit/s)	n	
7.1424	9600	0	0	18.00	24194	0	
10.00	13441	0	0	20.00	26882	0	
13.00	17473	0	0	21.4272	28800	0	
14.2848	19200	0	0	25.00	33602	0	
16.00	21505	0	0	33.00	44355	0	

5 4	IrCKS1 IrCKS0	0 0	R/W R/W	These bits specify the high-level width of the pulse during IrTxD output pulse encoding whe IrDA function is enabled. 000: B x 3/16 (three sixteenths of the bit rate) 001: ϕ/2 010: ϕ/4 011: ϕ/8 100: ϕ/16 101: ϕ/32 110: ϕ/64 111: ϕ/128
3, 2	—	All 0	R/W	Reserved The initial value should no be changed.
1, 0	—	All 0	R	Reserved These bits are always read as 0 and cannot b modified.

				1: Enables the external pins to select the SCI f • SCI_0 SSE0I pin input = 0 (selected state): SCI_0 op normally SSE0I pin input = 1 (non-selected state): SCI_ operation (TxD0 = high-impedance state, SCK0 = fixed t • SCI_2 SSE2I pin input = 0 (selected state): SCI_2 op normally SSE2I pin input = 1 (non-selected state): SCI_ operation (TxD2 = high-impedance state, SCK2 = fixed t
6, 5	—	All 0	R	Reserved These bits are always read as 0 and cannot be modified.
3	ABCS	0	R/W	Asynchronous Mode Basic Clock Select Specifies the basic clock for a 1-bit cycle in asynchronous mode. This bit is valid only in asynchronous mode (C/ SMR is 0). 0: The basic clock has a frequency 16 times th clock frequency (normal operation) 1: The basic clock has a frequency 8 times the clock frequency (double-speed operation)

0010: Average transfer rate operation at 460.
when the system clock frequency is 10.
(operated using the basic clock with a f
8 times the transfer clock frequency)

0011: Average transfer rate operation at 720 k
the system clock frequency is 32 MHz (
using the basic clock with a frequency
the transfer clock frequency)

0100: Reserved

0101: Average transfer rate operation at 115.
when the system clock frequency is 16
(operated using the basic clock with a f
16 times the transfer clock frequency)

0110: Average transfer rate operation at 460.
when the system clock frequency is 16
(operated using the basic clock with a f
16 times the transfer clock frequency)

0111: Average transfer rate operation at 720
the system clock frequency is 16 MHz (
using the basic clock with a frequency 8
transfer clock frequency)

1000: Average transfer rate operation at 115.
when the system clock frequency is 16
(operated using the basic clock with a f
16 times the transfer clock frequency)

1001: Average transfer rate operation at 230.
when the system clock frequency is 16
(operated using the basic clock with a f
16 times the transfer clock frequency)

1101: Average transfer rate operation at 230.
when the system clock frequency is 24 M
(operated using the basic clock with a fre
16 times the transfer clock frequency)

1110: Average transfer rate operation at 460.7
when the system clock frequency is 24 M
(operated using the basic clock with a fre
16 times the transfer clock frequency)

1111: Average transfer rate operation at 720 k
the system clock frequency is 24 MHz (o
using the basic clock with a frequency 8
transfer clock frequency)

Table 14.10 Asynchronous Mode Clock Source Select

ACS 4	ACS 2	ACS 1	ACS 0	Average Transfer Rate	System Clock (ϕ)	Operating
0	0	0	0	None	External clock input, normal operation	Transfer ra Transfer ra
0	0	0	1	115.152 kbps	10.667 MHz	Transfer ra
0	0	1	0	460.606 kbps	10.667 MHz	Transfer ra
0	0	1	1	Reserved	Reserved	Reserved
0	1	0	0	Reserved	Reserved	Reserved
0	1	0	1	115.196 kbps	16 MHz	Transfer ra
0	1	1	0	460.784 kbps	16 MHz	Transfer ra
0	1	1	1	720 kbps	16 MHz	Transfer ra
1	0	0	0	115.196 kbps	16 MHz	Transfer ra
1	0	0	1	230.392 kbps	16 MHz	Transfer ra
1	0	1	0	115.196 kbps	20 MHz	Transfer ra

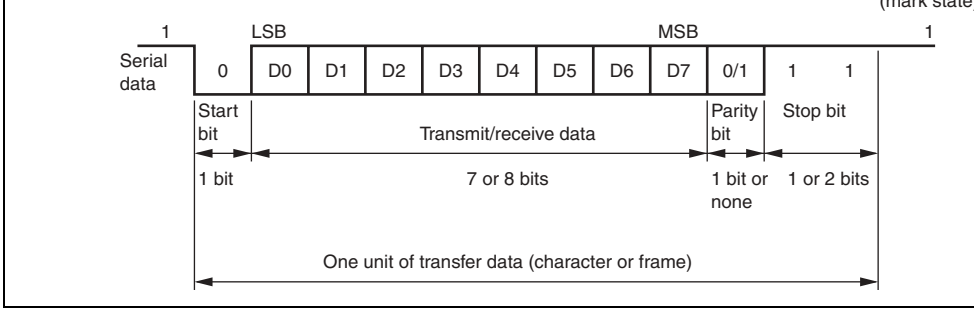


Figure 14.3 Data Format in Asynchronous Communication (Example with 8-Bit Data, Parity, Two Stop Bits)

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 | S
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 | S
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

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 allowe system design.

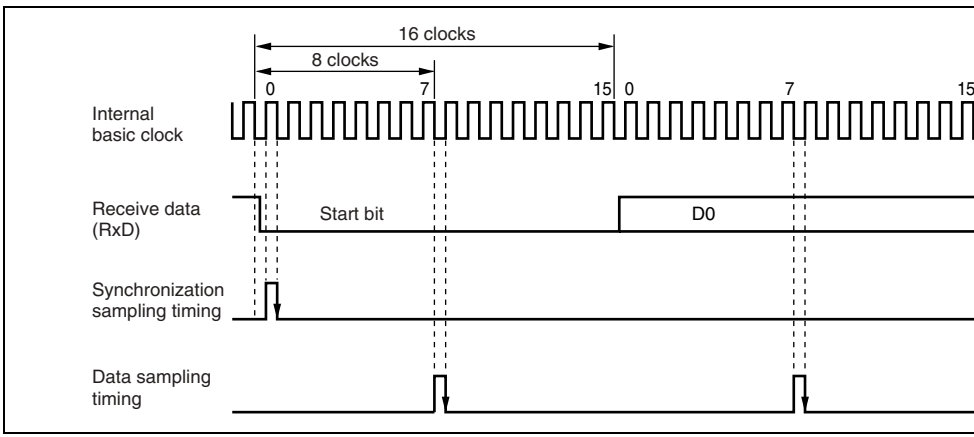


Figure 14.4 Receive Data Sampling Timing in Asynchronous Mode

Figure 14.5 Relation between Output Clock and Transmit Data Phase (Asynchronous Mode)

14.4.4 Serial Enhanced Mode Clock

SCI_0 and SCI_2 can be operated not only based on the clocks described in section 14.4 but based on the following clocks, which are specified by the serial enhanced mode regi SEMR_0 and SEMR_2.

Double-Speed Operation: Operations that are usually achieved using the clock with fre times the normal bit rate can be achieved using the clock with frequency 8 times the bit mode. That is, double transfer rate can be achieved using a single basic clock.

Double-speed operation can be specified by the ABCS bit in SEMR and is available for sources of an internal clock generated by the on-chip baud rate generator and an externa input at the SCK pin. However, double-speed operation cannot be specified when the av transfer rate operation is selected.

Average Transfer Rate Operation: The SCI can be operated based on the clock with a transfer rate generated from the system clock instead of the external clock input at the S this case, the SCK pin is fixed to input.

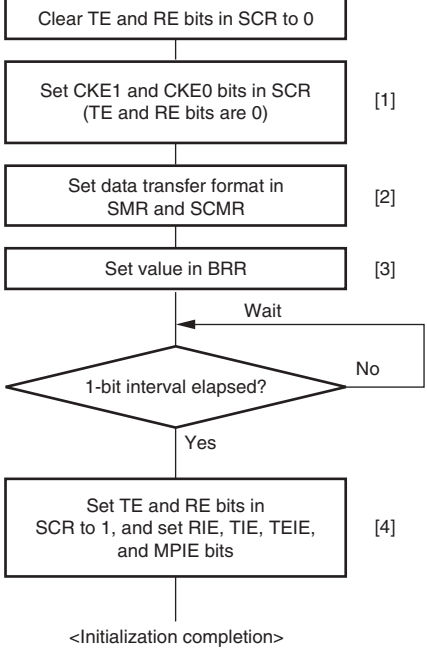
Average transfer rate operation can be specified by the ACS4 and ACS2 to ACS0 bits in Double-speed operation may be selected by clearing the ACS4 and ACS2 to ACS0 bits

Figures 14.6 and 14.7 show some examples of internal basic clock operations when aver transfer rate operation is selected.

Figure 14.6 Basic Clock Examples When Average Transfer Rate is Selected

Figure 14.7 Basic Clock Examples When Average Transfer Rate is Selected

RENESAS



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 14.8 Sample SCI Initialization Flowchart

3. Data is sent from the TXD pin in the following order: start bit, transmit data, parity b 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, a 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 state" is entered in which 1 is output. If the TEIE bit in SCR is set to 1 at this time, a interrupt request is generated.

Figure 14.10 shows a sample flowchart for transmission in asynchronous mode.

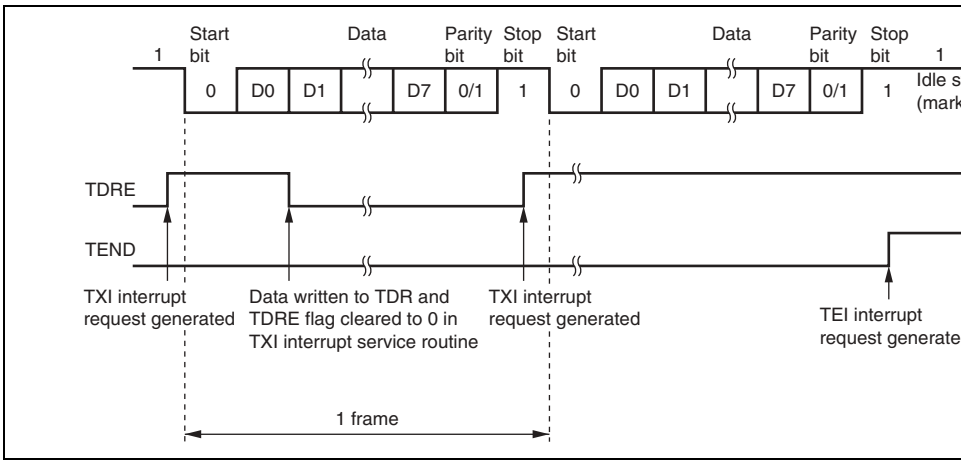
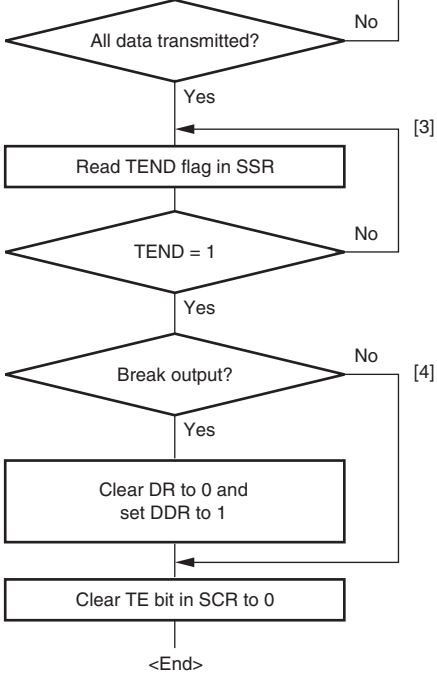


Figure 14.9 Example of Operation in Transmission in Asynchronous Mode (Exa 8-Bit Data, Parity, One Stop Bit)



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 bit in SCR to 0.

Figure 14.10 Sample Serial Transmission Flowchart

RDR. If the RIE bit in SCR is set to 1 at this time, an ERI interrupt request is genera

4. If a framing error (when the stop bit is 0) is detected, the FER bit in SSR is set to 1 a data is transferred to RDR. If the RIE bit in SCR is set to 1 at this time, an ERI interr 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 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.

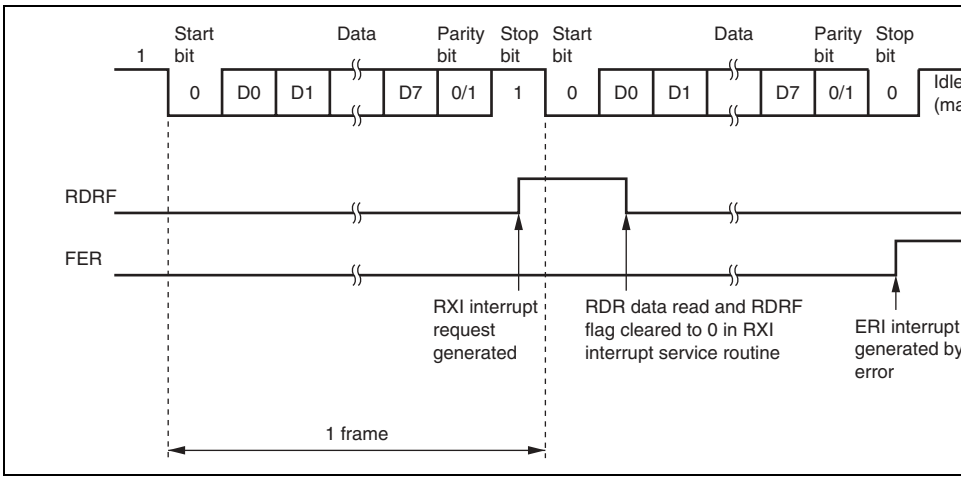
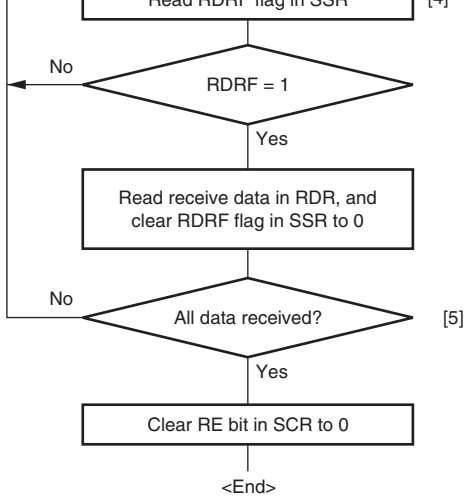


Figure 14.11 Example of SCI Operation in Reception (Example with 8-Bit Data One Stop Bit)

1	1	1	0	Lost	Overrun error + framin
1	1	0	1	Lost	Overrun error + parity
0	0	1	1	Transferred to RDR	Framing error + parity
1	1	1	1	Lost	Overrun error + framin parity error

Note: * The RDRF flag retains the state it had before data reception.



[4] SCI status check and receive data
Read SSR and check that RDRF
then read the receive data in RDF
clear the RDRF flag to 0. Transiti
the RDRF flag from 0 to 1 can als
identified by an RXI interrupt.

[5] Serial reception continuation proc
To continue serial reception, befo
stop bit for the current frame is
received, read the RDRF flag, rea
RDR, and clear the RDRF flag to
However, the RDRF flag is cleare
automatically when the DTC is ini
by an RXI interrupt and reads dat
RDR.

Legend
∨ : Logical add (OR)

Figure 14.12 Sample Serial Reception Flowchart (1)

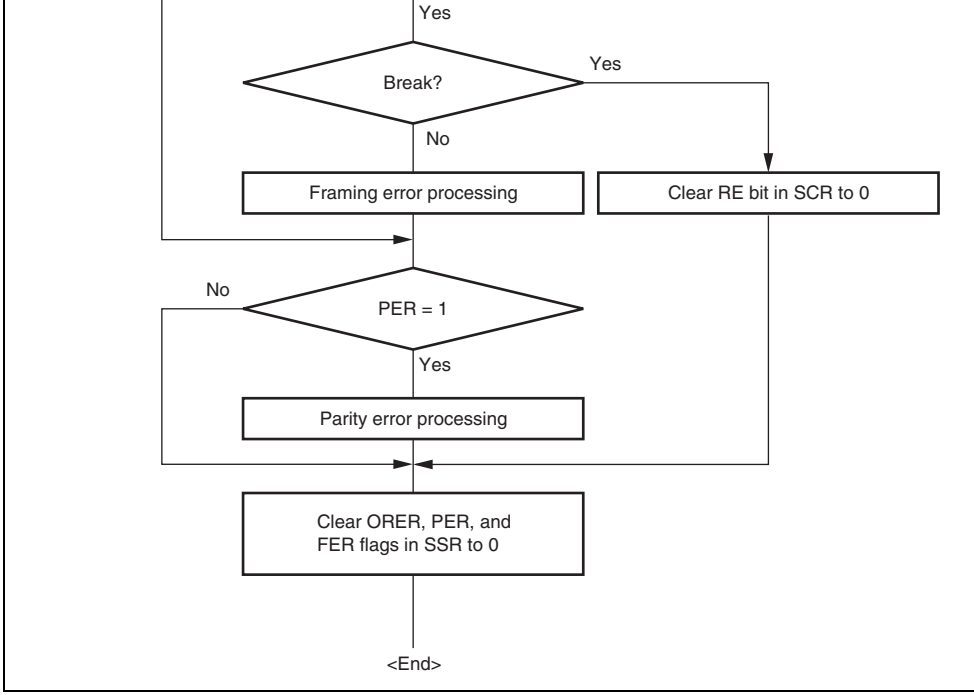


Figure 14.12 Sample Serial Reception Flowchart (2)

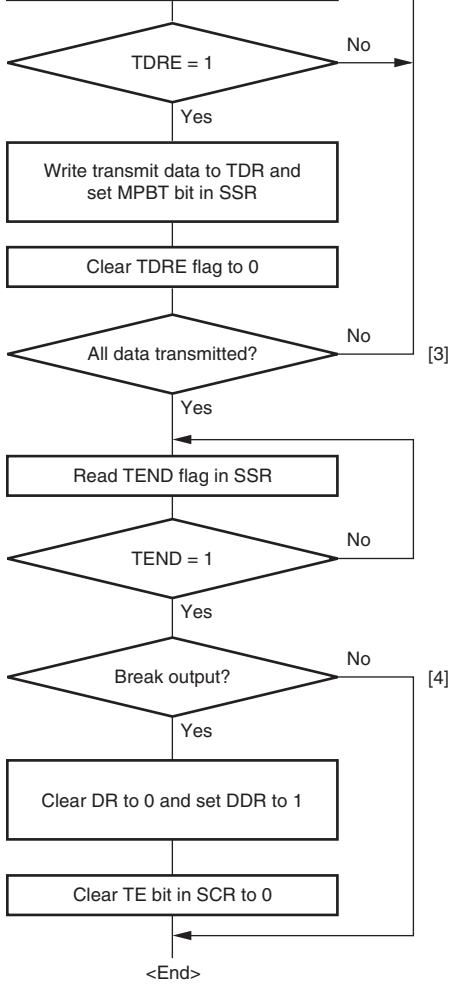
transmitting station first sends the ID code of the receiving station with which it wants to
serial communication as data with a 1 multiprocessor bit added. It then sends transmit da
with a 0 multiprocessor bit added. When data with a 1 multiprocessor bit is received, the
station compares that data with its own ID. The station whose ID matches then receives
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
transfer of receive data from RSR to RDR, error flag detection, and setting the RDRF, F
ORER status flags in SSR to 1 are prohibited until data with a 1 multiprocessor bit is rec
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 SC
1 at this time, an RXI interrupt is generated.

When the multiprocessor format is selected, the parity bit setting is invalid. All other bit
are the same as those in normal asynchronous mode. The clock used for multiprocessor
communication is the same as that in normal asynchronous mode.

Legend
MPB: Multiprocessor bit

Figure 14.13 Example of Communication Using Multiprocessor Format (Transmi
Data H'AA to Receiving Station A)



[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 14.14 Sample Multiprocessor Serial Transmission Flowchart

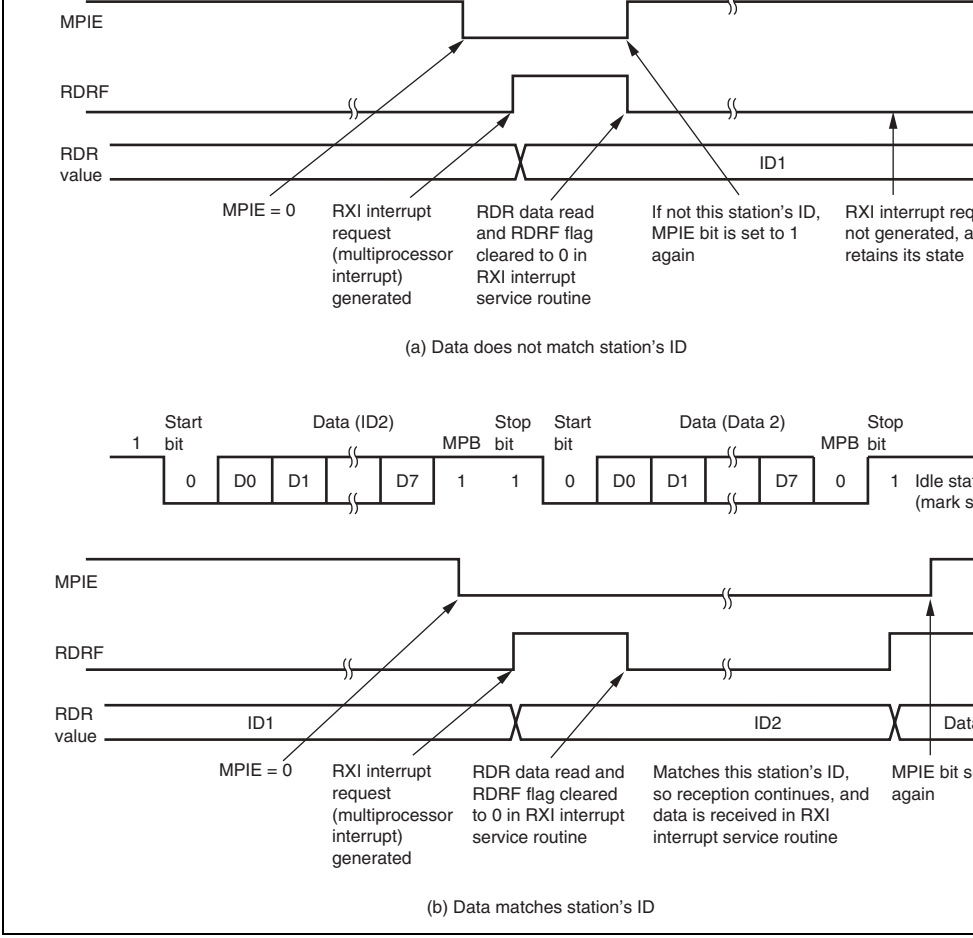
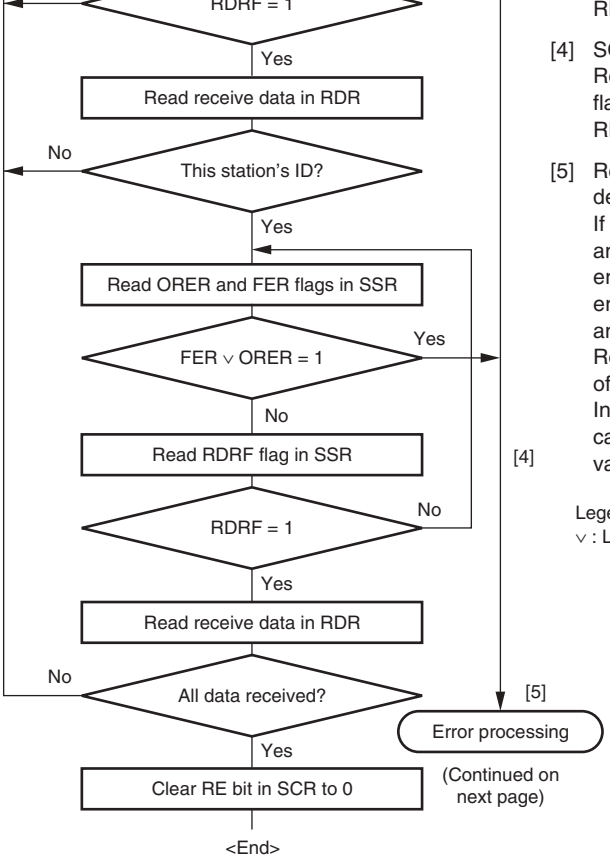


Figure 14.15 Example of SCI Operation in Reception (Example with 8-Bit Da Multiprocessor Bit, One Stop Bit)



RDRF flag to 0.

[4] SCI status check and data recep
Read SSR and check that the RD
flag is set to 1, then read the data
RDR.

[5] Receive error processing and bre
detection:
If a receive error occurs, read the
and FER flags in SSR to identify
error. After performing the appro
error processing, ensure that the
and FER flags are all cleared to 0
Reception cannot be resumed if
of these flags is set to 1.
In the case of a framing error, a b
can be detected by reading the R
value.

Legend
∨ : Logical add (OR)

Figure 14.16 Sample Multiprocessor Serial Reception Flowchart (1)

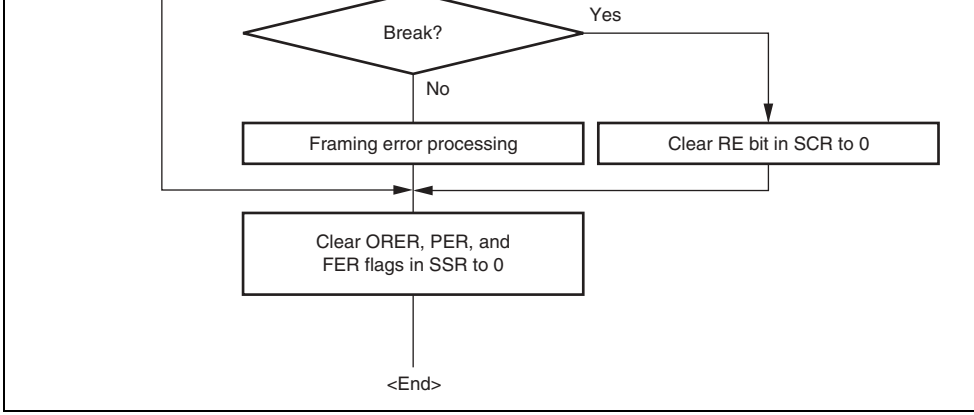


Figure 14.16 Sample Multiprocessor Serial Reception Flowchart (2)

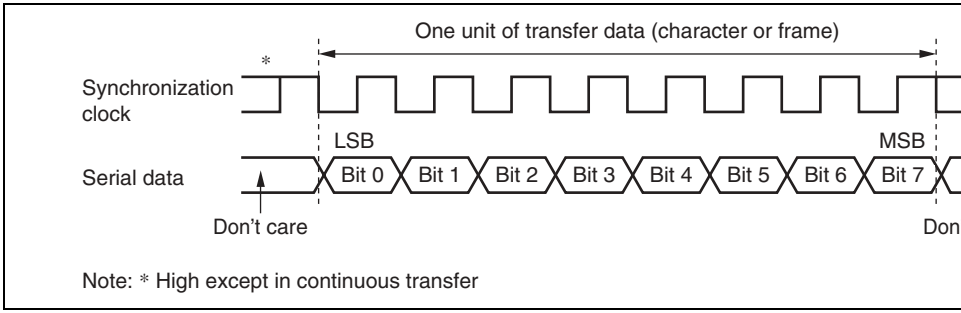
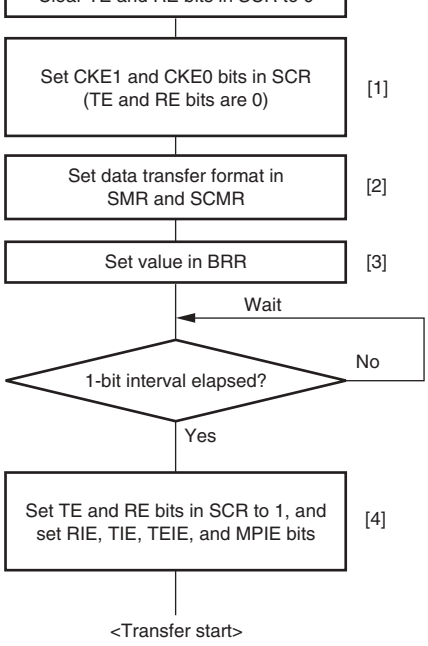


Figure 14.17 Data Format in Synchronous Communication (LSB-First)

14.6.1 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 th and CKE0 bits in SCR. When the SCI is operated on an internal clock, the synchronizati 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.



[2] Set the data transfer format in SMR ar SCMR.

[3] Write a value corresponding to the bit rate to BRR. This step is not necessa if an external clock is used.

[4] Wait at least one bit interval, then set the TE bit or RE bit in SCR to 1. Also set the RIE, TIE TEIE, and MPIE bits.
Setting the TE and RE bits enables the TxD and RxD pins to be used.

Note: In simultaneous transmit and receive operations, the TE and RE bits should both be cleared to 0 or set to 1 simultaneously.

Figure 14.18 Sample SCI Initialization Flowchart

mode has been specified and synchronized with the input clock when use of an exter
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 tran
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 maint
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 14.20 shows a sample flowchart for serial data transmission. Even if the TDRE fl
cleared to 0, transmission will not start while a receive error flag (ORER, FER, or PER)
Make sure to clear the receive error flags to 0 before starting transmission. Note that cle
RE bit to 0 does not clear the receive error flags.

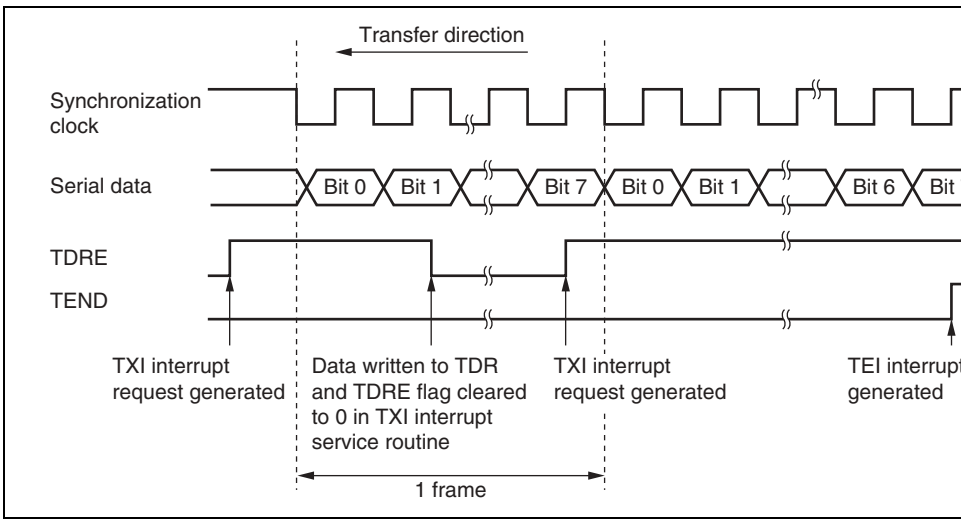
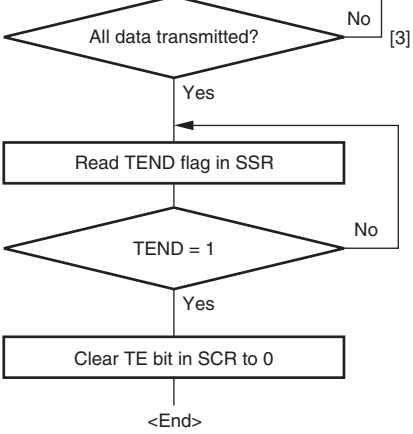


Figure 14.19 Sample SCI Transmission Operation in Clock Synchronous M



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 14.20 Sample Serial Transmission Flowchart

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.

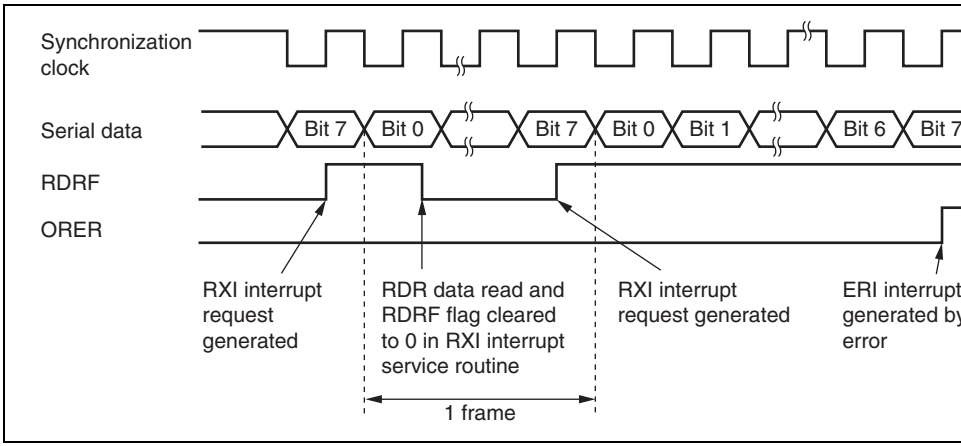
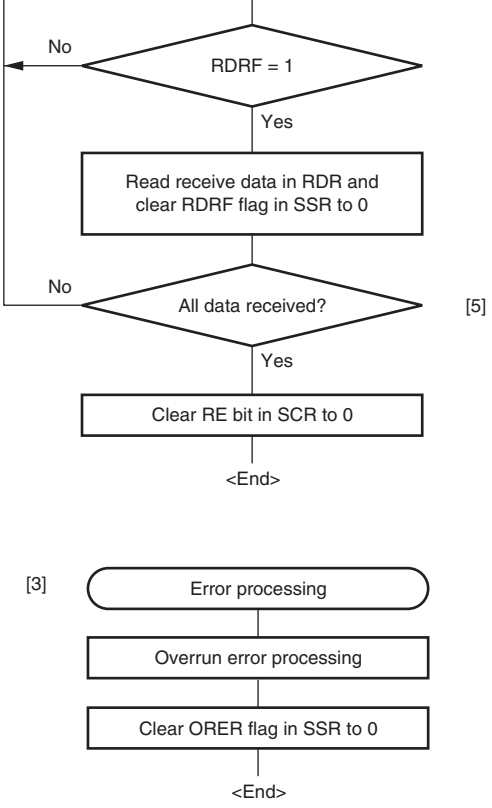


Figure 14.21 Example of SCI Receive Operation in Clock Synchronous Mo

Reception cannot be resumed while a receive error flag is set to 1. Accordingly, clear th
FER, PER, and RDRF bits to 0 before resuming reception. Figure 14.22 shows a sample
for serial data reception.



[5]

[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.

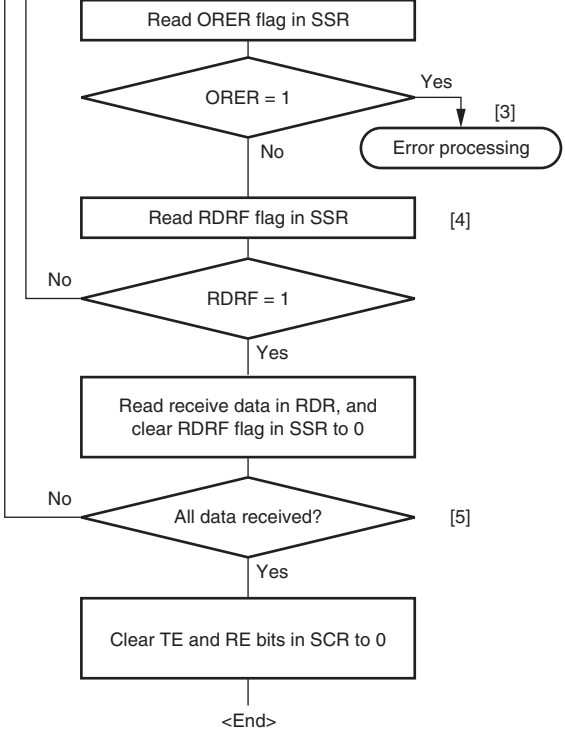
Figure 14.22 Sample Serial Reception Flowchart

14.6.6 SCI Selection in Serial Enhanced Mode

SCI_0 and SCI_2 provide the following capability according to the serial enhanced mod
(SEMR_0 and SEMR_2) settings.

If the SCI is used in clock synchronous mode with clock input, the SCI channel can be
enabled/disabled using the input at the external pins. The external pins include PA0/SSE
(SCI_0) and PA1/SSE2I (SCI_2); therefore, this capability is not available in modes wh
PA0 and PA1 pins are automatically set for address output.

When the SCI operation is disabled (not selected) by input at the external pins, TxD outp
fixed to the high-impedance state and SCK input is internally fixed to high. One-to-mult
communication is possible if the master device, which outputs SCK, controls these exter
for chip selection. SCI selection capability is selected using the SSE bits in SEMR.



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.

ORER flag in SSR, and after performing the appropriate er processing, clear the ORER f Transmission/reception cannc resumed if the ORER flag is s

[4] SCI status check and receive read:
Read SSR and check that the flag is set to 1, then read the data in RDR and clear the RD to 0. Transition of the RDRF 0 to 1 can also be identified b interrupt.

[5] Serial transmission/reception continuation procedure:
To continue serial transmissio reception, before the MSB (bi the current frame is received, reading the RDRF flag, readir and clearing the RDRF flag to before the MSB (bit 7) of the c frame is transmitted, read 1 fr TDRE flag to confirm that writ possible. Then write data to T clear the TDRE flag to 0.
However, the TDRE flag is ch and cleared automatically wh DTC is initiated by a transmit empty interrupt (TXI) request writes data to TDR. Similarly, RDRF flag is cleared automat when the DTC is initiated by a data full interrupt (RXI) and re data from RDR.

Figure 14.23 Sample Flowchart of Simultaneous Serial Transmission and Rece

the SCK pin output to the CLK pin of the IC card. A reset signal can be supplied via the
port of this LSI.

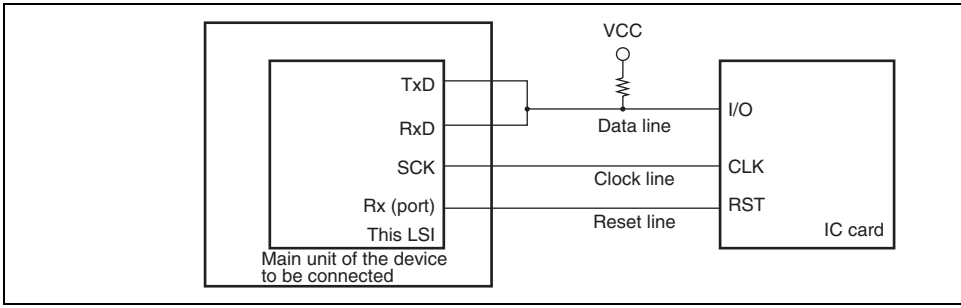


Figure 14.24 Pin Connection for Smart Card Interface

14.7.2 Data Format (Except in Block Transfer Mode)

Figure 14.25 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.

Ds: Start bit
D0 to D7: Data bits
Dp: Parity bit
DE: Error signal

Figure 14.25 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.

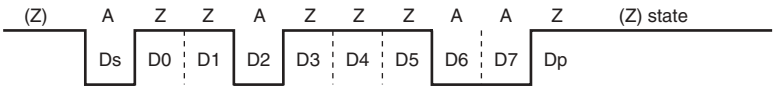


Figure 14.26 Direct Convention (SDIR = SINV = O/$\overline{E}$ = 0)

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 14.26. 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.

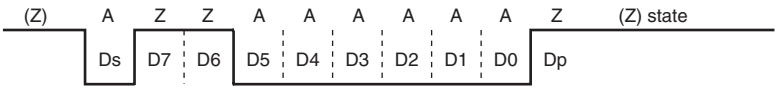


Figure 14.27 Inverse Convention (SDIR = SINV = O/$\overline{E}$ = 1)

For the inverse convention type, logic levels 1 and 0 correspond to states A and Z, respec
and data is transferred with MSB-first as the start character, as shown in figure 14.27. Th
data in the start character in the figure is H'3F. When using the inverse convention type, v
both the SDIR and SINV bits in SCMR. The parity bit is logic level 0 to produce even pa

Since the same data is not transmitted during transmission, the TEND flag in SSR
etu after transmission start.

- Although the ERS flag in block transfer mode displays the error signal status as in no
smart card interface mode, the flag is always read as 0 because no error signal is tran

14.7.4 Receive Data Sampling Timing and Reception Margin

Only the internal clock generated by the internal baud rate generator can be used as a communication clock in smart card interface mode. In this mode, the SCI can operate us clock with a frequency of 32, 64, 372, or 256 times the bit rate according to the BCP1 an settings (the frequency is always 16 times the bit rate in normal asynchronous mode). A reception, the falling edge of the start bit is sampled using the internal basic clock in ord perform internal synchronization. Receive data is sampled at the 16th, 32nd, 186th and 1 rising edges of the basic clock pulses so that it can be latched at the center of each bit as figure 14.28. The reception margin here is determined by the following formula.

$$M = \left| (0.5 - \frac{1}{2N}) - (L - 0.5)\,F - \frac{|D - 0.5|}{N}(1 + F) \right| \times 100\ [\%] \quad \cdots \quad \text{Formula (}$$

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.

$$M = (0.5 - 1/2 \times 372) \times 100\ [\%] = 49.866\%$$

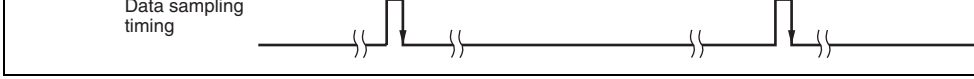


**Figure 14.28 Receive Data Sampling Timing in Smart Card Interface Mode (Whe
Frequency is 372 Times the Bit Rate)**

14.7.5 Initialization

Before starting transmitting and receiving data, initialize the SCI using the following proc
Initialization is also necessary before switching from transmission to reception and vice v

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. A
the PE bit to 1.
4. Set the SMIF, SDIR, and SINV bits in SCMR appropriately. When the SMIF bit is se
TxD and RxD pins are changed from port pins to SCI pins, placing the pins into high
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, MPI
TEIE bits to 0 simultaneously. When the CKE0 bit is set to 1, the SCK pin is allowed
clock pulses.
7. Set the TIE, RIE, TE, and RE bits in SCR appropriately after waiting for at least 1 bit
Setting prohibited the TE and RE bits to 1 simultaneously except for self diagnosis.

To switch from reception to transmission, first verify that reception has completed, and in
the SCI. At the end of initialization, RE and TE should be set to 0 and 1, respectively. Re
completion can be verified by reading the RDRF flag or PER and ORER flags. To switch
transmission to reception, first verify that transmission has completed, and initialize the S

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 case, one frame of data is determined to have been transmitted including re-transfer, TEND bit in SSR is set to 1. Here, a TXI interrupt request is generated if the TIE bit set to 1. Writing transmit data to TDR starts transmission of the next data.

Figure 14.31 shows a sample flowchart for transmission. All the processing steps are automatically performed using a TXI interrupt request to activate the DTC. In transmiss 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 transmit data if the TXI interrupt request is specified as a source of DTC activation befo The TDRE and TEND flags are automatically cleared to 0 at data transfer by the DTC. I 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 th specified number of bytes, including re-transmission in the case of error occurrence. Ho ERS flag is not automatically cleared; the ERS flag must be cleared by previously settin 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 ma settings. For DTC settings, see section 7, Data Transfer Controller (DTC).

Note that the TEND flag is set in different timings depending on the GM bit setting in SM
which is shown in figure 14.30.

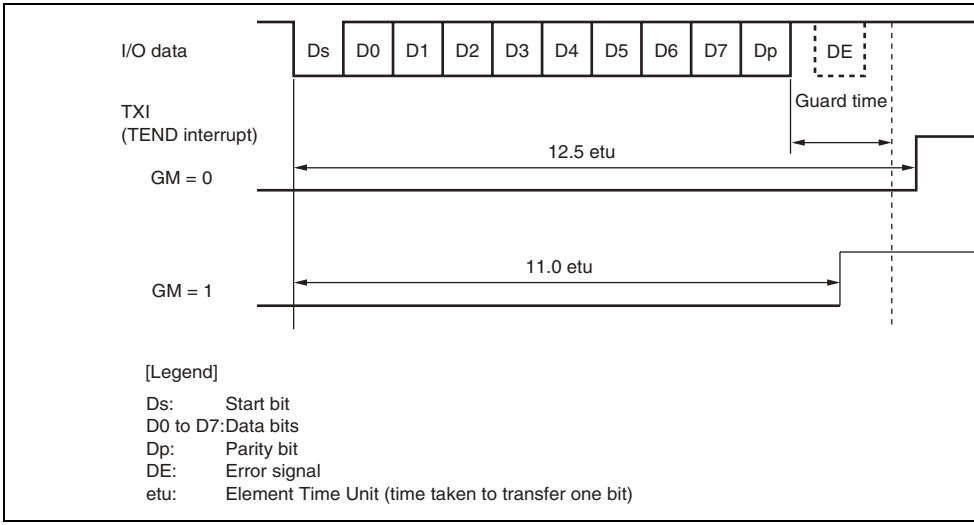


Figure 14.30 TEND Flag Set Timings during Transmission

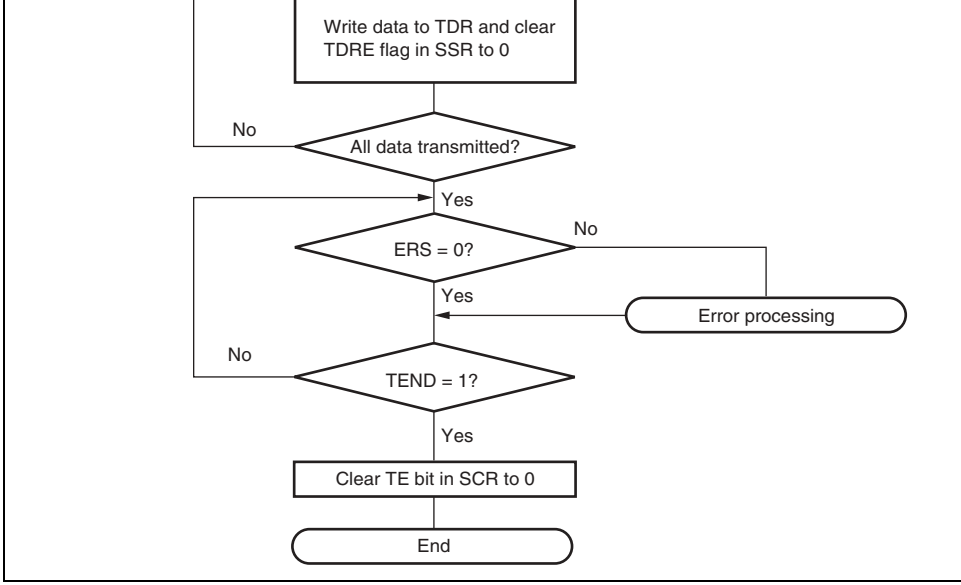


Figure 14.31 Sample Transmission Flowchart

Figure 14.33 shows a sample flowchart for reception. All the processing steps are automa
performed using an RXI interrupt request to activate the DTC. In reception, setting the RI
allows an RXI interrupt request to be generated when the RDRF flag is set to 1. This activ
DTC by an RXI request thus allowing transfer of receive data if the RXI interrupt request
specified as a source of DTC activate beforehand. The RDRF flag is automatically cleare
data transfer by DTC. If an error occurs during reception, i.e., either the ORER or PER fl
to 1, a transmit/receive error interrupt (ERI) request is generated and the error flag must b
cleared. If an error occurs, DTC is not activated and receive data is skipped, therefore, the
of bytes of receive data specified in DTC are transferred. Even if a parity error occurs and
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 14.4, Operation in Asynchronc

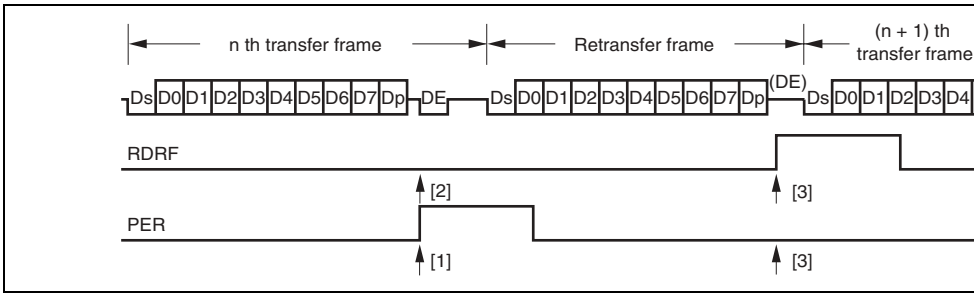


Figure 14.32 Data Re-transfer Operation in SCI Reception Mode

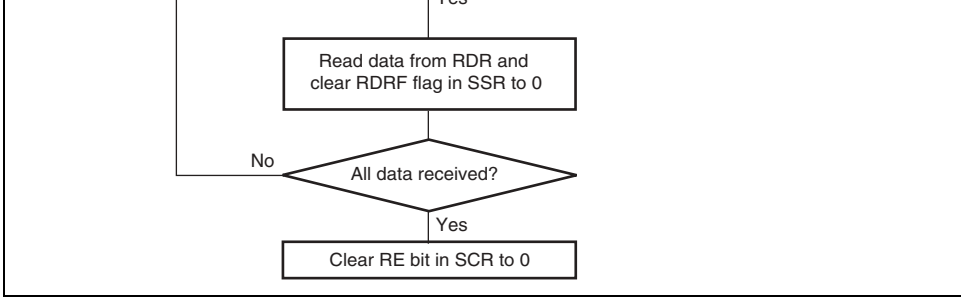


Figure 14.33 Sample Reception Flowchart

14.7.8 Clock Output Control

Clock output can be fixed using the CKE1 and CKE0 bits in SCR when the GM bit in S to 1. Specifically, the minimum width of a clock pulse can be specified.

Figure 14.34 shows an example of clock output fixing timing when the CKE0 bit is cont with GM = 1 and CKE1 = 0.

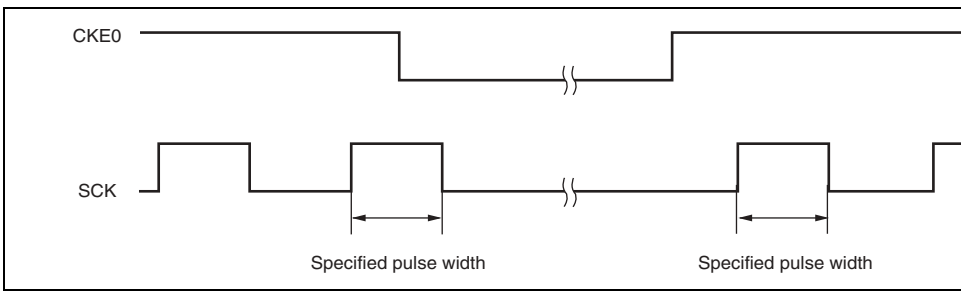


Figure 14.34 Clock Output Fixing Timing

At Transition from Smart Card Interface Mode to Software Standby Mode:

1. Set the port data register (DR) and data direction register (DDR) corresponding to the pins to the values for the output fixed state in software standby mode.
2. Write 0 to the TE and RE bits in SCR to stop transmission/reception. Simultaneously, CKE1 bit to the value for the output fixed state in software standby mode.
3. Write 0 to the CKE0 bit in SCR to stop the clock.
4. Wait for one cycle of the serial clock. In the mean time, the clock output is fixed to th specified level with the duty ratio retained.
5. Make the transition to software standby mode.

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 appropr ratio is then generated.

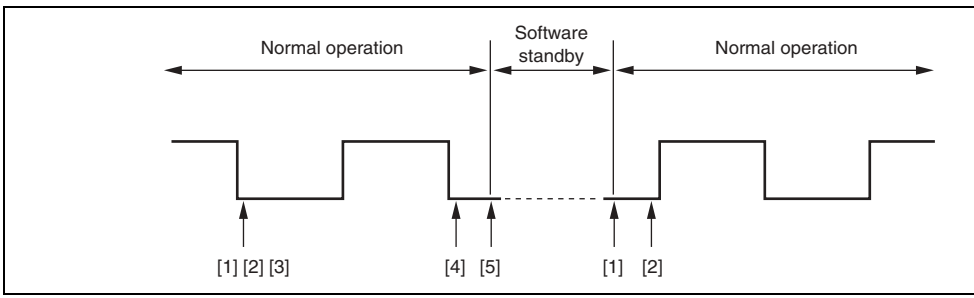


Figure 14.35 Clock Stop and Restart Procedure

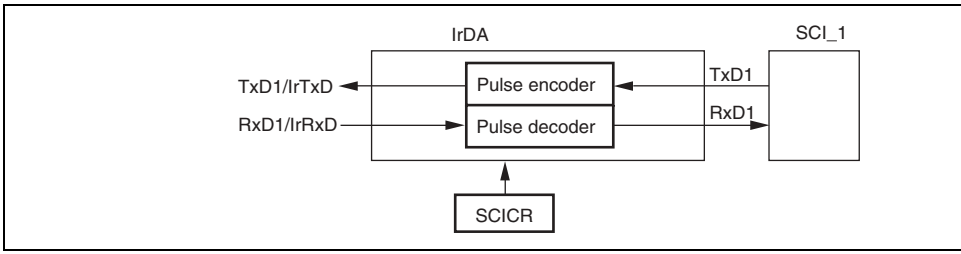


Figure 14.36 IrDA Block Diagram

Transmission: During transmission, the output signals from the SCI (UART frames) ar converted to IR frames using the IrDA interface (see figure 14.37).

For serial data of level 0, a high-level pulse having a width of 3/16 of the bit rate (1-bit i output (initial setting). The high-level pulse can be selected using the IrCKS2 to IrCKS0 SCICR.

The high-level pulse width is defined to be 1.41 µs at minimum and (3/16 + 2.5%) × bit (3/16 × bit rate) +1.08 µs at maximum. For example, when the frequency of system cloc MHz, a high-level pulse width of at least 1.41 µs to 1.6 µs can be specified.

For serial data of level 1, no pulses are output.

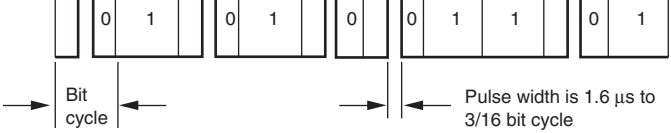


Figure 14.37 IrDA Transmission and Reception

Reception: During reception, IR frames are converted to UART frames using the IrDA in
before inputting to SCI_1.

Data of level 0 is output each time a high-level pulse is detected and data of level 1 is out
no pulse is detected in a bit cycle. If a pulse has a high-level width of less than 1.41 µs, th
minimum width allowed, the pulse is recognized as level 0.

High-Level Pulse Width Selection: Table 14.13 shows possible settings for bits IrCKS2
IrCKS0 (minimum pulse width), and this LSI's operating frequencies and bit rates, for ma
pulse width shorter than 3/16 times the bit rate in transmission.

10	100	100	100	100	100	10
12	101	101	101	101	101	10
12.288	101	101	101	101	101	10
14	101	101	101	101	101	10
14.7456	101	101	101	101	101	10
16	101	101	101	101	101	10
16.9344	101	101	101	101	101	10
17.2032	101	101	101	101	101	10
18	101	101	101	101	101	10
19.6608	101	101	101	101	101	10
20	101	101	101	101	101	10
25	110	110	110	110	110	11
33	110	110	110	110	110	11

When the RDRF flag in SSR is set to 1, an RXI interrupt request is generated. When the
PER, or FER flag in SSR is set to 1, an ERI interrupt request is generated. An RXI interru
activate the DTC to allow data transfer. The RDRF flag is automatically cleared to 0 at da
transfer by the DTC.

A TEI interrupt is requested when the TEND flag is set to 1 while the TEIE bit is set to 1
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 14.14 SCI Interrupt Sources

Channel	Name	Interrupt Source	Interrupt Flag	DTC Activation
0	ERI0	Receive error	ORER, FER, PER	Not possible
	RXI0	Receive data full	RDRF	Possible
	TXI0	Transmit data empty	TDRE	Possible
	TEI0	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
2	ERI2	Receive error	ORER, FER, PER	Not possible
	RXI2	Receive data full	RDRF	Possible
	TXI2	Transmit data empty	TDRE	Possible
	TEI2	Transmit end	TEND	Not 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
2	ERI2	Receive error, error signal detection	ORER, PER, ERS	Not possible
	RXI2	Receive data full	RDRF	Possible
	TXI2	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 simu 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 sourc 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 S DTC automatically transmit the specified number of bytes, including re-transmission in 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 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 prio 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 activates the DTC by an RXI interrupt request thus allowing transfer of receive data if th interrupt request is specified as a source of DTC activation beforehand. The RDRF flag automatically cleared to 0 at data transfer by the DTC. If an error occurs, the RDRF flag 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.

and the PER flag may also be set. Note that, since the SCI continues the receive operation after receiving a break, even if the FER flag is cleared to 0, it will be set to 1 again.

14.10.3 Mark State and Break Sending

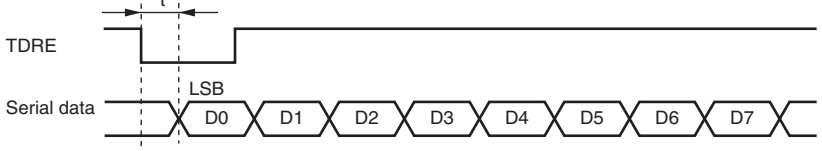
When the TE bit in SCR is 0, the TxD pin is used as an I/O port whose direction (input or and level are determined by DR and DDR of the port. This can be used to set the TxD pin state (high level) or send a break during serial data transmission. To maintain the commu line at mark state until TE is set to 1, set both DDR and DR to 1. Since the TE bit is clear 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 s TxD pin becomes an I/O port, and 0 is output from the TxD pin.

14.10.4 Receive Error Flags and Transmit Operations (Clock Synchronous Mode

Transmission cannot be started when a receive error flag (ORER, FER, or RER) in SSR i even if the TDRE flag in SSR is cleared to 0. Be sure to clear the receive error flags to 0 b starting transmission. Note also that the receive error flags cannot be cleared to 0 even if bit in SCR is cleared to 0.

14.10.5 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 14.38 Sample Transmission using DTC in Clock Synchronous Mo

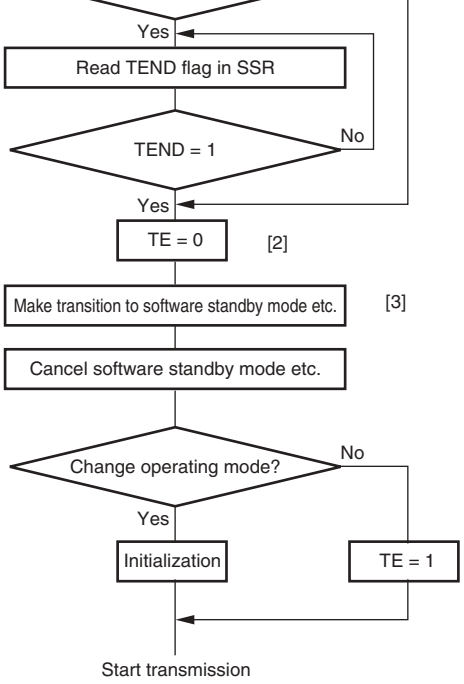
14.10.7 SCI Operations during Mode Transitions

Transmission: Before making the transition to module stop, software standby, or sub-sl stop all transmit operations (TE = TIE = TEIE = 0). TSR, TDR, and SSR are reset. The the output pins during each mode depend on the port settings, and the pins output a high signal after mode cancellation. If the transition is made during data transmission, the dat transmitted will be undefined.

To transmit data in the same transmission mode after mode cancellation, set TE to 1, rea write to TDR, clear TDRE in this order, and then start transmission. To transmit data in transmission mode, initialize the SCI first.

Figure 14.39 shows a sample flowchart for mode transition during transmission. Figures 14.41 show the pin states during transmission.

Before making the transition from the transmission mode using DTC transfer to module software standby, or sub-sleep mode, stop all transmit operations (TE = TIE = TEIE = 0 TE and TIE to 1 after mode cancellation generates a TXI interrupt request to start transn using the DTC.

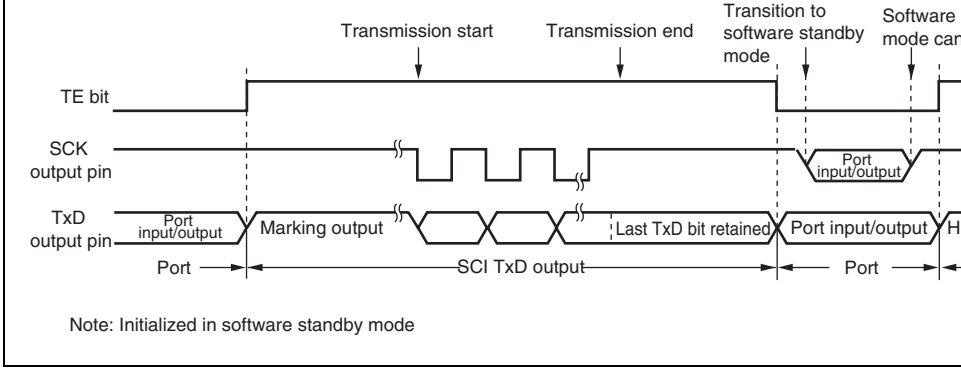


transmitted from the CPU by setting TE to 1, reading SSR, writing to TDR, and clearing TDRE to 0 after mode cancellation; however, if the DTC has been initiated, the data remaining in DTC RAM will be transmitted when TE and TIE are set to 1.

[2] Also clear TIE and TEIE to 0 when they are 1.

[3] Module stop, watch, sub-active, and sub-sleep modes are included.

Figure 14.39 Sample Flowchart for Mode Transition during Transmission



Note: Initialized in software standby mode

Figure 14.41 Pin States during Transmission in Clock Synchronous Mod (Internal Clock)

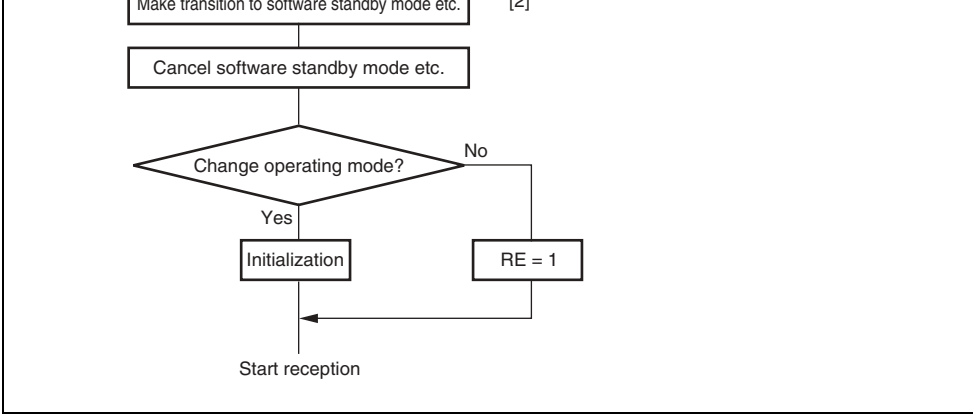


Figure 14.42 Sample Flowchart for Mode Transition during Reception

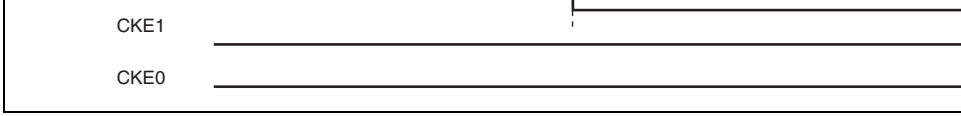


Figure 14.43 Switching from SCK Pins to Port Pins

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

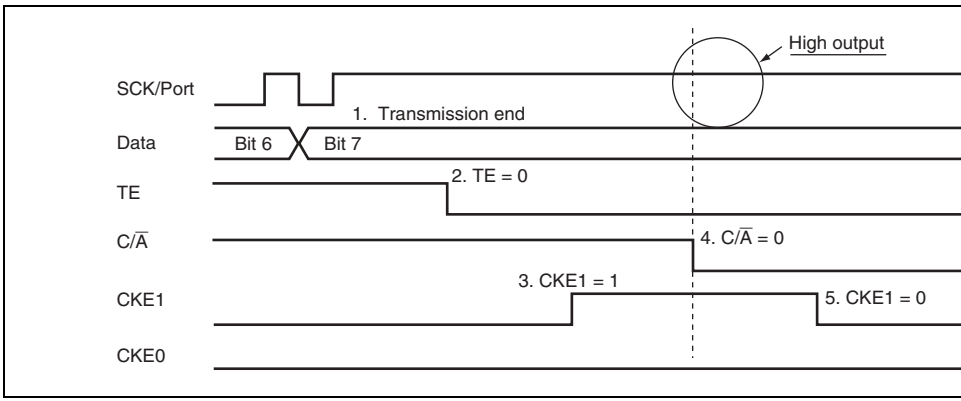


Figure 14.44 Prevention of Low Pulse Output at Switching from SCK Pins to P

Figure 14.45 shows a block diagram of the CRC operation circuit.

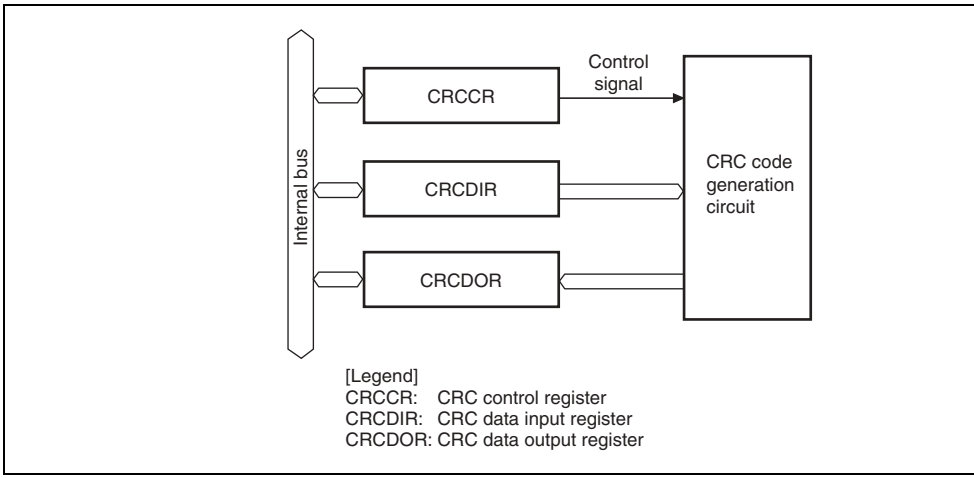


Figure 14.45 Block Diagram of CRC Operation Circuit

14.11.2 Register Descriptions

The CRC operation circuit has the following registers.

- CRC control register (CRCCR)
- CRC data input register (CRCDIR)
- CRC data output register (CRCDOR)

				communication. The lower byte (bits 7 t transmitted when CRCDOR contents (C are divided into two bytes to be transmi parts. 1: Performs CRC operation for MSB-first communication. The upper byte (bits 15 first transmitted when CRCDOR conten code) are divided into two bytes to be tr in two parts.
1	G1	0	R/W	CRC Generating Polynomial Select These bits select the polynomial. 00: Reserved 01: $X^8 + X^2 + X + 1$ 10: $X^{16} + X^{15} + X^2 + 1$ 11: $X^{16} + X^{12} + X^5 + 1$
0	G0	0	R/W	

CRC Data Input Register (CRCDIR): CRCDIR is an 8-bit readable/writable register, the bytes to be CRC-operated are written. The result is obtained in CRCDOR.

CRC Data Output Register (CRCDOR): CRCDOR is a 16-bit readable/writable regis contains the result of CRC operation when the bytes to be CRC-operated are written to C after CRCDOR is cleared. When the CRC operation result is additionally written to the which CRC operation is to be performed, the CRC operation result will be H'0000 if the contains no CRC error. When bits 1 and 0 in CRCCR (G1 and G0 bits) are set to 0 and 1 respectively, the lower byte of this register contains the result.

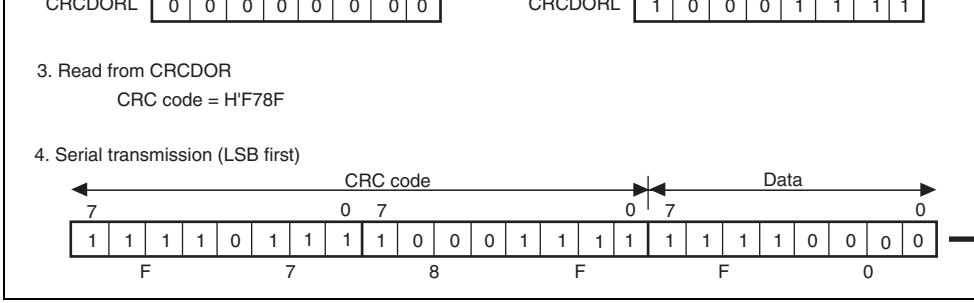


Figure 14.46 LSB-First Data Transmission

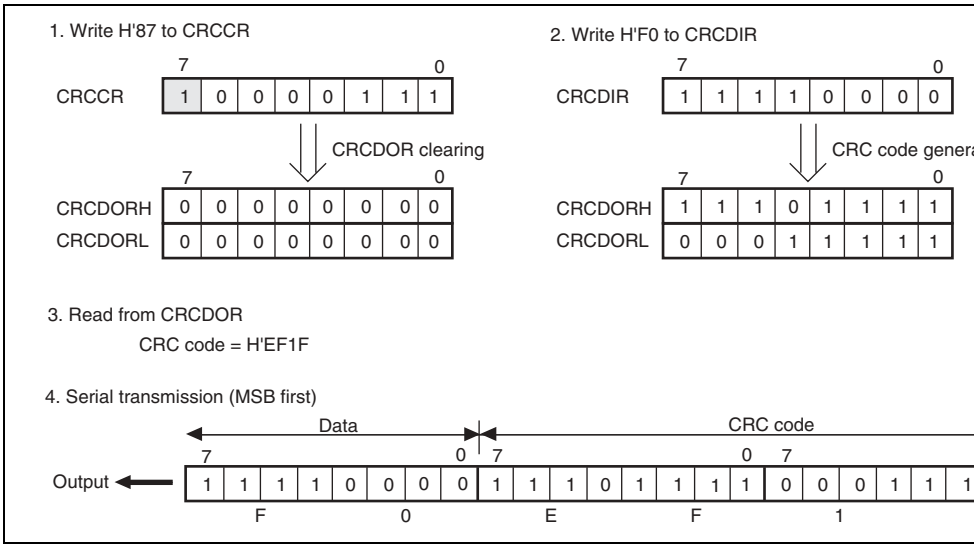


Figure 14.47 MSB-First Data Transmission

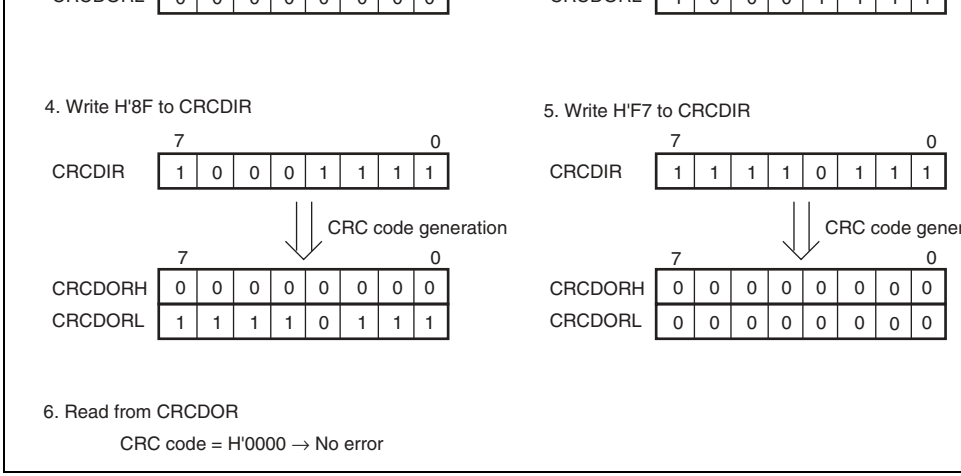


Figure 14.48 LSB-First Data Reception

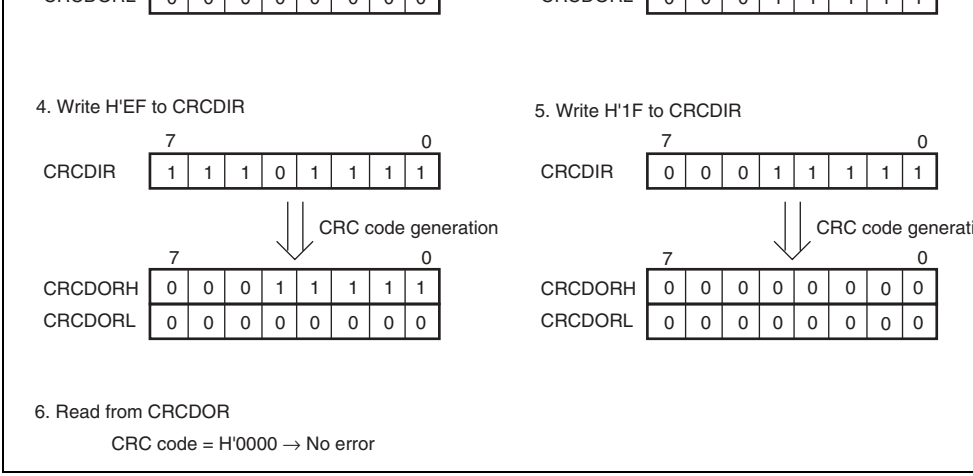


Figure 14.49 MSB-First Data Reception

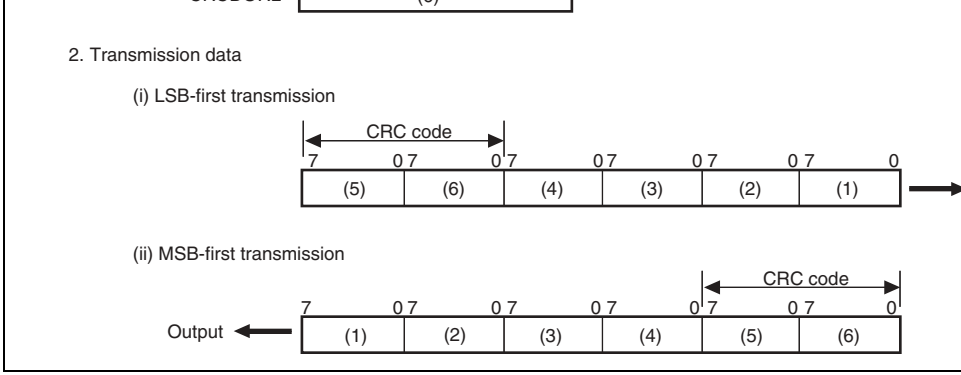


Figure 14.50 LSB-First and MSB-First Transmit Data

- 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 form 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
  - Pins—SCL0 to SCL5 and SDA0 to SDA5 —(normally NMOS push-pull outputs as NMOS open-drain outputs when the bus drive function is selected.

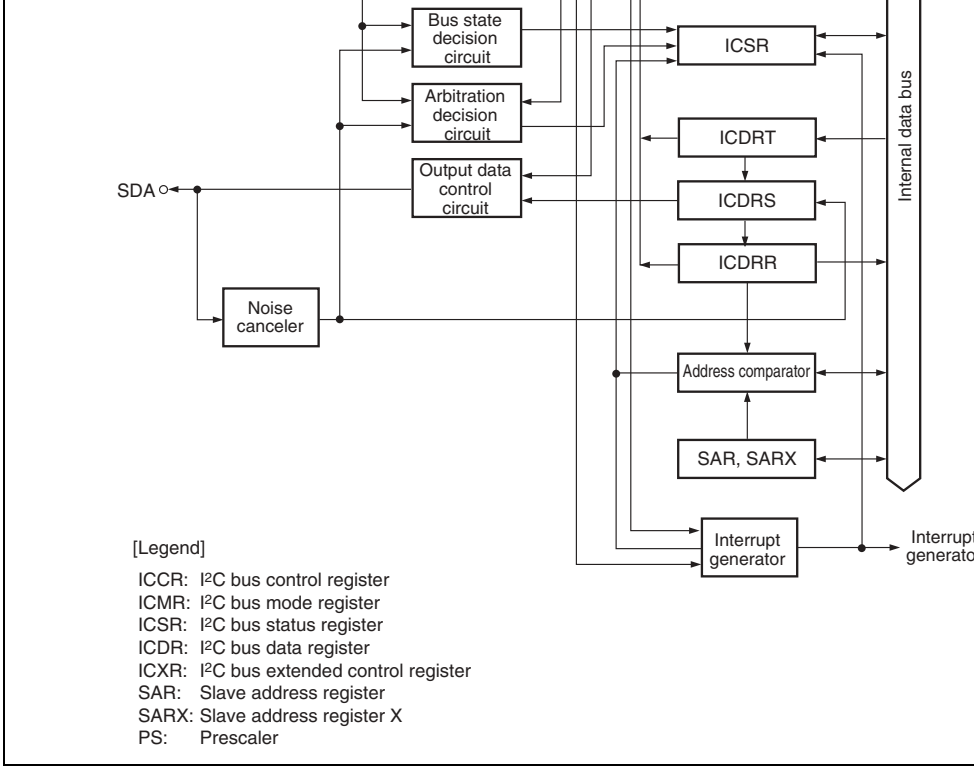


Figure 15.1 Block Diagram of I²C Bus Interface

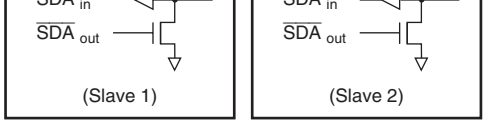


Figure 15.2 I²C Bus Interface Connections (Example: This LSI as Master

	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)

15.3.1 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 wr ICDR in receive mode.

Reading receive data from ICDR is performed after data is transferred from ICDRS to IC

15.3.2 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 Address
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 th bit in SARX. Refer to table 15.2.
				This bit should be set to 0 when general call addre recognition is performed.

5	SVAX4			
4	SVAX3			
3	SVAX2			
2	SVAX1			
1	SVAX0			Set the second slave address.
0	FSX	1	R/W	Format Select X Selects the communication format together with t in SAR. Refer to table 15.2.

Table 15.2 Transfer Format

SAR FS	SARX FSX	Operating Mode
0	0	I^2C bus format • SAR and SARX slave addresses recognized • General call address recognized
	1	I^2C bus format • SAR slave address recognized • SARX slave address ignored • 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

				1: LSB-first Set this bit to 0 when the I^2C bus format is used.
6	WAIT	0	R/W	Wait Insertion Bit This bit is valid only in master mode with the I^2C b 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 (8t the IRIC flag is set to 1 in ICCR, and a wait stat (with SCL at the low level). When the IRIC flag i cleared to 0 in ICCR, the wait ends and the acknowledge bit is transferred. For details, refer to section 15.4.7, IRC Setting Tir SCL Control.
5 4 3	CKS2 CKS1 CKS0	All 0	R/W	Transfer Clock Select These bits are used only in master mode. These bits select the required transfer rate, togeth the IICX5 (channel 5), IICX4 (channel 4), and IICX (channel 3) bits in IICX3, and the IICX2 (channel 2 (channel 1), and IICX0 (channel 0) bits in STCR. F table 15.3.

	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

15.3.5 I²C Bus Transfer Rate Select Register (IICX3)

IICX3 selects the IIC transfer rate clock and sets the transfer rate of IIC channels 3 to 5.

Bit	Bit Name	Initial Value	R/W	Description
7 to 4	—	—	—	Reserved These bits cannot be modified.
3	TCSS	0	R/W	Transfer Rate Clock Source Select This bit selects a clock rate to be applied to the I² transfer rate. 0: ϕ/2 1: ϕ/4
2 1 0	IICX5 IICX4 IICX3	All 0	R/W	IIC Transfer Rate Select These bits are used to control IIC operation. These bits select the transfer rate in master mod together with the CKS2 to CKS0 bits in ICMR. Fo transfer rate, see table 15.3. IICX5, IICX4, and II control IIC_5, IIC_4, and IIC_3, respectively

			1	φ/100	50.0	80.0	100.0	160.0	200.0	250.0
		1	0	φ/112	44.6	71.4	89.3	142.9	178.6	223.2
			1	φ/128	39.1	62.5	78.1	125.0	156.3	195.3
1	0	0	0	φ/56	89.3	142.9	178.6	285.7	357.1	446.4*[1]
			1	φ/80	62.5	100.0	125.0	200.0	250.0	312.5
		1	0	φ/96	52.1	83.3	104.2	166.7	208.3	260.4
			1	φ/128	39.1	62.5	78.1	125.0	156.3	195.3
	1	0	0	φ/160	31.3	50.0	62.5	100.0	125.0	156.3
			1	φ/200	25.0	40.0	50.0	80.0	100.0	125.0
		1	0	φ/224	22.3	35.7	44.6	71.4	89.3	111.6
			1	φ/256	19.5	31.3	39.1	62.5	78.1	97.7

Notes: 1. The correct operation cannot be guaranteed since the value is outside the I^2C interface specifications (high-speed mode: max. 400 kHz)

2. When operate IIC in this setting, see 5 in section 15.6, Usage Notes.
(n = 0 to 5)

			1	φ/200	25.0	40.0	50.0	80.0	100.0	125.0
		1	0	φ/224	22.3	35.7	44.6	71.4	89.3	111.6
			1	φ/256	19.5	31.3	39.1	62.5	78.1	97.7
1	0	0	0	φ/112	44.6	71.4	89.3	142.9	178.6	223.2
			1	φ/160	31.3	50.0	62.5	100.0	125.0	156.3
		1	0	φ/190	26.0	41.7	52.1	83.3	104.2	130.2
			1	φ/256	19.5	31.3	39.1	62.5	78.1	97.7
	1	0	0	φ/320	15.6	25.0	31.3	50.0	62.5	78.1
			1	φ/400	12.5	20.0	25.0	40.0	50.0	62.5
		1	0	φ/448	11.2	17.9	22.3	35.7	44.6	55.8
			1	φ/512	9.8	15.6	19.5	31.3	39.1	48.8

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)

				accessed.
6	IEIC	0	R/W	I²C Bus Interface Interrupt Enable 0: Disables interrupts from the I²C bus interface to 1: Enables interrupts from the I²C bus interface to
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 in a bus contention in master mode of the I²C bus In slave receive mode with I²C bus format, the R/$\overline{W}$ the first frame immediately after the start condition automatically sets these bits in receive mode or tr mode by hardware. Modification of the TRS bit during transfer is defer transfer is completed, and the changeover is mad completion of the transfer.

				(1) When 0 is written by software (except for TRS 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 TRS 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 c received acknowledge bit is not indicated by bit in ICSR, which is always 0. 1: If the acknowledge bit is 1, continuous transfe halted. Depending on the receiving device, the acknowle may be significant, in indicating completion of pr of the received data, for instance, or may be fixe have no significance.

[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 instru

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.

after a start condition is issued (when the ICDI
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 of
transmit/receive clock)
- At the end of data transfer (rise of the 9th
transmit/receive clock)
- When a slave address is received after bus ma
is lost
- If 1 is received as the acknowledge bit (when t
bit in ICSR is set to 1) when the ACKE bit is 1
- When the AL flag is set to 1 after bus mastersl
while the ALIE bit is 1

I^2C bus format slave mode:

- When the slave address (SVA or SVAX) matcl
the AAS or AASX flag in ICSR is set to 1) and
end of data transfer up to the subsequent
retransmission start condition or stop condition
(rise of the 9th clock)
- When the general call address is detected (wh
is received for R/$\overline{W}$ bit, and ADZ flag in ICSR i
and at the end of data reception up to the subs
retransmission start condition or stop condition
(rise of the 9th receive clock)
- When 1 is received as an acknowledge bit whi
ACKE bit is 1 (when the ACKB bit is set to 1)
- When a stop condition is detected while the ST
is 0 (when the STOP or ESTP flag in ICSR is s

(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 *2 (This may
clearing condition. For details, see the descrip
the DTC operation on the next page.

Notes: 1. Only 0 can be written to clear the flag to 0.
2. The DTC does not support IIC_4 and IIC_5.
3. If the BBSY bit is written to, the value of the flag is not changed.

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 15.4 and 15.5 show the relationship between the flags and the transfer states.

													the a
1	1	1	0	0	—	0	0	0	0	0	—	1	Tran end v 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 v s
0	1	1	0	0	—	—	—	—	0	0	—	1	T e I
0	1	1	0	0	—	—	0↓	0↓	0	0	—	0↓	I v s
0	1	1	0	0	1↑/0 *2	—	0	0	0	0	—	1↑	A o fr I tl s
0	0	1	0	0	1↑/0 *2	—	—	—	—	—	1↑	—	P v
0	0	1	0	0	—	—	0↓	0↓	0↓	—	0↓	—	I v s

[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 the IRIC flag in ICCR is cleared to 0
6	STOP	0	R/(W)*	Normal Stop Condition Detection Flag This bit is valid in I²C 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)*	I²C Bus Interface Continuous Transfer Interrupt F Flag Indicates that the I²C 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 i possible. When the IRTR flag is set to 1, the IRIC also set to 1 at the same time. [Setting conditions] I²C bus format slave mode: • When the ICDRE or ICDRF flag in ICDR is se when AASX = 1 I²C 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 i

				• 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 the 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 the SCL in master transmit mode • If the SDA pin is driven low by another device b I^2C bus interface drives the SDA pin low, after t condition instruction was executed in master tra mode [Clearing conditions] • When ICDR is written to (transmit mode) or rea (receive mode) • When 0 is written in AL after reading AL = 1

				(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 ge address (H'00). [Setting condition] When the general call address (one frame includ bit is H'00) is detected in slave receive mode and 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 a FSX=0, the ADZ flag is set to 1; however, the ge address is not recognized (AAS flag is not set to

Receive mode:

0: Returns 0 as acknowledge data after data recep

1: Returns 1 as acknowledge data after data recep

When this bit is read, the value loaded from the bu
(returned by the receiving device) is read in transr
(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-manipulati
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 0
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.

				1: Disables IRIC flag setting and interrupt genera the stop condition is detected.
6	HNDS	0	R/W	Handshake Receive Operation Select Enables or disables continuous receive operation receive mode. 0: Enables continuous receive operation 1: Disables continuous receive operation When the HNDS bit is cleared to 0, receive opera performed continuously after data has been rece 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 w ICDRF flag is 0; thus disabling the next data to b transferred. The bus line is released and next rec operation is enabled by reading the receive data

(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 fr
ICDRS to ICDRR. Be sure to read data from ICDF
receive mode (TRS = 0).

- 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 IC

[Clearing conditions]

- When data is written to ICDR (ICDRT).
- When the stop condition is detected in I^2C bu 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 for enabling acknowledge bit decision, ICDRE is not data is transmitted completely while the acknowl 1.

When ICDRE is set due to the condition (2) abov 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.

				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 15.6, Usage Notes. 00: Restrictions on operation remaining in effect 01: Setting prohibited 10: Setting prohibited 11: Restrictions on operation canceled

6 5 4 3 2	SMB4E SMB3E SMB2E SMB1E SMB0E			These bits enable/disable to support the SMBus, combining with bits FSEL1 and FSEL0. The SMB controls IIC_5, the SMB4E bit controls IIC_4, the bit controls IIC_3, the SMB2E bit controls IIC_2, SMB1E bit controls IIC_1, the SMB0E bit control 0: Disables to support the SMBus 1: Enables to support the SMBus
1 0	FSEL1 FSEL0	0 0	R/W R/W	Frequency Selection These bits must be specified to match the systen frequency in order to support the SMBus. For de setting, see table 15.7.

	Max.	2200*	1667*	1375*	1100*	827	688	550	440
1	Min.	2000*	1515*	1250*	1000*	752	625	500	400
	Max.	3800*	2879*	2375*	1900*	1429*	1188*	950	760

Notes: n = 0 to 5

* Since the value is outside the SMBus specification, it should not be set.

Table 15.7 ISCMBCR Setting

System Clock	SMBnE	FSEL1	FSEL0
5 to 6.6 MHz	0	0	0
6.6 to 10 MHz	1	0	0
10 to 13.3 MHz	1	0	1
13.3 to 20 MHz	1	1	0
20 to 33 MHz	1	1	1

n = 0 to 5

The symbols used in figures 15.3 to 15.5 are explained in table 15.8.

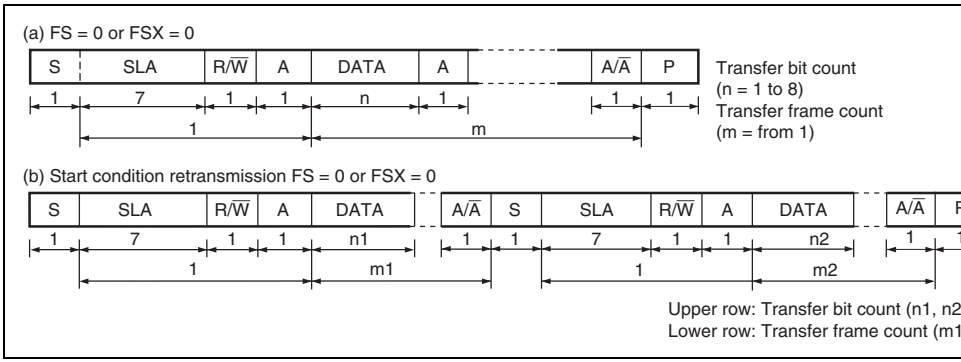


Figure 15.3 I²C Bus Data Formats (I²C Bus Formats)

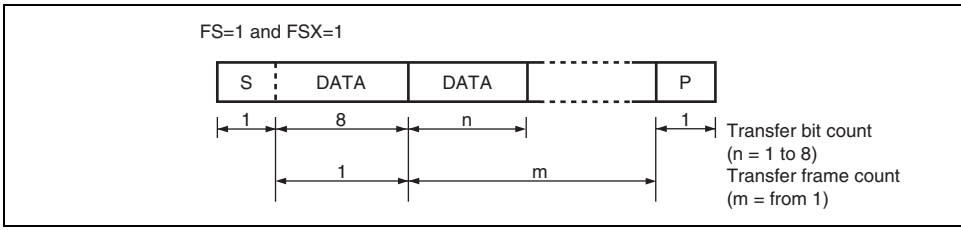


Figure 15.4 I²C Bus Data Formats (Serial Formats)

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

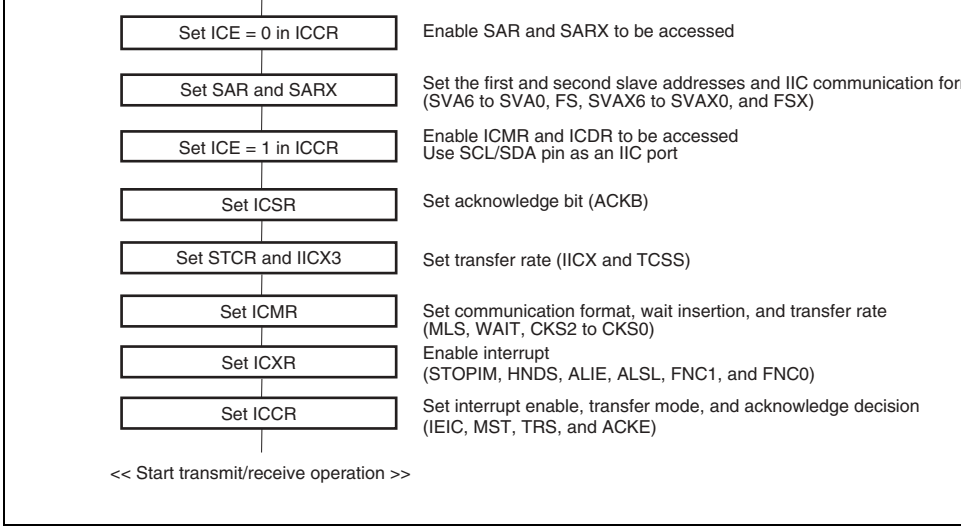


Figure 15.6 Sample Flowchart for IIC Initialization

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.

15.4.3 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.

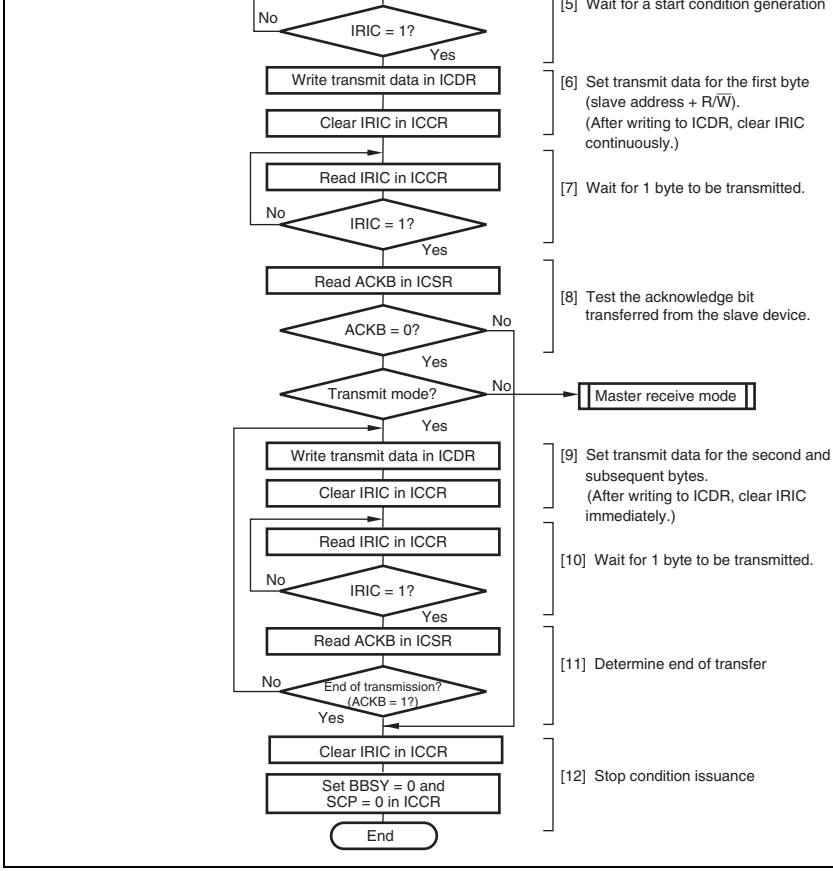


Figure 15.7 Sample Flowchart for Operations in Master Transmit Mode

The transmission procedure and operations by which data is sequentially transmitted in synchronization with ICDR (ICDRT) write operations, are described below.

clear IRIC continuously so no other interrupt handling routine is executed. If the ti
transmission of one frame of data has passed before the IRIC clearing, the end of
transmission cannot be determined. The master device sequentially sends the transf
clock and the data written to ICDR. The selected slave device (i.e. the slave device
matching slave address) drives SDA low at the 9th transmit clock pulse and returns
acknowledge signal.

[7] When one frame of data has been transmitted, the IRIC flag is set to 1 at the rise of
transmit clock pulse. After one frame has been transmitted, SCL is automatically fi
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
has not acknowledged (ACKB bit is 1), operate step [12] to end transmission, and r
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 ICD
and the IRIC flag clearing sequentially, just as in step [6]. Transmission of the next
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
transmit clock pulse. After one frame has been transmitted, SCL is automatically fi
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
data to be transmitted, go to step [9] to continue the next transmission operation. W
slave device has not acknowledged (ACKB bit is set to 1), operate step [12] to end
transmission.

[12] Clear the IRIC flag to 0.

Write 0 to ACKE in ICCR, to clear received ACKB contents to 0. Write 0 to BBSY
in ICCR. This changes SDA from low to high when SCL is high, and generates the
condition.

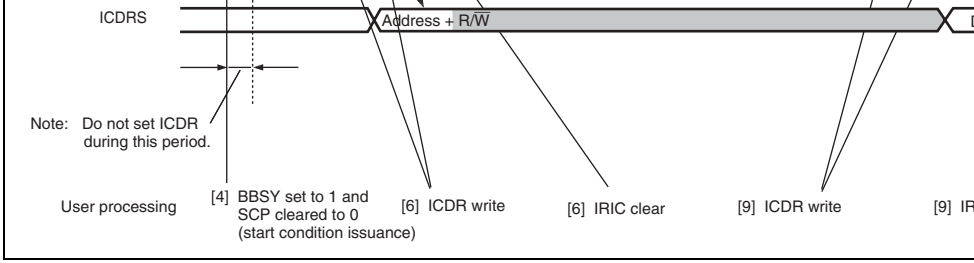


Figure 15.8 Operation Timing Example in Master Transmit Mode (MLS = WAI

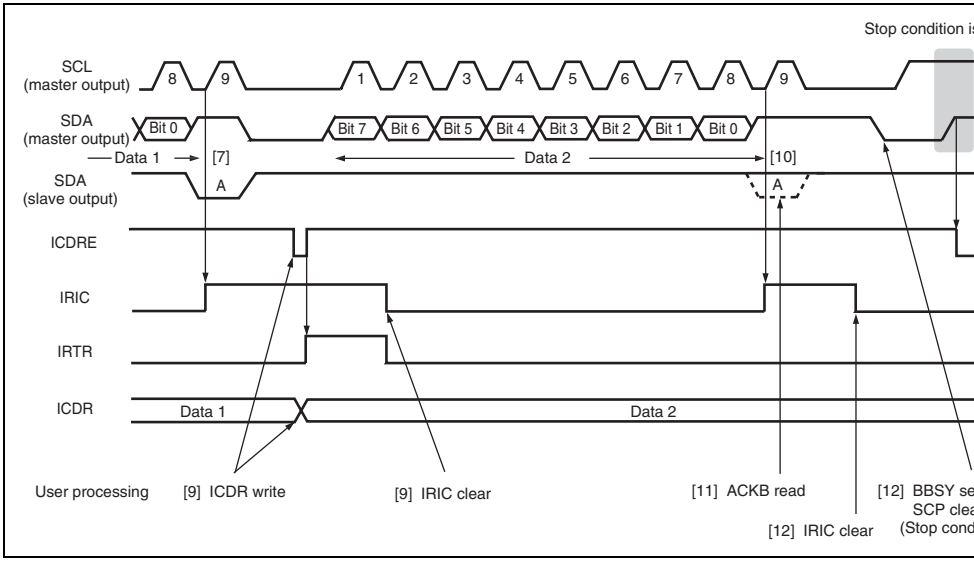


Figure 15.9 Stop Condition Issuance Operation Timing Example in Master Transm (MLS = WAIT = 0)

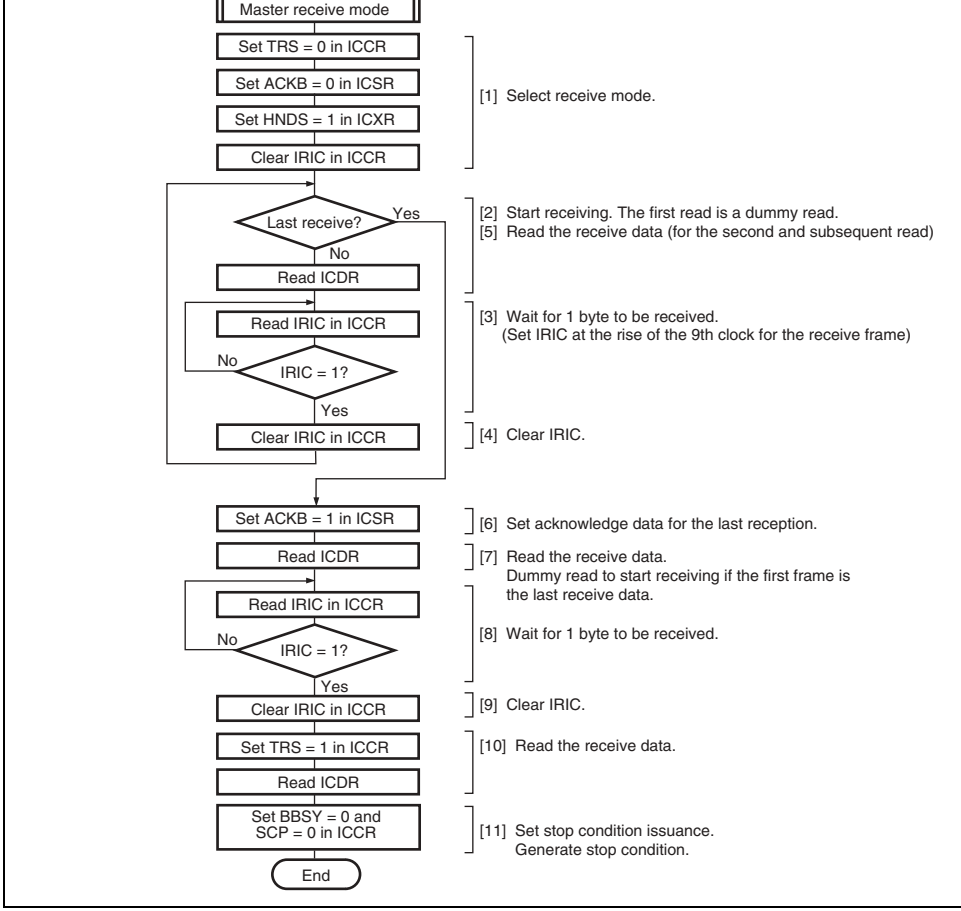


Figure 15.10 Sample Flowchart for Operations in Master Receive Mode (HND

[3] The master device drives SDA low to return the acknowledge data at the 9th recei pulse. The receive data is transferred to ICDRR from ICDRS at the rise of the 9th pulse, setting the ICDRF, IRIC, and IRTR flags to 1. If the IEIC bit has been set t interrupt request is sent to the CPU.

The master device drives SCL low from the fall of the 9th receive clock pulse to tl data 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 outpu 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 outpu receive clock to receive data.

[8] When one frame of data has been received, the ICDRF, IRIC, and IRTR flags are the 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 SCL is high, and generates the stop condition.

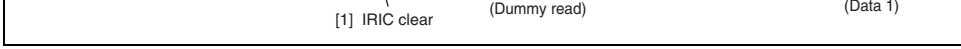


Figure 15.11 Master Receive Mode Operation Timing Example (MLS = WAIT = 0, HNDS = 1)

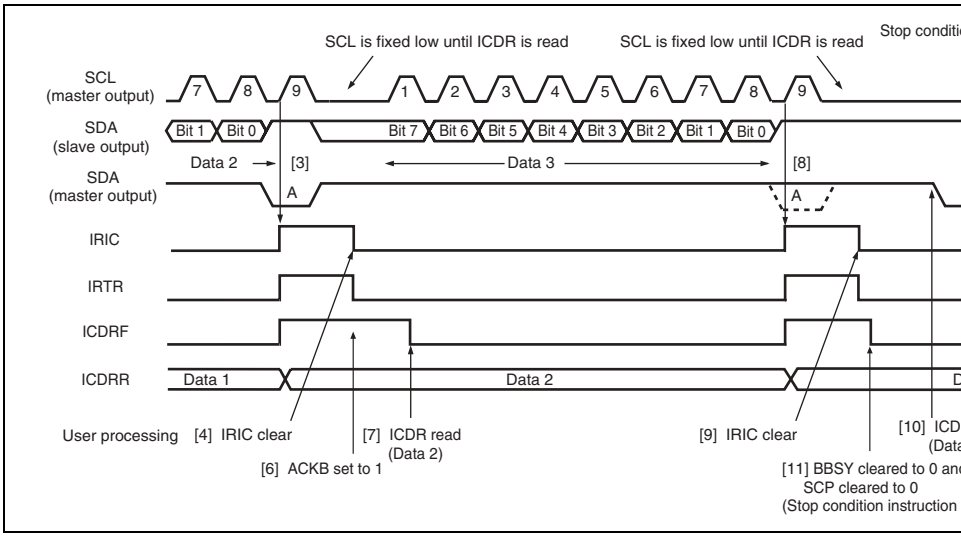


Figure 15.12 Stop Condition Issuance Timing Example in Master Receive M (MLS = WAIT = 0, HNDS = 1)

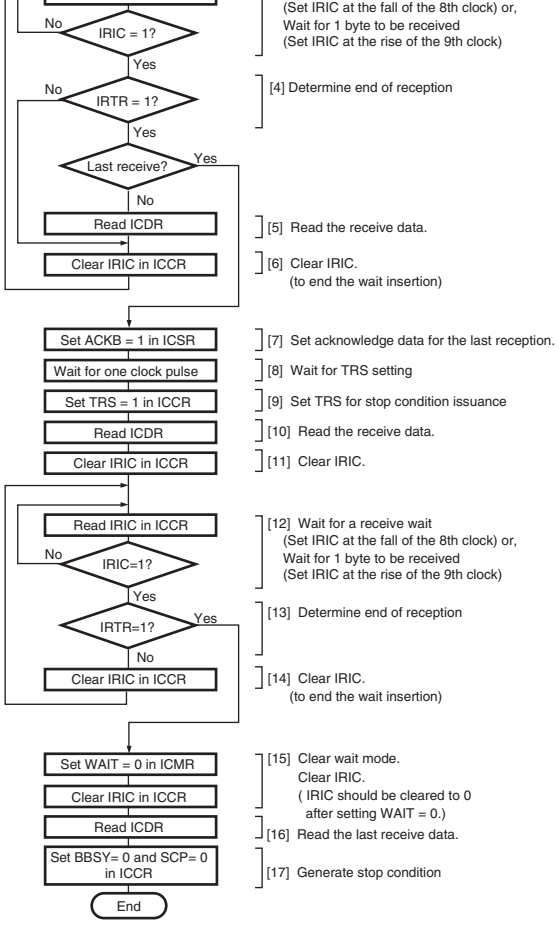


Figure 15.13 Sample Flowchart for Operations in Master Receive Mode (receiving multiple bytes) (WAIT = 1)

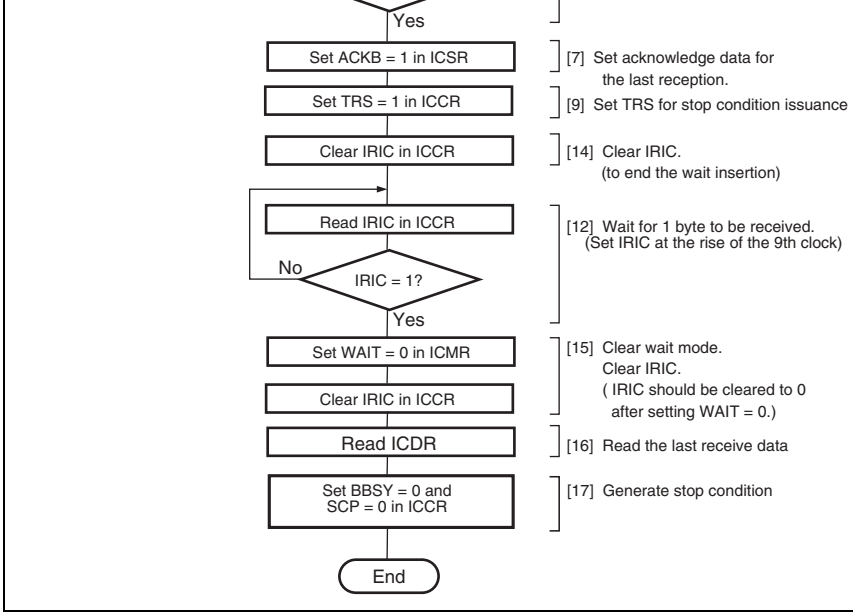


Figure 15.14 Sample Flowchart for Operations in Master Receive Mode (receiving a single byte) (WAIT = 1)

The reception procedure and operations using the wait function (WAIT bit), by which da sequentially received in synchronization with ICDR (ICDRR) read operations, are descr below.

The following describes the multiple-byte reception procedure. In single-byte reception, steps of the following procedure are omitted. At this time, follow the procedure shown i 15.14

(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 b received. The master device outputs the receive clock continuously to receive t 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

If the IRTR flag is 1 and the next data is the last receive data, execute step [7] to h reception.

[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 outpu clock and drives SDA low at the 9th receive clock pulse to return an acknowledge

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 rec

[8] After the IRIC flag is set to 1, wait for at least one clock pulse until the rise of the clock pulse for the next receive data.

[9] Set the TRS bit in ICCR to 1 to switch from receive mode to transmit mode. The T value 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.

[12] The IRIC flag is set to 1 in either of the following cases.

(1) At the fall of the 8th receive clock pulse for one frame

SCL is automatically fixed low in synchronization with the internal clock until flag is cleared.

(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 b received.

[5] Clear the BBSY flag and SCP bit to 0 in ICCR. This changes SDA from low to high when SCL is high, and generates the stop condition.

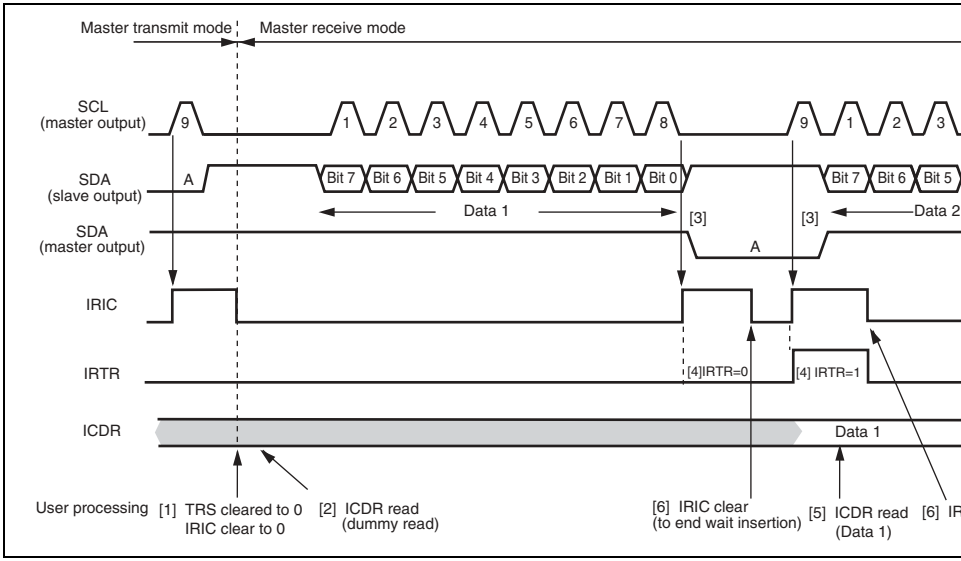


Figure 15.15 Master Receive Mode Operation Timing Example (MLS = ACKB = 0, WAIT = 1)

Figure 15.16 Stop Condition Issuance Timing Example in Master Receive Mo (MLS = ACKB = 0, WAIT = 1)

15.4.5 Slave Receive Operation

In I^2C bus format slave receive mode, the master device outputs the transmit clock and tra data, and the slave device returns an acknowledge signal.

The slave device operates as the device specified by the master device when the slave add the first frame following the start condition that is issued by the master device matches its address.

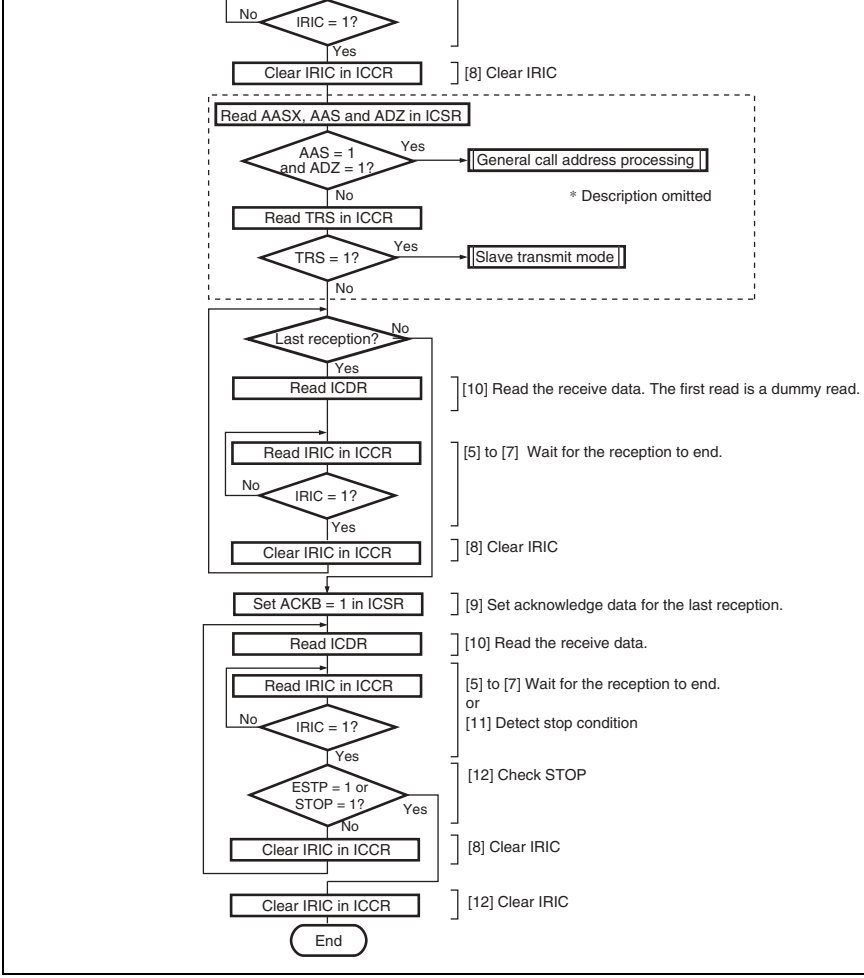


Figure 15.17 Sample Flowchart for Operations in Slave Receive Mode (HND

[4] When the slave address matches in the first frame following the start condition, the 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 da (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 d

[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, 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. The slave device drives SCL low from the fall of the 9t 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 ena 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.

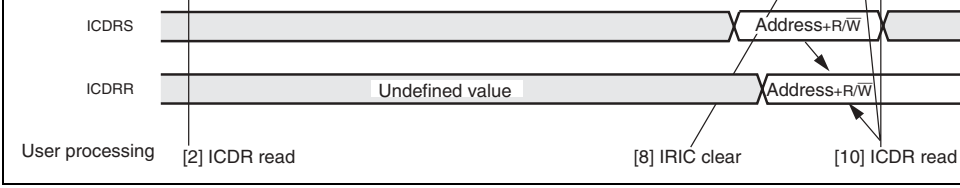


Figure 15.18 Slave Receive Mode Operation Timing Example (1)
(MLS = 0, HNDS= 1)

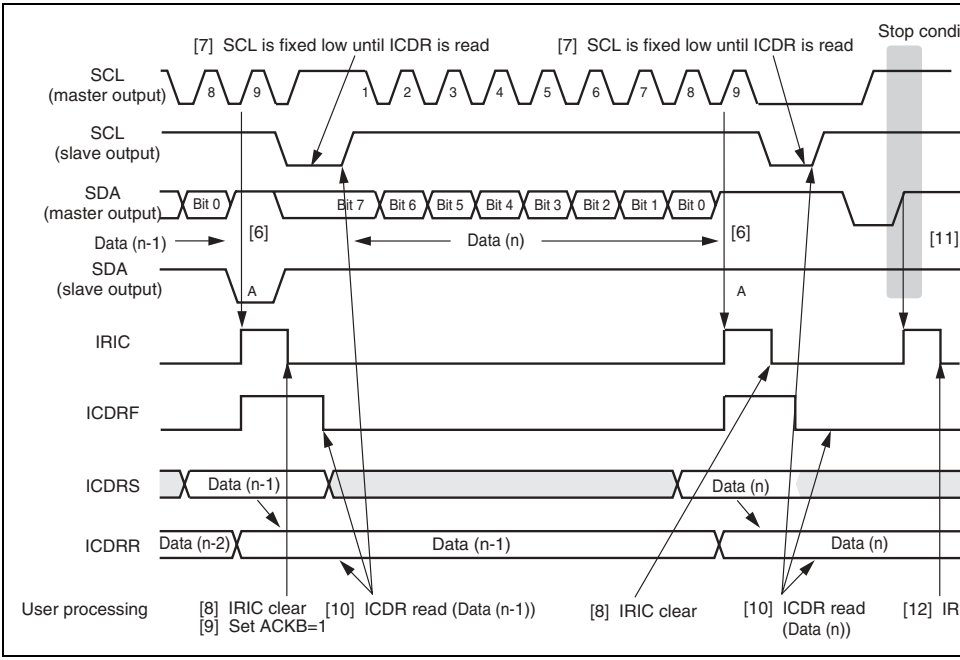


Figure 15.19 Slave Receive Mode Operation Timing Example (2)
(MLS = 0, HNDS= 1)

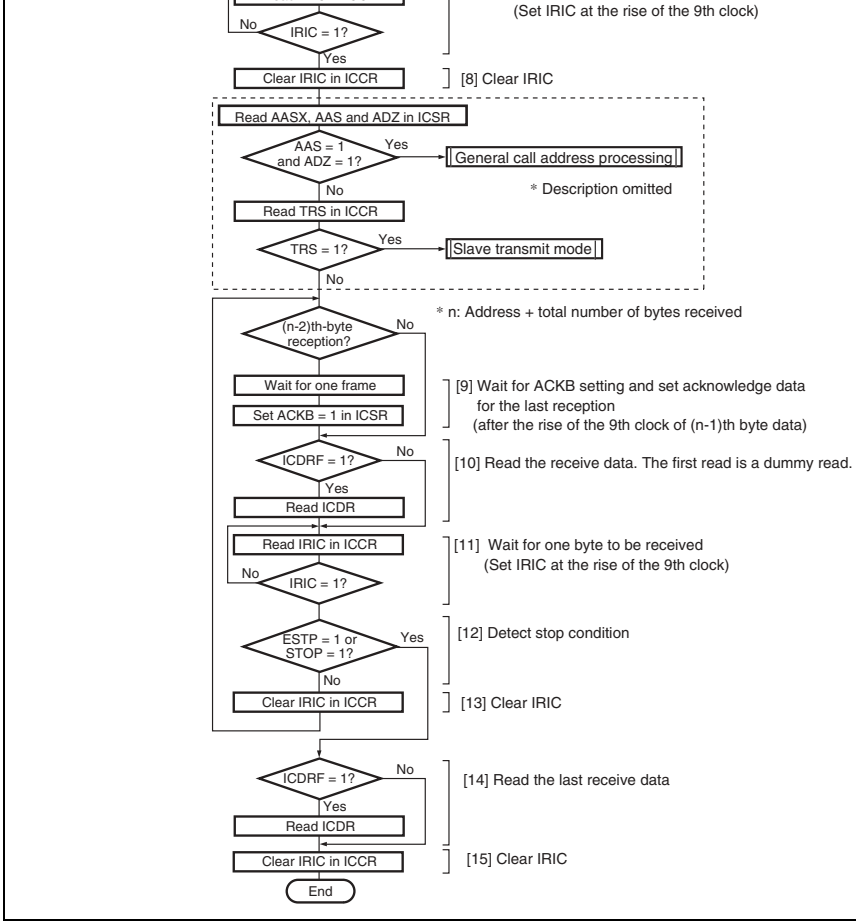


Figure 15.20 Sample Flowchart for Operations in Slave Receive Mode (HNDS

TRS bit remains cleared to 0, and slave receive operation is performed. If the 8th d
(R/$\overline{W}$) is 1, the TRS bit is set to 1, and slave transmit operation is performed. Whe
address does not match, receive operation is halted until the next start condition is

[5] At the 9th clock pulse of the receive frame, the slave device returns the data in the 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 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 IC 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 ACKB bit. Set the ACKB bit after the rise of the 9th clock pulse of the second last frame.

[10] Confirm that the ICDRF flag is set to 1 and read ICDR. This clears the ICDRF flag

[11] At the rise of the 9th clock pulse or when the receive data is transferred from IRDR 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 the BBSY flag is cleared to 0 and the STOP or ESTP flag is set to 1. If the STOPI been cleared to 0, the IRIC flag is set to 1. In this case, execute step [14] to read th receive data.

[13] Clear the IRIC flag to 0.

Receive operations can be performed continuously by repeating steps [9] to [13].

[14] Confirm that the ICDRF flag is set to 1, and read ICDR.

[15] Clear the IRIC flag.

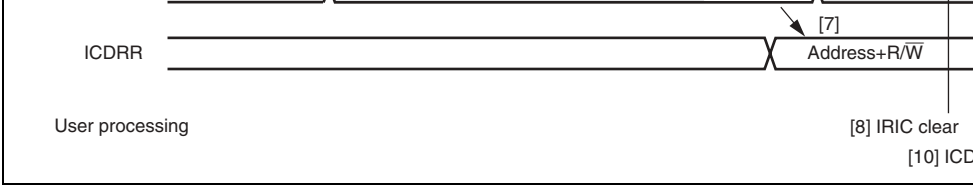


Figure 15.21 Slave Receive Mode Operation Timing Example (1)
(MLS = ACKB = 0, HNDS = 0)

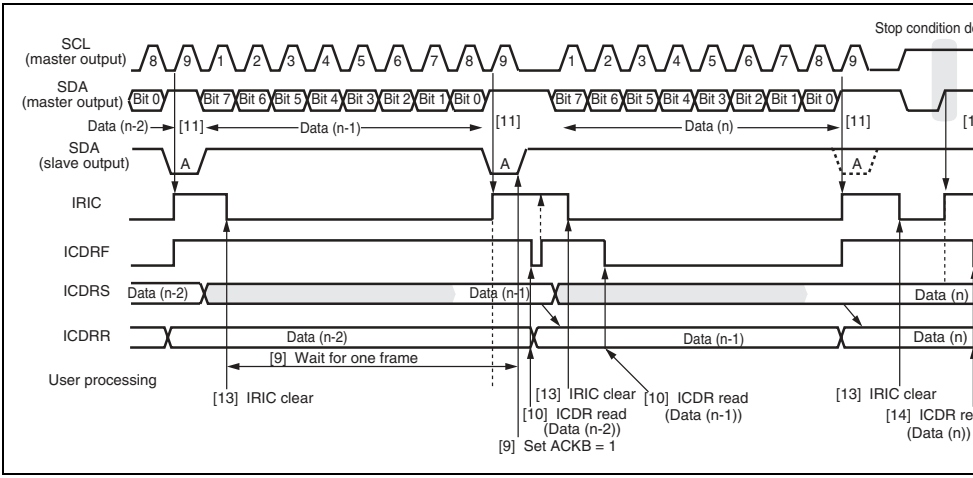


Figure 15.22 Slave Receive Mode Operation Timing Example (2)
(MLS = ACKB = 0, HNDS = 0)

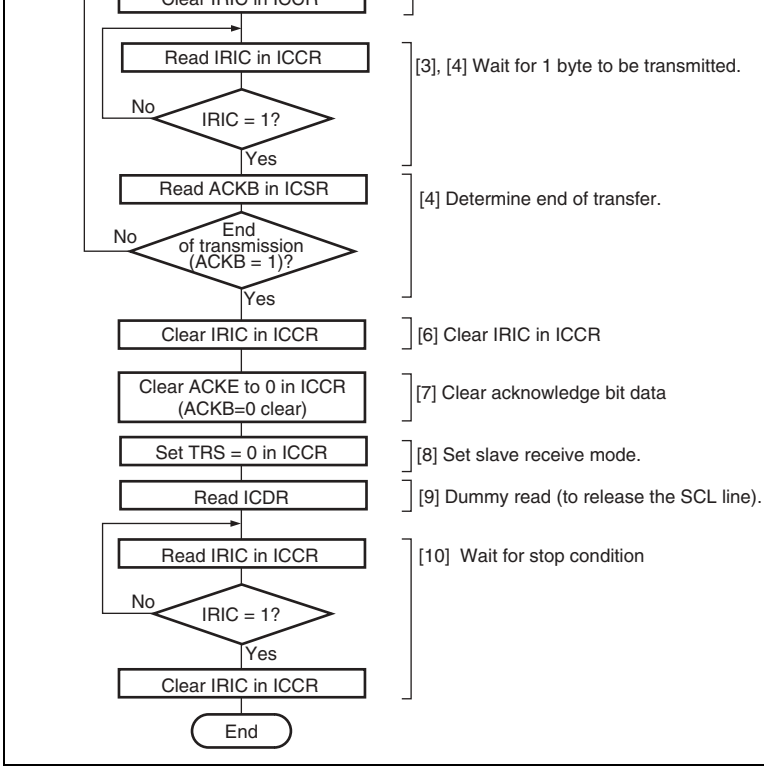


Figure 15.23 Sample Flowchart for Slave Transmit Mode

In slave transmit mode, the slave device outputs the transmit data, while the master devi the receive clock and returns an acknowledge signal. The transmission procedure and op slave transmit mode are described below.

the clock output by the master device.

The IRIC flag is cleared to 0 to detect the end of transmission. Processing from the l register writing to the IRIC flag clearing should be performed continuously. Prevent other interrupt processing from being inserted.

[4] The master device drives SDA low at the 9th clock pulse, and returns an acknowled As this acknowledge signal is stored in the ACKB bit in ICSR, this bit can be used t determine whether the transfer operation was performed successfully. When one fra data has been transmitted, the IRIC flag in ICCR is set to 1 at the rise of the 9th tran clock pulse. When the ICDRE flag is 0, the data written into ICDR is transferred to and the ICDRE and IRIC flags are set to 1 again. If the ICDRE flag has been set to slave device drives SCL low from the fall of the 9th transmit clock until data is writ ICDR.

[5] To continue transmission, write the next data to be transmitted into ICDR. The ICD 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 continuous Prevent 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 ackn 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.

[10] When the stop condition is detected, that is, when SDA is changed from low to high SCL is high, the BBSY flag in ICCR is cleared to 0 and the STOP flag in ICSR is se When the STOPIM bit in ICXR is 0, the IRIC flag is set to 1. If the IRIC flag has be is cleared to 0.

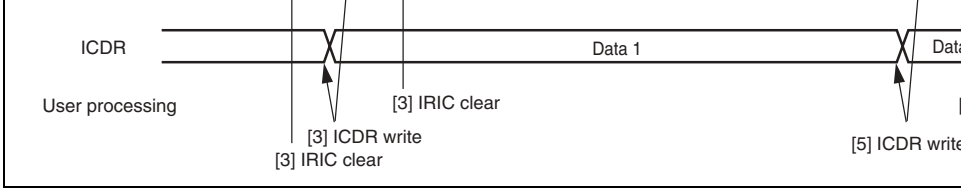
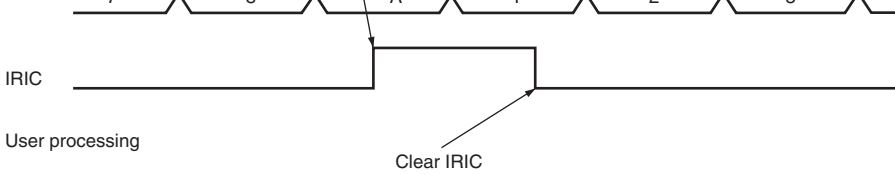
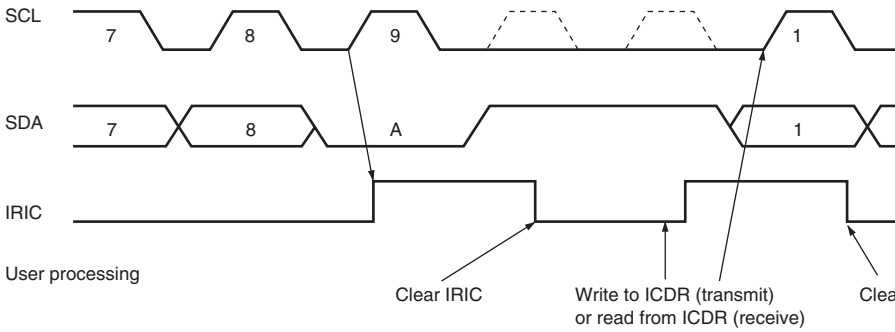


Figure 15.24 Slave Transmit Mode Operation Timing Example (MLS = 0)

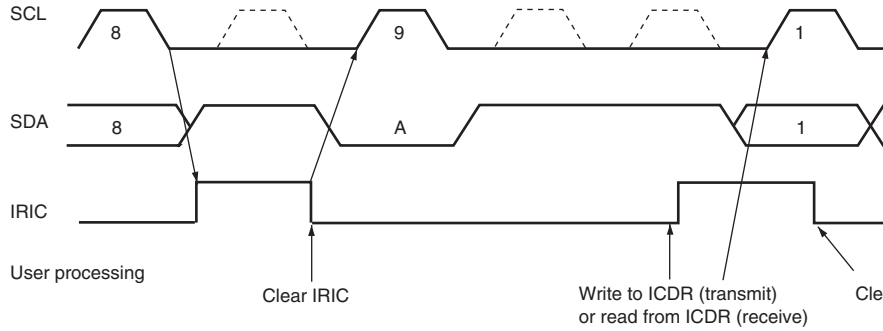


(a) Data transfer ends with ICDRE=0 at transmission, or ICDRF=0 at reception.



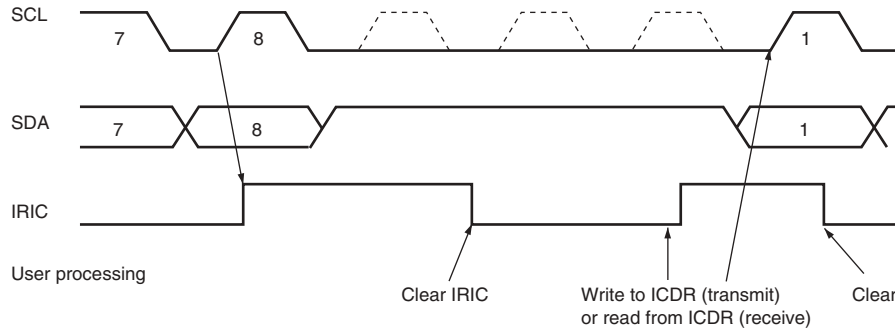
(b) Data transfer ends with ICDRE=1 at transmission, or ICDRF=1 at reception.

Figure 15.25 IRIC Setting Timing and SCL Control (1)



(b) Data transfer ends with ICDRE=1 at transmission, or ICDRF=1 at reception.

Figure 15.26 IRIC Setting Timing and SCL Control (2)



(b) Data transfer ends with ICDRE=1 at transmission, or ICDRF=1 at reception.

Figure 15.27 IRIC Setting Timing and SCL Control (3)

15.4.8 Operation Using the DTC

This LSI provides the DTC to allow continuous data transfer. IIC_4 and IIC_5 cannot use DTC. The DTC is initiated when the IRTR flag is set to 1, which is one of the two interru (IRTR and IRIC). When the ACKE bit is 0, the ICDRE, IRIC, and IRTR flags are set at t data transmission regardless of the acknowledge bit value. When the ACKE bit is 1, the I IRIC, and IRTR flags are set if data transmission is completed with the acknowledge bit v 0, and when the ACKE bit is 1, only the IRIC flag is set if data transmission is completed acknowledge bit value of 1.

When initiated, DTC transfers specified number of bytes, clears the ICDRE, IRIC, and IR to 0. Therefore, no interrupt is generated during continuous data transfer; however, if data 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.

Item	Master Transmit Mode	Master Receive Mode	Slave Transmit Mode	Slave Re Mode
Slave address + R/$\overline{W}$ bit transmission/ reception	Transmission by DTC (ICDR write)	Transmission by CPU (ICDR write)	Reception by CPU (ICDR read)	Reception (ICDR rea
Dummy data read	—	Processing by CPU (ICDR read)	—	—
Actual data transmission/ reception	Transmission by DTC (ICDR write)	Reception by DTC (ICDR read)	Transmission by DTC (ICDR write)	Reception (ICDR 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

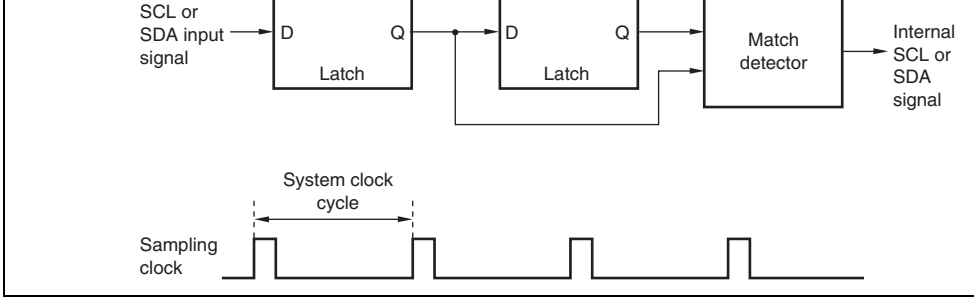


Figure 15.28 Block Diagram of Noise Canceler

15.4.10 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.)

The following items are not initialized:

- Actual register values (ICDR, SAR, SARX, ICMR, ICCR, ICSR, ICXR(other than IC
  ICDRF))
- Internal latches used to retain register read information for setting/clearing flags in the
  ICCR, and ICSR registers

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

— 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 15.11 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 15.11 I^2C Bus Timing (SCL and SDA Outputs)

Item	Symbol	Output Timing	Unit	No
SCL output cycle time	t_{SCLO}	$28t_{cyc}$ to $512t_{cyc}$	ns	Se
SCL output high pulse width	t_{SCLHO}	$0.5t_{SCLO}$	ns	25 (re
SCL output low pulse width	t_{SCLLO}	$0.5t_{SCLO}$	ns	
SDA output bus free time	t_{BUFO}	$0.5t_{SCLO} - 1t_{cyc}$	ns	
Start condition output hold time	t_{STAHO}	$0.5t_{SCLO} - 1t_{cyc}$	ns	
Retransmission start condition output setup time	t_{STASO}	$1t_{SCLO}$	ns	
Stop condition output setup time	t_{STOSO}	$0.5t_{SCLO} + 2t_{cyc}$	ns	
Data output setup time (master)	t_{SDASO}	$1t_{SCLLO} - 3t_{cyc}$	ns	
Data output setup time (slave)		$1t_{SCLLO} - (6t_{cyc}$ or $12t_{cyc}$ *)		
Data output hold time	t_{SDAHO}	$3t_{cyc}$	ns	

Note: * $6t_{cyc}$ when IICXn is 0, $12t_{cyc}$ when IICXn is 1 (n = 0 to 5).

				Time Indication [ns]						
TCSS	IICXn	t_{cyc} Indication		I^2C Bus Specification (Max.)	ϕ = 5 MHz	ϕ = 8 MHz	ϕ = 10 MHz	ϕ = 16 MHz	ϕ = 20 MHz	ϕ = 2 MHz
0	0	7.5 t_{cyc}	Standard mode	1000	1000	937	750	468	375	300
			High-speed mode	300	300	300	300	300	300	300
	1	17.5 t_{cyc}	Standard mode	1000	1000	1000	1000	1000	875	700
1	0		High-speed mode	300	300	300	300	300	300	300
1	1	37.5 t_{cyc}	Standard mode	1000	1000	1000	1000	1000	1000	1000
			High-speed mode	300	300	300	300	300	300	300

Note: n = 0 to 5

6. The I^2C bus interface specifications for the SCL and SDA rise and fall times are unde and 300 ns. The I^2C bus interface SCL and SDA output timing is prescribed by t_{cyc}, as table 15.11. However, because of the rise and fall times, the I^2C bus interface specific may not be satisfied at the maximum transfer rate. Table 15.13 shows output timing calculations for different operating frequencies, including the worst-case influence of fall times.

		High-speed mode	–250	1300	2550	1500	1150*[1]	1000*[1]	1000*[1]	1000*
t_{BUFO}	0.5 t_{SCLO} –1 t_{cyc} ($-t_{Sr}$)	Standard mode	–1000	4700	3800*[1]	3875*[1]	3900*[1]	3938*[1]	3950*[1]	3960*
		High-speed mode	–300	1300	2300	1325	1000*[1]	888*[1]	900*[1]	910*[1]
t_{STAHO}	0.5 t_{SCLO} –1 t_{cyc} ($-t_{Sf}$)	Standard mode	–250	4000	4550	4625	4650	4688	4700	4710
		High-speed mode	–250	600	2350	1375	1050	938	950	960
t_{STASO}	1 t_{SCLO} ($-t_{Sr}$)	Standard mode	–1000	4700	9000	9000	9000	9000	9000	9000
		High-speed mode	–300	600	5300	3200	2500	2200	2200	2200
t_{STOSO}	0.5 t_{SCLO} + 2 t_{cyc} ($-t_{Sr}$)	Standard mode	–1000	4000	4400	4250	4200	4125	4100	4080
		High-speed mode	–300	600	2900	1700	1300	1075	1050	1030
t_{SDASO} (master)	1 t_{SCLLO}*[3] –3 t_{cyc} ($-t_{Sr}$)	Standard mode	–1000	250	3150	3375	3450	3563	3600	3630
		High-speed mode	–300	100	1650	825	550	513	550	580
t_{SDASO} (slave)	1 t_{SCLL}*[3] –12 t_{cyc}*[2] ($-t_{Sr}$)	Standard mode	–1000	250	1300	2200	2500	2950	3100	3220
		High-speed mode	–300	100	–1400*[1]	–500*[1]	–200*[1]	250	400	520

The values in the above table will vary depending on the settings of the bits I
IICX5 to IICX0 and CKS0 to CKS2. Depending on the frequency it may not be
to achieve the maximum transfer rate; therefore, whether or not the I²C bus in
specifications are met must be determined in accordance with the actual setti
conditions.

2. Value when the IICXn bit is set to 1. When the IICXn bit is cleared to 0, the va
($-6t_{cyc}$) (n = 0 to 5).
3. Calculated using the I²C bus specification values (standard mode: 4700 ns m
speed mode: 1300 ns min.).

7. Notes on ICDR register read at end of master reception

To halt reception at the end of a receive operation in master receive mode, set the TR
and write 0 to BBSY and SCP in ICCR. This changes SDA from low to high when S
high, and generates the stop condition. After this, receive data can be read by means
ICDR read, but if data remains in the buffer the ICDRS receive data will not be trans
ICDR, and so it will not be possible to read the second byte of data.

If it is necessary to read the second byte of data, issue the stop condition in master re
mode (i.e. with the TRS bit cleared to 0). When reading the receive data, first confirm
BBSY bit in the ICCR register is cleared to 0, the stop condition has been generated,
bus has been released, then read the ICDR register with TRS cleared to 0.

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 ICC
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 15.29 (after confirming that the BBS
been cleared to 0 in the ICCR register).

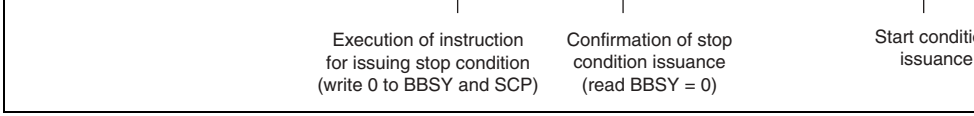


Figure 15.29 Notes on Reading Master Receive Data

Note: This restriction on usage can be canceled by setting the FNC1 and FNC0 bits to 1
ICXR.

8. Notes on start condition issuance for retransmission

   Figure 15.30 shows the timing of start condition issuance for retransmission, and the t
   subsequently writing data to ICDR, together with the corresponding flowchart. Write
   transmit data to ICDR after the start condition for retransmission is issued and then th
   condition is actually generated.

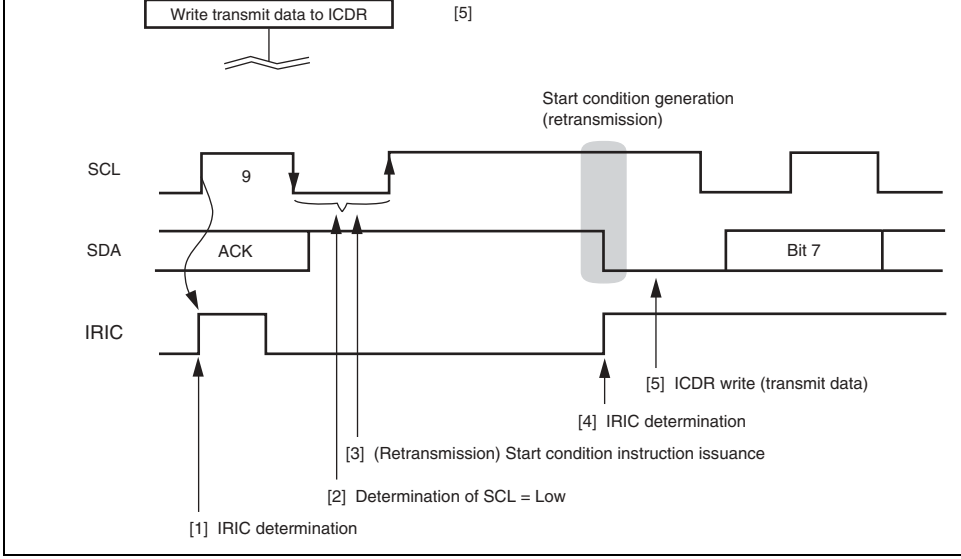


Figure 15.30 Flowchart for Start Condition Issuance Instruction for Retransmis Timing

Note: This restriction on usage can be canceled by setting the FNC1 and FNC0 bits to ICXR.

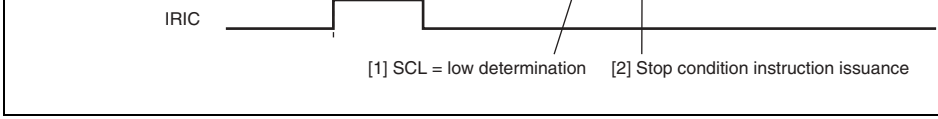


Figure 15.31 Stop Condition Issuance Timing

Note: This restriction on usage can be canceled by setting the FNC1 and FNC0 bits to B 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 wh rise time of SCL exceeds the stipulated value or where a slave device in which a wait inserted by driving the SCL pin low is used, the IRIC flag should be cleared after dete that the SCL is low.

If the IRIC flag is cleared to 0 when WAIT = 1 while the SCL is extending the high le the SDA level may change before the SCL goes low, which may generate a start or st condition erroneously.

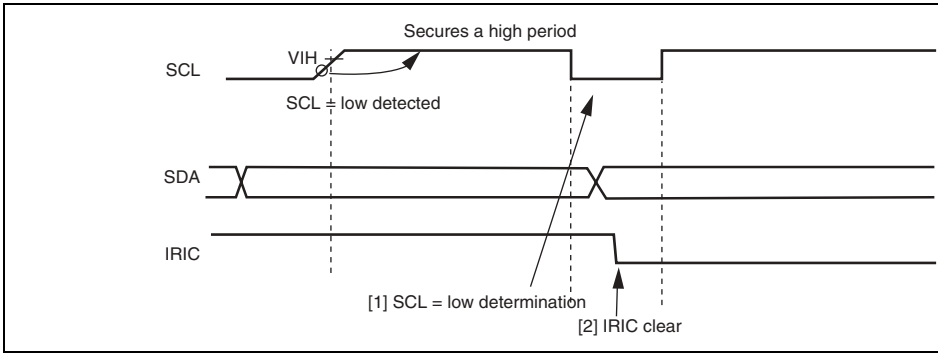


Figure 15.32 IRIC Flag Clearing Timing When WAIT = 1

— Monitor the BC2 to BC0 counter in ICMR, when the count is B'000 (8th or 9th c
pulse), wait for at least two transfer clock times in order to read ICDR or read/wr
ICCR during the time other than the shaded time.

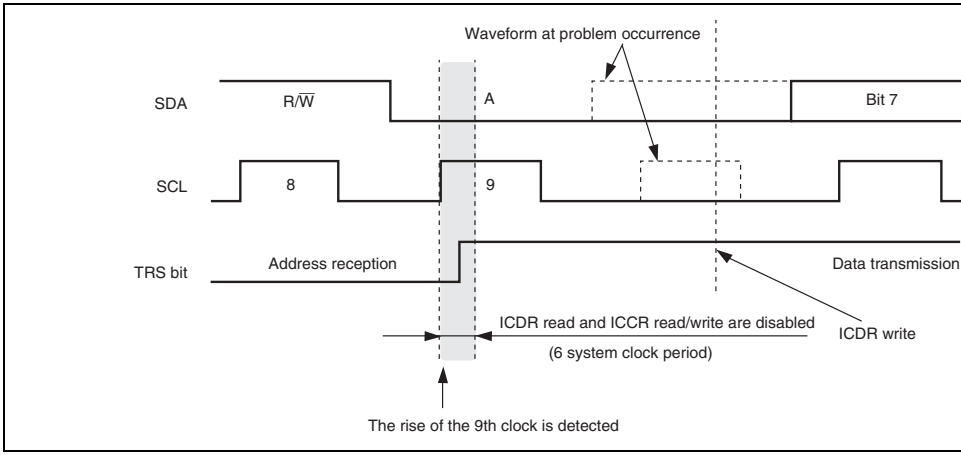


**Figure 15.33 ICDR Register Read and ICCR Register Access Timing in Slave T
Mode**

Note: This restriction on usage can be canceled by setting the FNC1 and FNC0 bits to
ICXR.

mode, clear the TRS bit to and then dummy read ICDR.

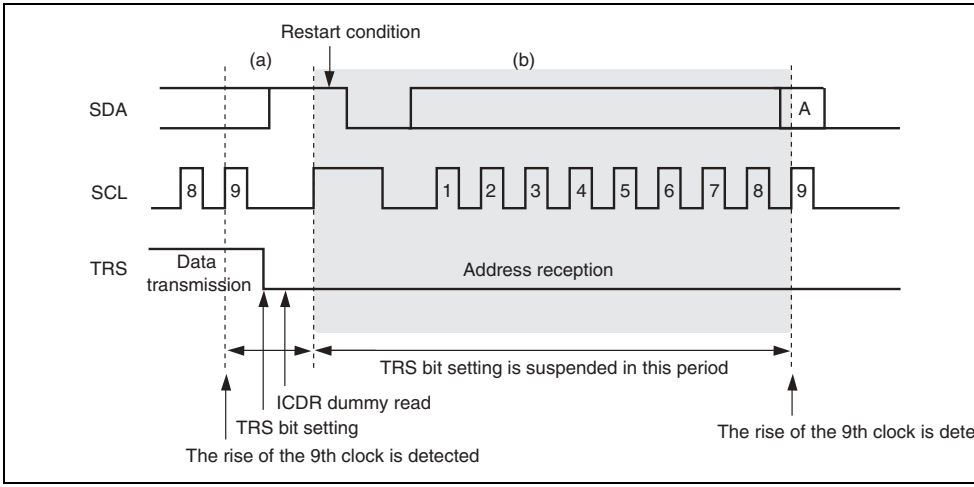


Figure 15.34 TRS Bit Set Timing in Slave Mode

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.

— Set receive mode (TRS = 0) before the next start condition is input in slave mode
Complete transmit operation by the procedure shown in figure 15.23, 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 15.35.)

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.

as a slave device.

Figure 15.35 Diagram of Erroneous Operation when Arbitration Lost

Though it is prohibited in the normal I²C 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.

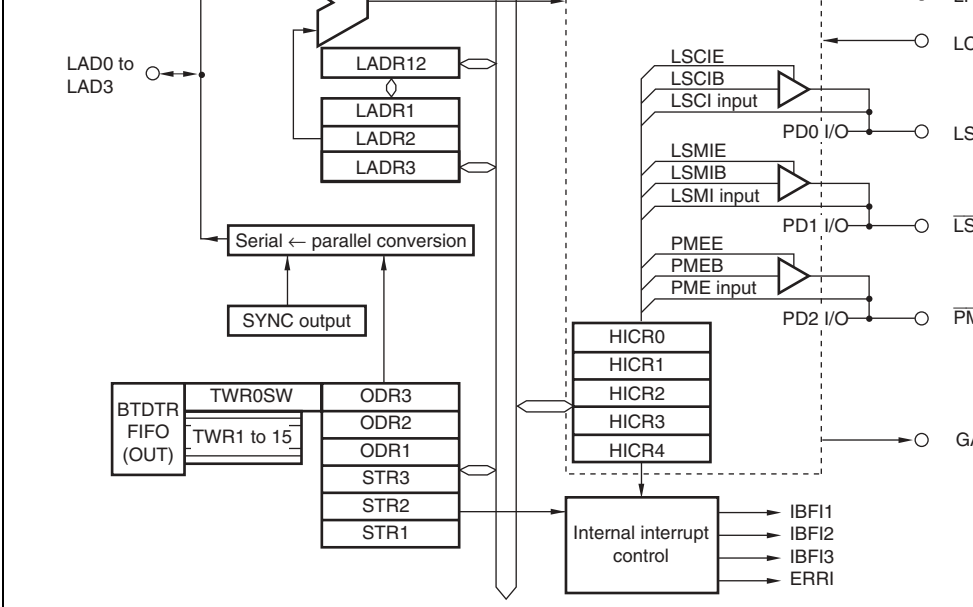
When the MST bit is set to 1 during data transmission or reception in slave mode, the arbitration decision circuit is enabled and arbitration is lost if conditions are satisfied. case, the transmit/receive data which is not an address may be erroneously recognized address.

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

- Supports LPC interface I/O read cycles and I/O write cycles
  Uses four signal lines (LAD3 to LAD0) to transfer the cycle type, address, and data.
  Uses three control signals: clock (LCLK), reset ($\overline{\text{LRESET}}$), and frame ($\overline{\text{LFRAME}}$).
- Has three register sets comprising data and status registers
  The basic register set comprises three bytes: an input register (IDR), output register (
  and status register (STR).
  Channels 1 to 3 have fixed I/O addresses of H'0000 to H'FFFF, respectively.
  A fast A20 gate function is also provided.
  Sixteen bidirectional data register bytes can be manipulated in addition to the basic r
- Supports SERIRQ
  Host interrupt requests are transferred serially on a single signal line (SERIRQ).
  On channel 1, HIRQ1 and HIRQ12 can be generated.
  On channels 2 and 3, SMI, HIRQ6, and HIRQ9 to HIRQ11 can be generated.
  Operation can be switched between quiet mode and continuous mode.
  The $\overline{\text{CLKRUN}}$ signal can be manipulated to restart the PCI clock (LCLK).
- Power-down functions, interrupts, etc.
  The LPC module can be shut down by inputting the $\overline{\text{LPCPD}}$ signal.
  Three pins, $\overline{\text{PME}}$, $\overline{\text{LSMI}}$, and LSCI, are provided for general input/output.
- Supports version 1.5 of the Intelligent Platform Management Interface (IPMI)
  Channel 3 supports the SMIC interface, KCS interface, and BT interface.



[Legend]

HICR0 to HICR4:	Host interface control register 0 to 4
LADR12H, 12L:	LPC channel 1, 2 address register 12H, 12L
LADR3H, 3L:	LPC channel 3 address register 3H, 3L
IDR1 to IDR3:	Input data register 1 to 3
ODR1 to ODR3:	Output data register 1 to 3
STR1 to STR3:	Status register 1 to 3
TWR0MW:	Bidirectional data register 0MW
TWR0SW:	Bidirectional data register 0SW
TWR1 to TWR15:	Bidirectional data registers 1 to 15
SIRQCR0 to SIRQCR2:	SERIRQ control registers 0 to 2
HISEL:	Host interface select register

Figure 16.1 Block Diagram of LPC

LPC clock	LCLK	PE6	Input	33 MHz PCI clock signal
Serialized interrupt request	SERIRQ	PE7	I/O*[1]	Serialized host interrupt re signal, synchronized with (SMI, HIRQ1, HIRQ6, HIF HIRQ12)
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]	A20 gate control signal ou
LPC clock run	$\overline{\text{CLKRUN}}$	PD4	I/O*[1, 2]	LCLK restart request sign of serial host interrupt req
LPC power-down	$\overline{\text{LPCPD}}$	PD5	Input*[1]	LPC module shutdown sig

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 goes to the high-impedance state, external resistor is necessary to pull the signal up to VCC.

- 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)
- Bidirectional data registers 0 to 15 (TWR0 to TWR15)
- Status register 1 (STR1)
- Status register 2 (STR2)
- Status register 3 (STR3)
- SERIRQ control register 0 (SIRQCR0)
- SERIRQ control register 1 (SIRQCR1)
- SERIRQ control register 2 (SIRQCR2)
- Host interface select register (HISEL)

SMIC mode:

The following registers are required when SMIC mode is used.

- SMIC flag register (SMICFLG)
- SMIC control status register (SMICCSR)
- SMIC data register (SMICDTR)
- SMIC interrupt register 0 (SMICIR0)
- SMIC interrupt register 1 (SMICIR1)

6	LPC2E	0	R/W	—	
5	LPC1E	0	R/W	—	

Enables or disables the LPC interface functi
When the host interface is enabled (at least
the three bits is set to 1), processing for data
between the slave processor and the host p
is performed using pins LAD3 to LAD0, $\overline{\text{LFR}}$
$\overline{\text{LRESET}}$, LCLK, SERIRQ, $\overline{\text{CLKRUN}}$, and $\overline{\text{LP}}$

- LPC3E

0: LPC channel 3 operation is disabled

No address (LADR3) matches for IDR3, C
STR3, TWR0 to TWR15, SMIC, KCS, or B

1: LPC channel 3 operation is enabled

- LPC2E

0: LPC channel 2 operation is disabled

No address (LADR2) matches for IDR2, C
STR2

1: LPC channel 2 operation is enabled

- LPC1E

0: LPC channel 1 operation is disabled

No address (LADR1) matches for IDR1, C
STR1

1: LPC channel 1 operation is enabled

					1: Fast A20 gate function enabled • GA20 pin output is open-drain (externa pull-up resistor required)
3	SDWNE	0	R/W	—	LPC Software Shutdown Enable Controls LPC interface shutdown. For deta LPC shutdown function, and the scope of initialization by an LPC reset and an LPC s see section 16.4.6, LPC Interface Shutdow Function (LPCPD). 0: Normal state, LPC software shutdown s enabled [Clearing conditions] • Writing 0 • LPC hardware reset or LPC software re • LPC hardware shutdown release (rising $\overline{\text{LPCPD}}$ signal) 1: LPC hardware shutdown state setting en • Hardware shutdown state when $\overline{\text{LPCPD}}$ low [Setting condition] • Writing 1 after reading SDWNE = 0

					1 1 : PME output enabled, PME pin output is h impedance
1	LSMIE	0	R/W	—	LSMI Output Enable Controls LSMI output in combination with the LSMIB HICR1. $\overline{\text{LSMI}}$ pin output is open-drain, and an externa resistor is needed to pull the output up to VCC When the LSMI output function is used, the DDR bit f must not be set to 1. LSMIE LSMIB 0 * : LSMI output disabled, other function enabled 1 0 : LSMI output enabled, $\overline{\text{LSMI}}$ pin outpu level 1 1 : LSMI output enabled, $\overline{\text{LSMI}}$ pin outpu impedance
0	LSCIE	0	R/W	—	LSCI Output Enable Controls LSCI output in combination with the LSCIB b HICR1. LSCI pin output is open-drain, and an externa resistor is needed to pull the output up to VCC When the LSCI output function is used, the DDR bit f must not be set to 1. LSCIE LSCIB 0 * : LSCI output disabled, other function enabled 1 0 : LSCI output enabled, LSCI pin outpu level 1 1 : LSCI output enabled, LSCI pin outpu impedance

Note: * Don't care

					• LPC hardware reset or LPC software re • LPC hardware shutdown or LPC softwa shutdown • Forced termination (abort) of transfer cy subject to processing • Normal termination of transfer cycle sul processing 1: LPC interface is performing transfer cycl processing [Setting condition] • Match of cycle type and address
6	CLKREQ	0	R	—	LCLK Request Indicates that the LPC interface's SERIRQ requesting a restart of LCLK. 0: No LCLK restart request [Clearing conditions] • LPC hardware reset or LPC software re • LPC hardware shutdown or LPC softwa shutdown • SERIRQ is set to continuous mode • There are no further interrupts for trans host in quiet mode 1: LCLK restart request issued [Setting condition] • In quiet mode, SERIRQ interrupt outpu necessary while LCLK is stopped

					1: SERIRQ transfer processing in progress [Setting condition] • Start of SERIRQ transfer frame
4	LRSTB	0	R/W	—	LPC Software Reset Bit Resets the LPC interface. For the scope of initialization by an LPC reset, see section 16 LPC Interface Shutdown Function (LPCPD). 0: Normal state [Clearing conditions] • Writing 0 • LPC hardware reset 1: LPC software reset state [Setting condition] • Writing 1 after reading LRSTB = 0

					• LPC hardware shutdown (falling edge of $\overline{\text{LPCPD}}$ signal when SD • LPC hardware shutdown release (rising edge of $\overline{\text{LPCPD}}$ signal when SD 1: LPC software shutdown state [Setting condition] • Writing 1 after reading SDWNB = 0
2	PMEB	0	R/W	—	PME Output Bit Controls PME output by the combination w PMEE bit. For details, see the PMEE bit in HICR0.
1	LSMIB	0	R/W	—	LSMI Output Bit Controls LSMI output by the combination w LSMIE bit. For details, see the LSMIE bit in HICR0.
0	LSCIB	0	R/W	—	LSCI Output Bit Controls LSCI output by the combination w LSCIE bit. For details, see the LSCIE bit in HICR0.

Bit	Bit Name	Initial Value	Slave	Host	Description
7	GA20	Undefined	R	—	GA20 Pin Monitor
6	LRST	0	R/(W)*	—	LPC Reset Interrupt Flag Interrupt flag that generates an ERRI interru 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 Interrupt flag that generates an ERRI interru an LPC hardware shutdown request is gene 0: [Clearing conditions] • Writing 0 after reading SDWN = 1 • LPC hardware reset • LPC software reset 1: [Setting condition] • $\overline{\text{LPCPD}}$ pin falling edge detection

					1: [Setting condition] • $\overline{\text{LFRAME}}$ pin falling edge detection duri transfer cycle
3	IBFIE3	0	R/W	—	IBFI3 Interrupt Enable Enables or disables IBFI3 interrupt to the s processor (this LSI). 0: Input data register IDR3 and TWR recei completed interrupt requests and SMIC BT mode interrupt requests are disabled 1: [When TWRE in LADR3 = 0] Input data register IDR3 receive comple interrupt request and SMIC mode and B interrupt requests are enabled [When TWRE in LADR3 = 1] Input data register IDR3 and TWR recei completed interrupt requests and SMIC BT mode interrupt requests are enabled
2	IBFIE2	0	R/W	—	IDR2 Receive Completion Interrupt Enable Enables or disables IBFI2 interrupt to the s processor (this LSI). 0: Input data register IDR2 receive complet interrupt requests disabled 1: Input data register IDR2 receive complet interrupt requests enabled

					0: Error interrupt requests disabled
					1: Error interrupt requests enabled

Note: * Only 0 can be written to clear bits 6 to 4.

- 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

					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 cycle from slave, thus can make the hos 0: Short wait is issued 1: Long wait is issued
2	KCSENBL	0	R/W	—	Enables or disables the use of the KCS included in channel 3. When the LPC3E HICR0 is 0, this bit is valid. 0: KCS interface operation is disabled No address (LADR3) matches for IDF or STR3 in KCS mode 1: KCS interface operation is enabled
1	SMICENBL	0	R/W	—	Enables or disables the use of the SMIC included in channel 3. When the LPC3E HICR0 is 0, this bit is valid. 0: SMIC interface operation is disabled No address (LADR3) matches for SM SSMICCSR, or SMICDTR 1: SMIC interface operation is enabled
0	BTENBL	0	R/W	—	Enables or disables the use of the BT in included in channel 3. When the LPC3E 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

3	Bit 11				
2	Bit 10				
1	Bit 9				
0	Bit 8				

- LADR3L

			R/W		
Bit	**Bit Name**	**Initial Value**	**Slave**	**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 regi operation.
					Clear this bit to 0 in KCS mode.
					0: TWR operation is disabled
					TWR-related address (LADR3) match d occur.
					1: TWR operation is enabled

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		

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 Selec**
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

1	0	0	1	0	1	TWR/BT mode	LADR3+2/+3	TWR fla
1	0	0	1	1	0	TWR/SMIC mode	LADR3+2/+3	TWR fla
1	0	0	1	1	1	TWR/SMIC/BT mode	LADR3+2/+3	TWR fla
1	0	1	0	0	0	KCS mode	LADR3+2/+3	TWR fla
1	0	1	0	0	1	KCS/BT mode	LADR3+2/+3	TWR fla
1	0	1	0	1	0	KCS/SMIC mode	LADR3+2/+3	TWR fla
1	0	1	0	1	1	KCS/SMIC/BT mode	LADR3+2/+3	TWR fla
1	1	0	0	0	0	Normal mode	LADR3+0/+4	User de
1	1	0	0	0	1	BT mode	Access disabled	Access
1	1	0	0	1	0	SMIC mode	Access disabled	Access
1	1	0	0	1	1	SMIC/BT mode	Access disabled	Access
1	1	0	1	0	0	TWR mode	LADR3+0/+4	TWR fla
1	1	0	1	0	1	TWR/BT mode	LADR3+2/+3	TWR fla
1	1	0	1	1	0	TWR/SMIC mode	LADR3+2/+3	TWR fla
1	1	0	1	1	1	TWR/SMIC/BT mode	LADR3+2/+3	TWR fla
1	1	1	0	0	0	KCS mode	LADR3+2/+3	User de
1	1	1	0	0	1	KCS/BT mode	LADR3+2/+3	User de
1	1	1	0	1	0	KCS/SMIC mode	LADR3+2/+3	User de
1	1	1	0	1	1	KCS/SMIC/BT mode	LADR3+2/+3	User de
1	—*	1	1	—*	—*	Setting prohibited	Setting prohibited	Setting

Note: * Don't care

Table 16.2 shows the initial value of each register. Table 16.3 shows the host register sele
address match determination. Table 16.4 shows the slave selection internal registers in sla
LSI) access.

Table 16.2 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

Table 16.3 Host Register Selection

I/O Address				Transfer	
Bits 15 to 3	Bit 2	Bit 1	Bit 0	Cycle	Host Register S
LADR1 (bits 15 to 3)	0	LADR1 (bit 1)	LADR1 (bit 0)	I/O write	IDR1 write (data) C/$\overline{D}$1 ← 0
LADR1 (bits 15 to 3)	1	LADR1 (bit 1)	LADR1 (bit 0)	I/O write	IDR1 write (comm C/$\overline{D}$1 ← 1
LADR1 (bits 15 to 3)	0	LADR1 (bit 1)	LADR1 (bit 0)	I/O read	ORD1 read
LADR1 (bits 15 to 3)	1	LADR1 (bit 1)	LADR1 (bit 0)	I/O read	STR1 read
LADR2 (bits 15 to 3)	0	LADR2 (bit 1)	LADR2 (bit 0)	I/O write	IDR2 write (data) C/$\overline{D}$2 ← 0
LADR2 (bits 15 to 3)	1	LADR2 (bit 1)	LADR2 (bit 0)	I/O write	IDR2 write (comm C/$\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

The IDR registers are 8-bit read-only registers to the slave processor (this LSI), and 8-bi
only registers to the host processor. The registers selected from the host according to the
address are described in the following sections: for information on IDR1 and IDR2 selec
section 16.3.5, LPC Channel 1, 2 Address Register H, L (LADR12H, LADR12L), and f
information on IDR3 selection, see section 16.3.4, LPC Channel 3 Address Register H,
(LADR3H, LADR3L). Data transferred in an LPC I/O write cycle is written to the selec
register. The state of bit 2 of the I/O address is latched into the C/$\overline{D}$ bit in STR, to indica
the written information is a command or data.

The initial values of the IDR registers are undefined.

16.3.7 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 16.3.5, LPC Channel 1, 2 Address Register H, L (LADR12H, LADR12L), and f
information on ODR3 selection, see section 16.3.4, 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.

cycle, the data in the selected register is transferred to the host.

The initial values of TWR0 to TWR15 are undefined.

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

Bit	Bit Name	Initial Value	R/W Slave	R/W 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 is written into this b indicate whether IDR1 contains data or a c
					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.

					[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.

					1: Content of input data register (IDR2) is a d
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 dat This bit is an internal interrupt source to the processor (this LSI). 0: There is not receive data in IDR2 [Clearing condition] When the slave processor reads IDR2 1: There is receive data in IDR2 [Setting condition] When the host processor writes to IDR2 usi write cycle
0	OBF2	0	R/(W)*	R	Output Data Register Full Indicates whether or not there is transmit da ODR2. 0: There is not transmit data in ODR2 [Clearing condition] When the host processor reads ODR2 using cycle, or the slave processor writes 0 to the 1: There is transmit data in ODR2 [Setting condition] When the slave processor writes to ODR2

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

					[Setting condition] When the host processor writes to TWR15 u write cycle
6	OBF3B	0	R/(W)*	R	Bidirectional Data Register Output Data Full Indicates whether or not there is transmit da TWR0 to TWR15. 0: There is not transmit data in TWR15 [Clearing condition] When the host processor reads TWR15 usir read cycle, or the slave processor writes 0 t OBF3B bit 1: There is transmit data in TWR0 to TWR1 [Setting condition] When the slave processor writes to TWR15
5	MWMF	0	R	R	Master Write Mode Flag Indicates that master write mode is entered writing to TWR0 from the host processor. 0: [Clearing condition] When the slave processor reads TWR15 1: [Setting condition] When the host processor writes to TWR0 us write cycle while SWMF = 0

					When the slave processor writes to TW MWMF = 0
3	C/$\overline{D}$3	0	R	R	Command/Data When the host processor writes to an IDR3 bit 2 of the I/O address is written into this b indicate whether IDR3 contains data or a c 0: Content of input data register (IDR3) is d 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 is an internal interrupt source to processor (this LSI). 0: There is not receive data in IDR3 [Clearing condition] When the slave processor reads IDR3 1: There is receive data in IDR3 [Setting condition] When the host processor writes to IDR3 us write cycle

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

- STR3

  (When TWRE = 0 and SELSTR3 = 1)

			R/W		
Bit	**Bit Name**	**Initial Value**	**Slave**	**Host**	**Description**
7 6 5 4	DBU37 DBU36 DBU35 DBU34	All 0	R/W	R	Defined by User The user can use these bits as necessary.
3	C/$\overline{D}$3	0	R	R	Command/Data When the host processor writes to an IDR3 r bit 2 of the I/O address is written into this bit indicate whether IDR3 contains data or a cor 0: Content of input data register (IDR3) is da 1: Content of input data register (IDR3) is a c
2	DBU32	0	R/W	R	Defined by User The user can use this bit as necessary.

					write cycle
0	OBF3A	0	R/(W)*	R	Output Data Register Full Indicates whether or not there is transmit d ODR3. 0: There is not receive data in ODR3 [Clearing condition] When the host processor reads ODR3 usir read cycle, or the slave processor writes 0 OBF3A bit 1: There is receive data in ODR3 [Setting condition] When the slave processor writes to ODR3

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

					0: Continuous mode [Clearing conditions] • LPC hardware reset, LPC software reset • Specification by the stop frame of the SE transfer cycle 1: Quiet mode [Setting condition] • Specification by the stop frame of the SE transfer cycle
6	SELREQ	0	R/W	—	Start Frame Initiation Request Select Specifies the condition of start frame activat the host interrupt request is cleared in quiet 0: When all host interrupt requests are clear quiet mode, start frame initiation is reque 1: When at least one host interrupt request i cleared in quiet mode, start frame initiatio requested
5	IEDIR	0	R/W	—	Interrupt Enable Direct Mode Specifies whether LPC channel 2 SERIRQ i source (SMI, HIRQ6, HIRQ9 to HIRQ11) ge is conditional upon OBF, or is controlled onl host interrupt enable bit. 0: Host interrupt is requested when host inte enable bit and corresponding OBF are bo 1 1: Host interrupt is requested when host inte enable bit is set to 1

					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 a host SMI interrupt re when 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 res • Clearing OBF3A 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 SMIE3A = 0

					Host SMI interrupt request by setting OB enabled [When IEDIR = 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 a HIRQ12 interrupt requ when OBF1 is set by an ODR1 write. 0: HIRQ12 interrupt request by OBF1 and IF is disabled [Clearing conditions] • Writing 0 to IRQ12E1 • LPC hardware reset, LPC software reset • Clearing OBF1 to 0 1: HIRQ12 interrupt request by setting OBF enabled [Setting condition] • Writing 1 after reading IRQ12E1 = 0

enabled

[Setting condition]

- Writing 1 after reading IRQ1E1 = 0

16.3.11 SERIRQ Control Register 1 (SIRQCR1)

The SIRQCR1 register contains status bits that enable or disable an SERIRQ interrupt re

The SIRQCR1 register is initialized to H'00 by a reset or in hardware standby mode.

					HIRQ11 interrupt request by setting OBF 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 a HIRQ10 interrupt requ when OBF3A is set by an ODR3 write. 0: HIRQ10 interrupt request by OBF3A and IRQ10E3 is disabled [Clearing conditions] • Writing 0 to IRQ10E3 • LPC hardware reset, LPC software reset • Clearing OB3FA to 0 (when IEDIR3 = 0) 1: [When IEDIR3 = 0] HIRQ10 interrupt request by setting OBF is enabled [When IEDIR3 = 1] HIRQ10 interrupt is requested. [Setting condition] • Writing 1 after reading IRQ10E3 = 0

					HIRQ9 interrupt request by setting OBF 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 a HIRQ6 interrupt requ 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 res • Clearing OBF3A to 0 (when IEDIR3 = 0 1: [When IEDIR3 = 0] HIRQ6 interrupt request by setting OBF enabled [When IEDIR3 = 1] HIRQ6 interrupt is requested. [Setting condition] • Writing 1 after reading IRQ6E3 = 0

					HIRQ11 interrupt request by setting OBF2 enabled [When IEDIR = 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 a HIRQ10 interrupt requ when OBF2 is set by an ODR2 write. 0: HIRQ10 interrupt request by OBF2 and IR is disabled [Clearing conditions] • Writing 0 to IRQ10E2 • LPC hardware reset, LPC software reset • Clearing OBF2 to 0 (when IEDIR = 0) 1: [When IEDIR = 0] HIRQ10 interrupt request by setting OBF2 enabled [When IEDIR = 1] HIRQ10 interrupt is requested. [Setting condition] • Writing 1 after reading IRQ10E2 = 0

					HIRQ9 interrupt request by setting OBF2 enabled [When IEDIR = 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 a HIRQ6 interrupt requ OBF2 is set by an ODR2 write. 0: HIRQ6 interrupt request by OBF2 and IR disabled [Clearing conditions] • Writing 0 to IRQ6E2 • LPC hardware reset, LPC software res • Clearing OBF2 to 0 (when IEDIR = 0) 1: [When IEDIR = 0] HIRQ6 interrupt request by setting OBF2 enabled [When IEDIR = 1] HIRQ6 interrupt is requested. [Setting condition] • Writing 1 after reading IRQ6E2 = 0

				0: The host interrupt is requested when both host interrupt enable bit and correspondi are set to 1 1: The host interrupt is requested when the interrupt enable bit is set to 1
6 to 0	—	All 0	R/W	—	Reserved The initial value should not be changed.

					0: Bits 7 to 4 in STR3 is the LPC interface 1: [When TWRE = 0] Bits 7 to 4 in STR3 are defined by user. [When TWRE = 1] Bits 7 to 4 in STR3 are the LPC interfac bits.
6 5 4 3 2 1 0	SELIRQ11 SELIRQ10 SELIRQ9 SELIRQ6 SELSMI SELIRQ12 SELIRQ1	All 0	R/W	—	Selects the SERIRQ Output These bits select the output status for LPC interrupt request (HIRQ11, HIRQ10, HIRQ SMI, HIRQ12, and HIRQ1). 0: [When host interrupt request has been c The SERIRQ output is high impedance. [When host interrupt request has been s The SERIRQ output is 0 level. 1: [When host interrupt request has been c The SERIRQ output is 0 level. [When host interrupt request has been s The SERIRQ output is high impedance.

					1: Slave is ready for the host read transfer
6	TX_DATA_RDY	0	R/W	R	Write Transfer Ready Indicates whether or not the slave is ready f host next write transfer. 0: The slave waits for ready status 1: The slave is ready for the host write trans
5	—	0	R/W	R	Reserved The initial value should not be changed.
4	SMI	0	R/W	R	SMI Flag This bit indicates that the SMI is asserted. 0: Indicates waiting for SMI assertion 1: Indicates SMI assertion

					1: There is a message
1	—	0	R/W	R	Reserved The initial value should not be changed.
0	BUSY	0	R/(W)*	W	SMIC Busy This bit indicates that the slave is now tran data. This bit can be cleared only by the sl set only by the host. The rising edge of this bit is a source of int interrupt to the slave. 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.

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.

16.3.17 SMIC Interrupt Register 0 (SMICIR0)

SMICIR0 is one of the registers used to implement SMIC mode. This register includes th indicate the source of interrupt to the slave.

				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 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.
2	STARI	0	R/(W)* —	Status Code Receive End Interrupt This is a status flag that indicates that the finished receiving the status code from SM When the IBFIE3 bit and STARIE bit are s IBFI3 interrupt is requested to the slave. 0: Status code receive wait state [Clearing condition] After the slave reads STARI = 1, writes 0 t 1: Status code receive end [Setting condition] When the host reads the status code of SM

				When the host writes the status code to SM
0	BUSYI	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.

					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

				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 un 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 I 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.

				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
0	HBTRI	0	R/(W)* —	BTDTR Host Read End Interrupt This status flag indicates that the host read data from BTDTR buffer. When the BFIE3 HBTRIE bit are set to 1, IBFI3 interrupt is r to the slave. 0: BTDTR host read end wait state [Clearing condition] After the slave reads HBTRI = 1 and writes bit. 1: BTDTR host read end [Setting condition] When the host finished reading the valid da BTDTR.

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

					bit and HRSTIE bit are set to 1, IBFI3 interru requested to the slave. 0: [Clearing condition] When the slave reads HRSTI = 1 and wri this bit. 1: [Setting condition] When the slave detects the rising edge o 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 IE and IRQCRIE bit are set to 1, IBFI3 interrup requested to the slave. 0: [Clearing condition] When the slave reads IRQCRI = 1 and w this bit. 1: [Setting condition] When the slave detects the falling edge o B2H_IRQ.

				BEVT_ATN.
3	B2HI	0	R/(W)*	— Read End Interrupt This status flag indicates that the host has reading all data from the BTDTR buffer. W IBFIE3 bit and B2HIE bit are set to 1, the I interrupt is requested to the slave. 0: [Clearing condition] When the slave reads B2HI = 1 and writ this bit. 1: [Setting conditions] ATNSW = 0: When the slave detects th edge of B2H_ATN. ATNSW = 1: When the slave detects th edge of H_BUSY.
2	H2BI	0	R/(W)*	— Write End Interrupt This status flag indicates that the host has writing all data to the BTDTR buffer. When IBFIE3 bit and H2BIE bit are set to 1, the I interrupt is requested to the slave. 0: [Clearing condition] After the slave reads H2BI = 1, writes 0 1: [Setting condition] When the slave detects the falling edge H2B_ATN.

				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 IB and CRWPIE bit are set to 1, the IBFI3 inter requested to the slave. 0: [Clearing condition] After the slave reads CRWPI = 1, writes 0 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.

					0 * :FIFO disabled 1 * :FIFO enabled The FIFO size: 64 bytes (for host write trans additional 64 bytes (for host read transfer).
4	FRDIE	0	R/W	—	FIFO Read Request Interrupt Enable Enables or disables the FRDI interrupt whic 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 whic IBFI3 interrupt source to the slave. When using FIFO, the HRDIE bit must not b 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 whic IBFI3 interrupt source to the slave. When using FIFO, the HWRIE bit must not b 1. 0: BT host write interrupt is disabled. 1: BT host write interrupt is enabled.

Note: * Don't care.

16.3.22 BT Control Status Register 1 (BTCSR1)

BTCSR1 is one of the registers used to implement the BT mode. The BTCSR1 register co 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.

			R/W		
Bit	Bit Name	Initial Value	Slave	Host	Description
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.
5	IRQCRIE	0	R/W	—	B2H_IRQ Clear Interrupt Enable Enables or disables the IRQCRI interrupt w an IBFI3 interrupt source to the slave. 0: B2H_IRQ clear interrupt is disabled. 1: B2H_IRQ clear interrupt is enabled.

					1: Read end interrupt is enabled.
2	H2BIE	0	R/W	—	Write End Interrupt Enable Enables or disables the H2BI interrupt whi 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 w 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 w IBFI3 interrupt source to the slave. 0: Write pointer clear interrupt is disabled. 1: Write pointer clear interrupt is enabled.

					1: Indicates that the BTDTR buffer is being
6	H_BUSY	0	R	(W)*3	BT Read Transfer Busy Flag This is a set/clear bit from the host. Indicat the BTDTR buffer is being used for BT rea transfer (read transfer is in progress.) 0: Indicates waiting for BT read transfer [Clearing condition] When the host writes a 1 while H_BUSY is 1. 1: Indicates that the BTDTR buffer is being [Setting condition] When the host writes a 1 while H_BUSY is 0.
5	OEM0	0	R/W	R/(W)*4	User defined bit This bit is defined by the user, and validate when set to 1 by a 0 written from the host. 0: [Clearing condition] When the slave writes a 0 after a 1 has read from OEM0. 1: [Setting condition] When the slave writes a 1, after a 0 has read from OEM0, or when the host write

					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 writing all data to the BTDTR buffer. Setti B2H_IRQ_EN bit in the BTIMSR register the B2H_ATN bit to be used as an interru 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 [Setting condition] When the slave writes a 1 after a 0 has b 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 ha 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 b from H2B_ATN. 1: Host has completed writing to the BTD [Setting condition] When the host writes a 1

				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 w pointer during write transfer. A host read operation always yields 0 on readout. 0: Write pointer clear wait [Clearing condition] When the slave writes a 0 after a 1 has b 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.

writing.

					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 5	— —	0 0	R/W R/W	R/W R/W	Reserved
4 3 2	OEM3 OEM2 OEM1	0 0 0	R/W R/W R/W	R/(W)*4 R/(W)*4 R/(W)*4	User defined bit These bits are defined by the user and a only when set to 1 by a 0 written from th 0: [Clearing condition] When the slave writes a 0, after a 1 h read from OEM. 1: [Setting condition] When the slave writes a 1, after a 0 h read from OEM, or when the host writ

					[Setting condition] When the slave writes a 1, after a 0 has read 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 the interrupt source from the slave host. 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.

					decremented by only the number of bytes th been read.

16.3.27 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 ca for host read transfer. When data is written f slave, the value in BTFVSR1 is incremented number of bytes that have been written to. F when data is read from the host, the value is decremented by only the number of bytes th been read.

1. Read the signal line status and confirm that the LPC module can be connected. Also the LPC module is initialized internally.
2. Set the I/O addresses of the channels to be used (LADR1 to LADR3) and whether or bidirectional registers, KCS interface, SMIC interface, and BT interface are to be use
3. Set the enable bit (LPC3E to LPC1E) for the channel to be used.
4. Set the enable bits (FGA20E, PMEE, LSMIE, and LSCIE) for the additional functio used.
5. Set the selection bits for other functions (SDWNE, IEDIR).
6. As a precaution, clear the interrupt flags (LRST, SDWN, ABRT, and OBF). Read ID TWR15 to clear IBF.
7. Set interrupt enable bits (IBFIE3 to IBFIE1, ERRIE) as necessary.

16.4.2 LPC I/O Cycles

There are ten kinds of LPC transfer cycle: memory read, memory write, I/O read, I/O wr read, DMA write, bus mastership memory read, bus mastership memory write, bus mast read, and bus mastership I/O write. Of these, the chip's LPC supports only I/O read and cycles.

An LPC transfer cycle is started when the $\overline{\text{LFRAME}}$ signal goes low in the bus idle state $\overline{\text{LFRAME}}$ signal goes low when the bus is not idle, this means that a forced termination the LPC transfer cycle has been requested.

In an I/O read cycle or I/O write cycle, transfer is carried out using LAD3 to LAD0 in th following order, in synchronization with LCLK. The host can be made to wait by sendin value other than B'0000 in the slave's synchronization return cycle. However, the LPC in always returns a value of B'0000 if the BT interface is not used.

			12			12
4	Address 2	Host	Bits 11 to 8	Address 2	Host	B
5	Address 3	Host	Bits 7 to 4	Address 3	Host	B
6	Address 4	Host	Bits 3 to 0	Address 4	Host	B
7	Turnaround (recovery)	Host	B'1111	Data 1	Host	B
8	Turnaround	None	B'ZZZZ	Data 2	Host	B
9	Synchronization	Slave	B'0000	Turnaround (recovery)	Host	B'
10	Data 1	Slave	Bits 3 to 0	Turnaround	None	B'
11	Data 2	Slave	Bits 7 to 4	Synchronization	Slave	B'
12	Turnaround (recovery)	Slave	B'1111	Turnaround (recovery)	Slave	B'
13	Turnaround	None	B'ZZZZ	Turnaround	None	B'

Figure 16.2 Typical $\overline{\text{LFRAME}}$ Timing

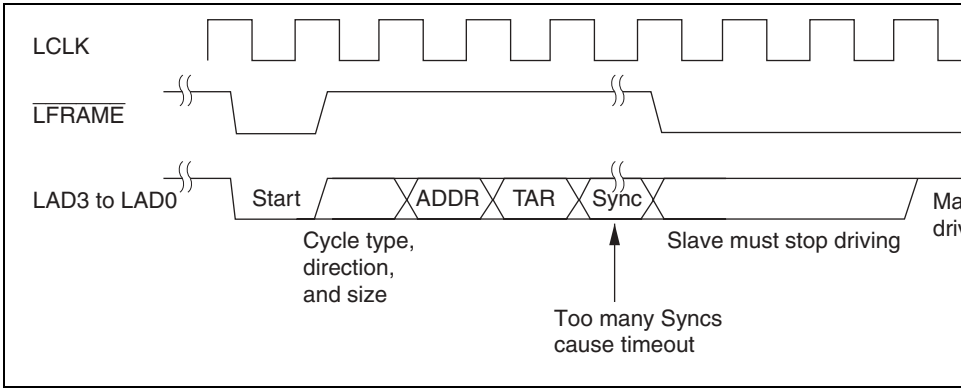


Figure 16.3 Abort Mechanism

16.4.3 SMIC Mode Transfer Flow

Figure 16.4 shows the write transfer flow and figure 16.5 shows the read transfer flow in
mode.

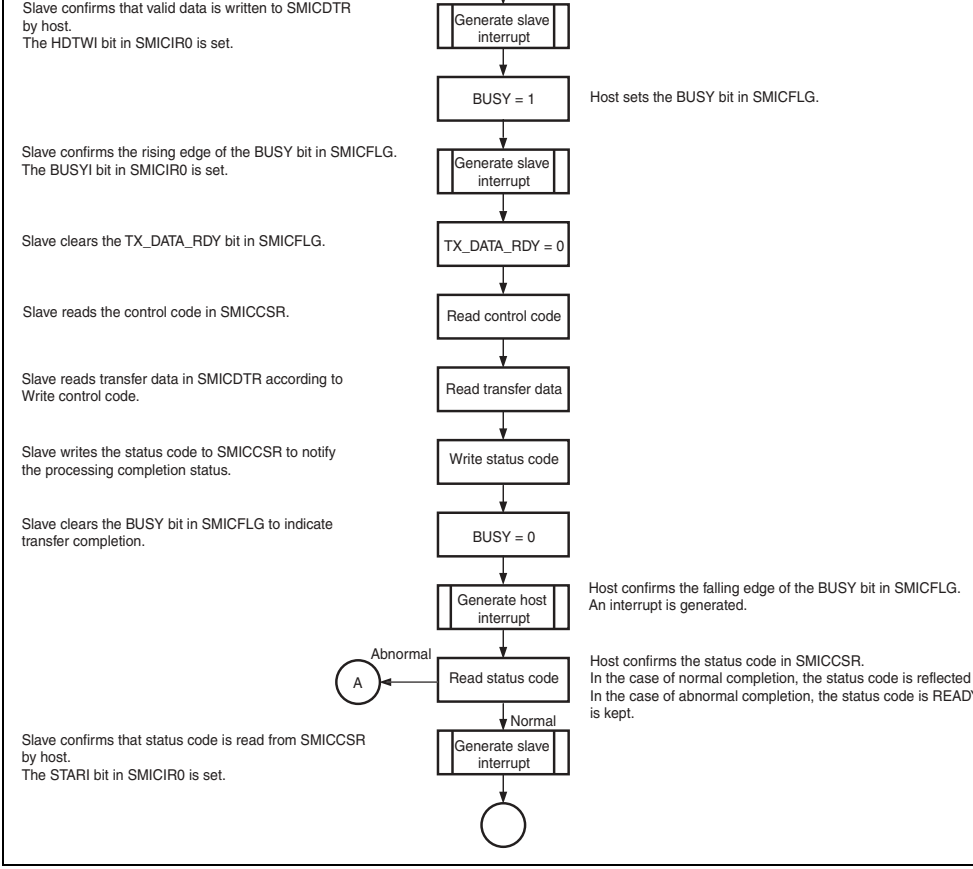


Figure 16.4 SMIC Write Transfer Flow

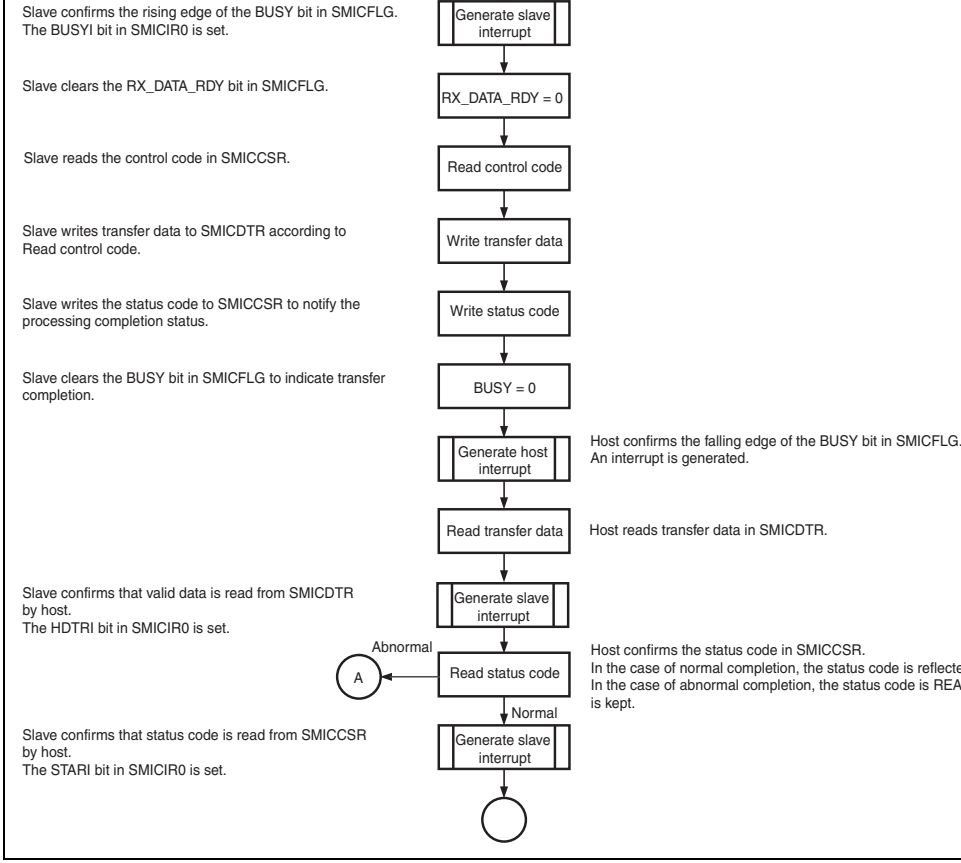


Figure 16.5 SMIC Read Transfer Flow

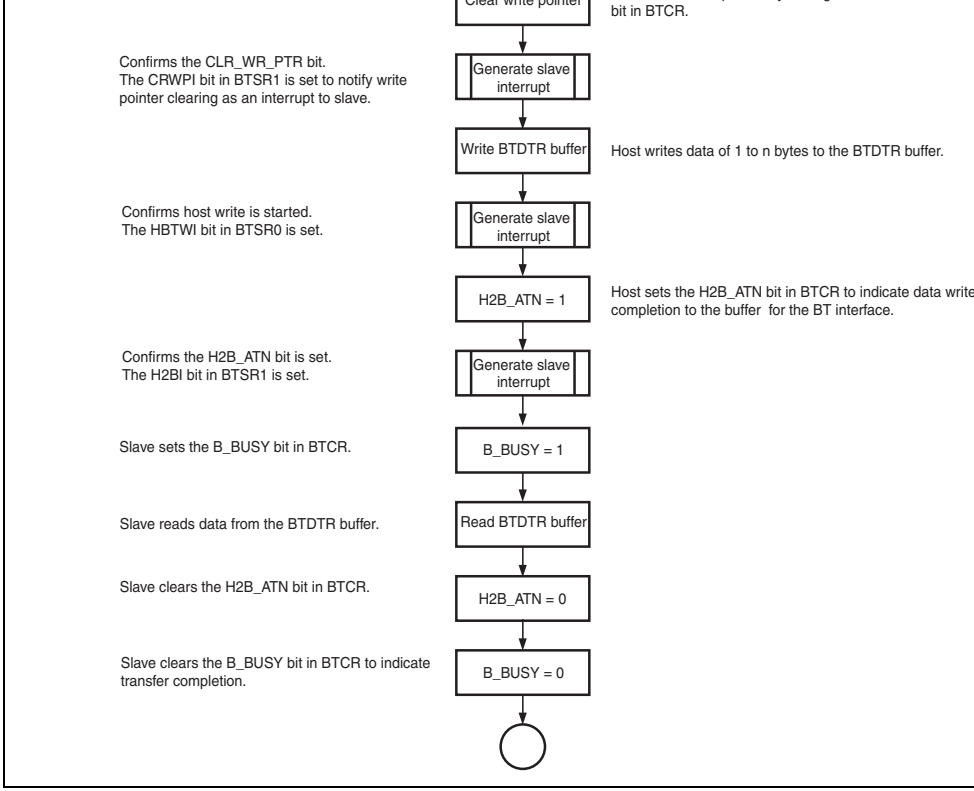


Figure 16.6 BT Write Transfer Flow

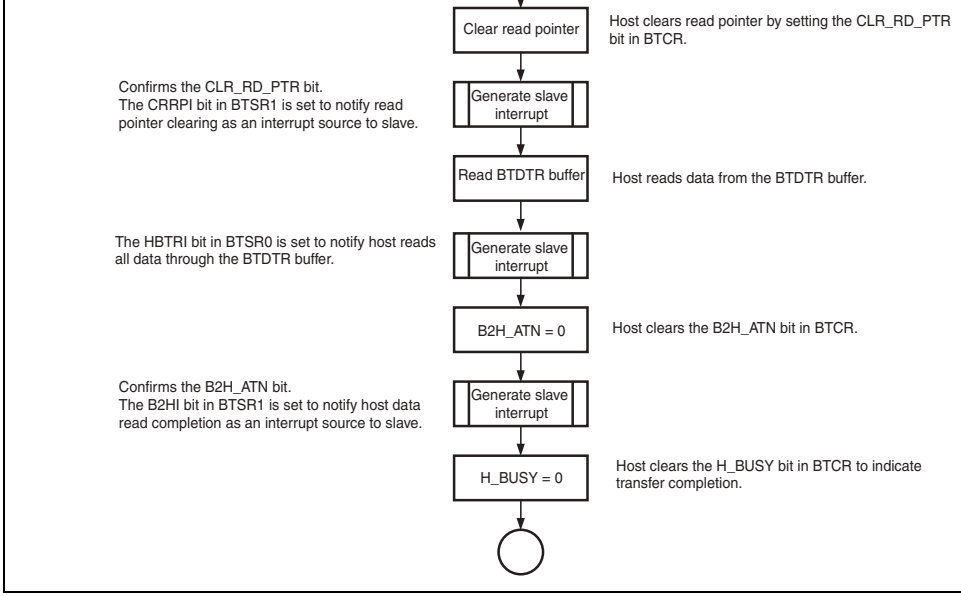


Figure 16.7 BT Read Transfer Flow

Fast A20 Gate Operation: The internal state of GA20 output is initialized to 1 when FG
0. When the FGA20E bit is set to 1, PD3/GA20 is used for output of a fast A20 gate signa
state of the PD3/GA20 pin can be monitored by reading the GA20 bit in HICR2.

The initial output from this pin will be a logic 1, which is the initial value. Afterward, the
processor can manipulate the output from this pin by sending commands and data. This fu
only available via the IDR1 register. The LPC interface decodes commands input from th
When an H'D1 host command is detected, bit 1 of the data following the host command is
from the GA20 output pin. This operation does not depend on firmware or interrupts, and
than the regular processing using interrupts. Table 16.6 shows the conditions that set and
GA20 (PD3). Figure 16.8 shows the GA20 output in flowchart form. Table 16.7 indicates
GA20 output signal values.

Table 16.6 GA20 (PD3) Set/Clear Conditions

Pin Name	Setting Condition	Clearing Condition
GA20 (PD3)	When bit 1 of the data that follows an H'D1 host command is 1	When bit 1 of the data follows a host command is 0

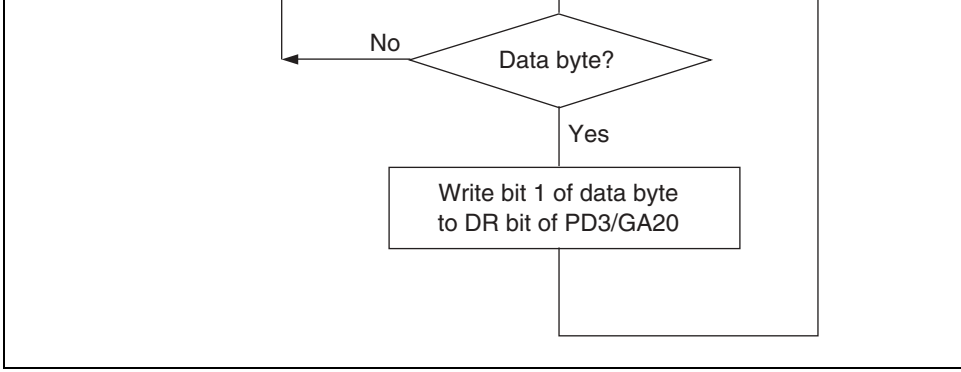


Figure 16.8 GA20 Output

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 sequer
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 sec
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. Arbitrary data with bit 1 set to 1.
2. Arbitrary data with bit 1 cleared to 0.

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 rise of the $\overline{\text{LPCPD}}$ signal. Taking these po consideration, the following operating procedure uses a combination of LPC software sh 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 shutdown mode.
5. Set the SDWNE bit to 1 and make a transition to LPC hardware shutdown mode. Th 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 rise steps 3 to 5. If the signal has risen, clear the SDWNE bit to 0 to return to the state in
7. Place the slave processor in sleep mode or software standby mode as necessary.
8. If software standby mode has been set, exit software standby mode by some means independent of the LPC.
9. When a rising edge is detected in the $\overline{\text{LPCPD}}$ signal, the SDWNE bit is automaticall to 0. If the slave processor has been placed in sleep mode, the mode is exited by mea $\overline{\text{LRESET}}$ signal input, on completion of the LPC transfer cycle, or by some other me

Table 16.8 shows the scope of LPC interface pin shutdown.

GA20	PD3	Δ	I/O	Hi-Z, only when FGA20E = 1
$\overline{\text{CLKRUN}}$	PD4	O	I/O	Hi-Z
$\overline{\text{LPCPD}}$	PD5	X	Input	Needed to clear shutdown state

Notes: O: Pins shut down by the shutdown function
Δ: Pins shut down only when the LPC function is selected by register setting
X: Pins not shut down

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 $\overline{\text{STBY}}$ or $\overline{\text{RES}}$ pin input, or WDT0 overflow)
   — All register bits, including bits LPC3E 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 16.9.

Host interrupt enable bits (IRQ1E1, IRQ12E1, SMIE2, IRQ6E2, IRQ9E2 to IRQ11E2, SMIE3B, SMIE3A, IRQ6E3, IRQ9E3 to IRQ11E3, SELREQ, IEDIR, IEDIR3, SMICIR1), Q/$\overline{C}$ flag	Initialized	Initialized	Reta
LRST flag	Initialized (0)	Can be set/cleared	Can set/
SDWN flag	Initialized (0)	Initialized (0)	Can set/
LRSTB bit	Initialized (0)	HR: 0 SR: 1	0 (c
SDWNB bit	Initialized (0)	Initialized (0)	HS: SS:
SDWNE bit	Initialized (0)	Initialized (0)	HS: SS:
LPC interface operation control bits (LPC3E to LPC1E, FGA20E, LADR12, LADR3, IBFIE1 to IBFIE3, PMEE, PMEB, LSMIE, LSMIB, LSCIE, LSCIB, TWRE, SELSTR3, SELIRQ1, SELSMI, SELIRQ6, SELIRQ9, SELIRQ10, SELIRQ11, SELIRQ12, HICR4, HISEL, BTCSR0, BTCSR1)	Initialized	Retained	Reta
$\overline{\text{LRESET}}$ signal	Input (port function)	Input	Inpu
$\overline{\text{LPCPD}}$ signal		Input	Inpu
LAD3 to LAD0, $\overline{\text{LFRAME}}$, LCLK, SERIRQ, $\overline{\text{CLKRUN}}$ signals		Input	Hi-Z
$\overline{\text{PME}}$, $\overline{\text{LSMI}}$, LSCI, GA20 signals (when function is selected)		Output	Hi-Z
$\overline{\text{PME}}$, $\overline{\text{LSMI}}$, LSCI, GA20 signals (when function is not selected)		Port function	Port

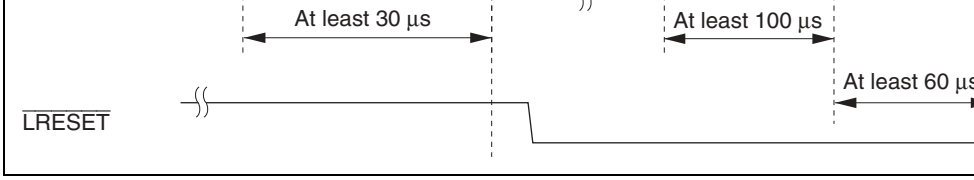


Figure 16.9 Power-Down State Termination Timing

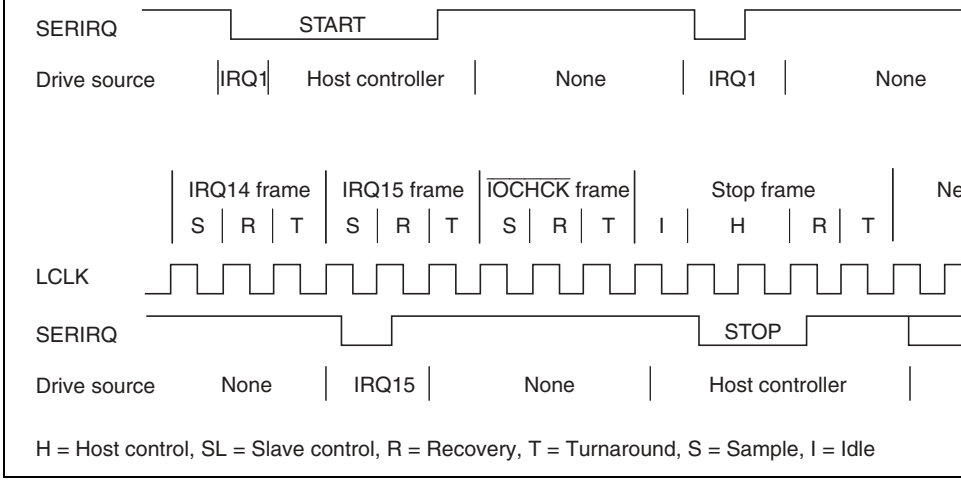


Figure 16.10 SERIRQ Timing

The serialized interrupt transfer cycle frame configuration is as follows. Two of the state comprising each frame are the recover state in which the SERIRQ signal is returned to t at the end of the frame, and the turnaround state in which the SERIRQ signal is not drive recover state must be driven by the host or slave processor that was driving the precedin

6	IRQ5	Slave	3	
7	IRQ6	Slave	3	Drive possible in LPC channels 2 and
8	IRQ7	Slave	3	
9	IRQ8	Slave	3	
10	IRQ9	Slave	3	Drive possible in LPC channels 2 and
11	IRQ10	Slave	3	Drive possible in LPC channels 2 and
12	IRQ11	Slave	3	Drive possible in LPC channels 2 and
13	IRQ12	Slave	3	Drive possible in LPC channel 1
14	IRQ13	Slave	3	
15	IRQ14	Slave	3	
16	IRQ15	Slave	3	
17	IOCHCK	Slave	3	
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

There are two modes—continuous mode and quiet mode—for serialized interrupts. The m
initiated in the next transfer cycle is selected by the stop frame of the serialized interrupt
cycle that ended before that cycle.

In continuous mode, the host initiates host interrupt transfer cycles at regular intervals. In
mode, the slave processor with interrupt sources requiring a request can also initiate an in
transfer cycle, in addition to the host. In quiet mode, since the host does not necessarily in
interrupt transfer cycles, it is possible to suspend the clock (LCLK) supply and enter the p
down state. In order for a slave to transfer an interrupt request in this case, a request to re

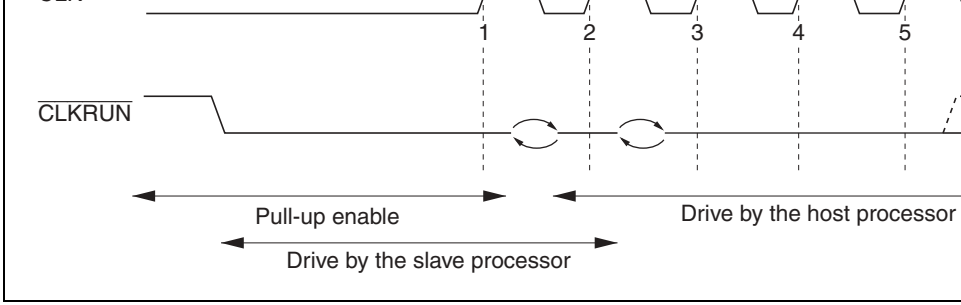


Figure 16.11 Clock Start or Speed-Up

Cases other than SERIRQ in quiet mode when clock restart is required must be handled different protocol, using the $\overline{\text{PME}}$ signal, etc.

Interrupt	Description
IBFI1	Requested when IBFIE1 is set to 1 and IDR1 reception is completed
IBFI2	Requested when IBFIE2 is set to 1 and IDR2 reception is completed
IBFI3	Requested when IBFIE3 is set to 1 and IDR3 reception is completed, or TWRE and IBFIE3 are set to 1 and reception is completed up to TWR15
	Interrupts by HDTWI, HDTRI, STARI, CTLWI, and BUSYI of SMIC mode
	Interrupts by FRDI, HRDI, HWRI, HBTWI, HBTRI, HRSTI, IRQCRI, BEV H2BI, CRRPI, and CRWPI of BT mode
ERRI	Requested when ERRIE is set to 1 and LRST, SDWN, or ABRT is set to

16.5.2 SMI, HIRQ1, HIRQ6, HIRQ9, HIRQ10, HIRQ11, HIRQ12

The LPC interface can request seven kinds of host interrupt by means of SERIRQ. HIRQ HIRQ12 are used on LPC channel 1 only, while SMI, HIRQ6, HIRQ9, HIRQ10, and HIR be requested from LPC channel 2 or 3.

There are two ways of clearing a host interrupt request.

When the IEDIR bit in SIRQCR0 and the IEDIR3 bit in SIRQCR2 are cleared to 0s, host sources and LPC channels are all linked to the host interrupt request enable bits. When th flag is cleared to 0 by a read by the host of ODR or TWR15 in the corresponding LPC ch corresponding host interrupt enable bit is automatically cleared to 0, and the host interrup is cleared.

When the IEDIR bit in SIRQCR0 and the IEDIR3 bit in SIRQCR2 are set to 1s, LPC cha and 3 interrupt requests are dependent only upon the host interrupt enable bits. The host i enable bit is not cleared when OBF for channel 2 or 3 is cleared. Therefore, SMIE2, SMI SMIE3B, IRQ6E2 and IRQ6E3, IRQ9E2 and IRQ9E3, IRQ10E2 and IRQ10E3, and IRQ and IRQ11E3 lose their respective functional differences when both bits IEDIR and IEDI

SMI (IEDIR = 0)	Slave • writes to ODR2, then reads 0 from bit SMIE2, and writes 1	Slave • writes 0 to bit SMIE2, o reads ODR2
SMI (IEDIR3 = 0)	Slave • writes to ODR3, then reads 0 from bit SMIE3A, and writes 1 • writes to TWR15, then reads 0 from bit SMIE3B, and writes 1	Slave • writes 0 to bit SMIE3A, reads ODR3 • writes 0 to bit SMIE3B, reads TWR15
SMI (IEDIR = 1)	Slave • reads 0 from bit SMIE2, then writes 1	• Slave writes 0 to bit SM
SMI (IEDIR3 = 1)	Slave • reads 0 from bit SMIE3A, then writes 1 • reads 0 from bit SMIE3B, then writes 1	Slave • writes 0 to bit SMIE3A • writes 0 to bit SMIE3B
HIRQi (i = 6, 9, 10, 11) (IEDIR = 0)	Slave • writes to ODR2, then reads 0 from bit IRQiE2, and writes 1	Slave • writes 0 to bit IRQiE2, o reads ODR2
HIRQi (i = 6, 9, 10, 11) (IEDIR3 = 0)	Slave • writes to ODR3, then reads 0 from bit IRQiE3 and writes 1	Slave • writes 0 to bit IRQiE3, o reads ODR3
HIRQi (i = 6, 9, 10, 11) (IEDIR = 1)	Slave • reads 0 from bit IRQiE2, then writes 1	Slave • writes 0 to bit IRQiE2
HIRQi (i = 6, 9, 10, 11) (IEDIR3 = 1)	Slave • reads 0 from bit IRQiE3, then writes 1	Slave • writes 0 to bit IRQiE3

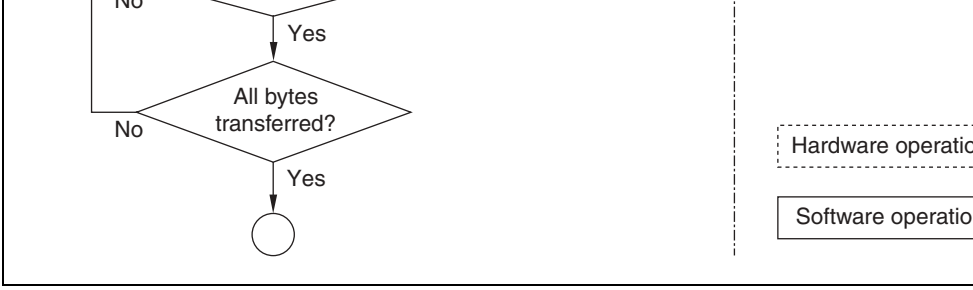


Figure 16.12 HIRQ Flowchart (Example of Channel 1)

contention. For example, if the host and slave processor both try to access IDR or ODR same time, the data will be corrupted. To prevent simultaneous accesses, IBF and OBF used to allow access only to data for which writing has finished.

Unlike the IDR and ODR registers, the transfer direction is not fixed for the bidirectiona registers (TWR). MWMF and SWMF are provided in STR to handle this situation. Afte to TWR0, MWMF and SWMF must be used to confirm that the right to access TWR1 t has been obtained.

Table 16.13 shows the host address example of registers LADR3, IDR3, ODR3, STR3, TWR0MW, TWR0SW, and TWR1 to TWR15.

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

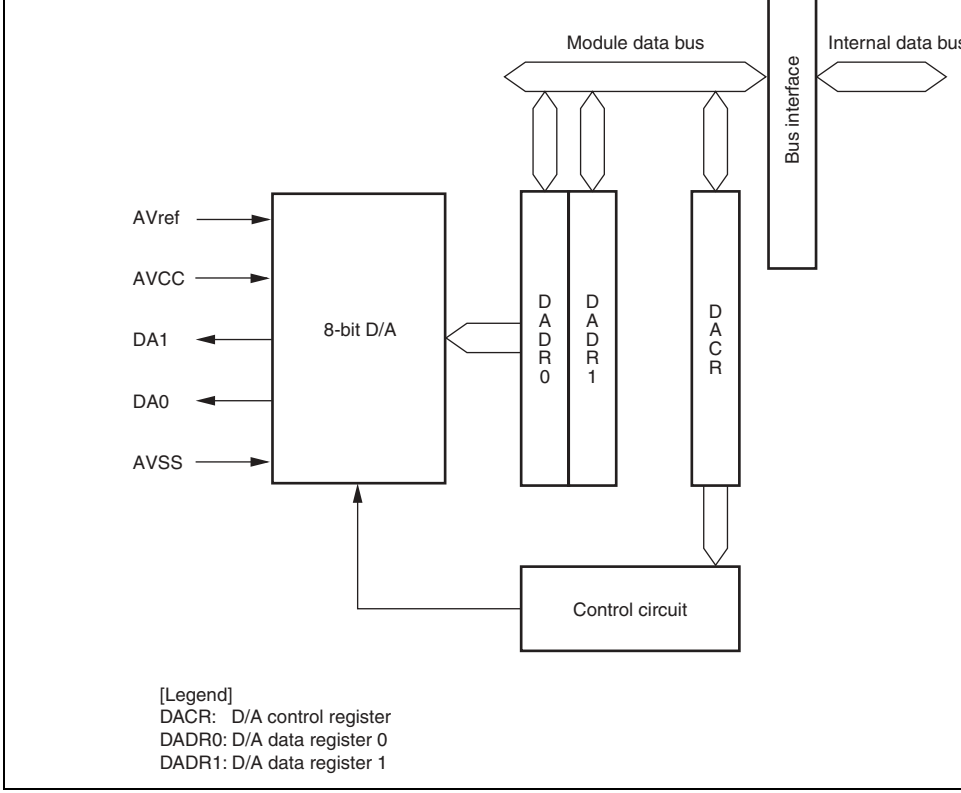


Figure 17.1 Block Diagram of D/A Converter

output pins.

17.3.2 D/A Control Register (DACR)

DACR controls D/A converter operation.

Bit	Bit Name	Initial Value	R/W	Description
7	DAOE1	0	R/W	D/A Output Enable 1 Controls D/A conversion and analog output. 0: Analog output DA1 is disabled 1: D/A conversion for channel 1 and analog ou are enabled
6	DAOE0	0	R/W	D/A Output Enable 0 Controls D/A conversion and analog output. 0: Analog output DA0 is disabled 1: D/A conversion for channel 0 and analog ou are enabled
5	DAE	0	R/W	D/A Enable Controls D/A conversion in conjunction with the and DAOE1 bits. When the DAE bit is cleared conversion for channels 0 and 1 are controlled individually. When the DAE bit is set to 1, D/A for channels 0 and 1 are controlled as one. Co result output is controlled by the DAOE0 and D bits. For details, see table 17.2 below.
4 to 0	—	All 1	R	Reserved The initial value should not be changed.

	1	Enables D/A conversion for channels 0 and 1

[Legend]

*: Don't care

conversion results are output from the analog output pin DA0. The conversion result output continuously until DADR0 is modified or the DAOE0 bit is cleared to 0. The value is calculated by the following formula:

DADR contents/256 x AVref

3. Conversion starts immediately after DADR0 is modified. After the interval of t_{DCONV}, conversion results are output.
4. When the DAOE bit is cleared to 0, analog output is disabled.

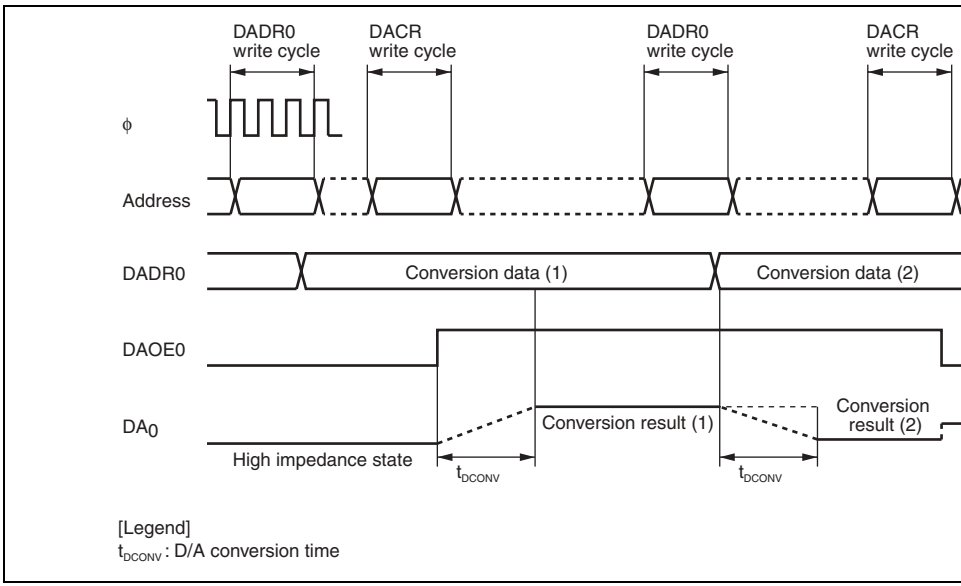
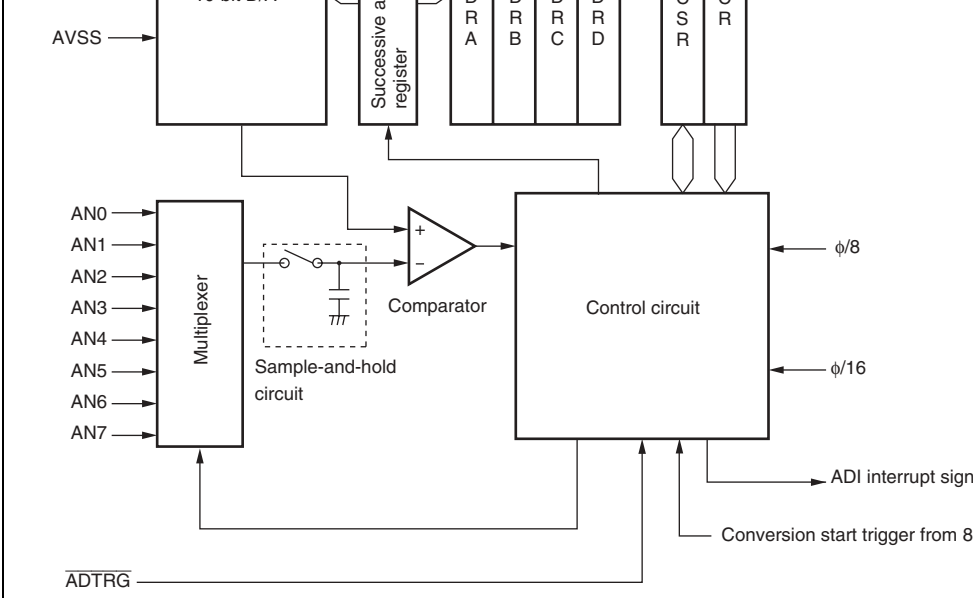


Figure 17.2 D/A Converter Operation Example

- Conversion time: 8.00 µs per channel (at 33-MHz operation)
- Two kinds of operating modes
  Single mode: Single-channel A/D conversion
  Scan mode: Continuous A/D conversion on 1 to 4 channels
- Four data registers
  Conversion results are held in a 16-bit data register for each channel
- Sample and hold function
- Three kinds of conversion start
  Software, 8-bit timer (TMR) conversion start trigger, or 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
ADCSR: A/D control/status register
ADDRA: A/D data register A
ADDRB: A/D data register B
ADDRC: A/D data register C
ADDRD: A/D data register D

Figure 18.1 Block Diagram of A/D Converter

Analog ground pin	AVSS	Input	Analog block ground and reference volta
Reference power supply pin	AVref	Input	Reference voltage for A/D conversion
Analog input pin 0	AN0	Input	Group 0 analog input pins
Analog input pin 1	AN1	Input	
Analog input pin 2	AN2	Input	
Analog input pin 3	AN3	Input	
Analog input pin 4	AN4	Input	Group 1 analog input pins
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 pin for starting A/D conversion

18.3.1 A/D Data Registers A to D (ADDRA to ADDRD)

There are four 16-bit read-only ADDR registers, ADDRA to ADDRD, used to store the r
A/D conversion. The ADDR registers, which store a conversion result for each channel, a
in table 18.2.

The converted 10-bit data is stored to bits 15 to 6. The lower 6-bit data is always read as 0

The data bus between the CPU and the A/D converter is 8-bit width. The upper byte can b
directly from the CPU, but the lower byte should be read via a temporary register. The te
register contents are transferred from the ADDR when the upper byte data is read. When
the ADDR, read only the upper byte in byte units or read in word units.

Table 18.2 Analog Input Channels and Corresponding ADDR Registers

Analog Input Channel		A/D Data Register to Store A/D Conv
Group 0	**Group 1**	**Results**
AN0	AN4	ADDRA
AN1	AN5	ADDRB
AN2	AN6	ADDRC
AN3	AN7	ADDRD

				• When 0 is written after reading ADF = 1 • When DTC starts by an ADI interrupt an read
6	ADIE	0	R/W	A/D Interrupt Enable Enables ADI interrupt by ADF when this bit
5	ADST	0	R/W	A/D Start Setting this bit to 1 starts A/D conversion. In mode, this bit is cleared to 0 automatically w conversion on the specified channel ends. I mode, conversion continues sequentially on specified channels until this bit is cleared to software, a reset, or a transition to standby module stop mode.
4	SCAN	0	R/W	Scan Mode Selects the A/D conversion operating mode 0: Single mode 1: Scan mode
3	CKS	0	R/W	Clock Select Sets A/D conversion time. 0: Conversion time is 266 states (max) 1: Conversion time is 134 states (max) (when the system clock (ϕ) is 16 MHz or low Switch conversion time while ADST is 0.

	111: AN7	111: AN4 to AN

Note: * Only 0 can be written for clearing the flag.

18.3.3 A/D Control Register (ADCR)

ADCR enables A/D conversion started by an external trigger signal.

Bit	Bit Name	Initial Value	R/W	Description
7	TRGS1	0	R/W	Timer Trigger Select 1 and 0
6	TRGS0	0	R/W	Enable the start of A/D conversion by a trigge Only set bits TRGS1 and TRGS0 while A/D conversion is stopped (ADST = 0). 00: A/D conversion start by external trigger is 01: A/D conversion start by external trigger is 10: A/D conversion start by conversion trigge TMR 11: A/D conversion start by $\overline{\text{ADTRG}}$ pin
5 to 0	—	All 1	R/W	Reserved The initial values should not be changed.

1. A/D conversion on the specified channel is started when the ADST bit in ADCSR is by software or an external trigger input.
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 bi 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 A automatically cleared to 0, and the A/D converter enters wait state.

18.4.2 Scan Mode

In scan mode, A/D conversion is to be performed sequentially on the specified channels channels max.). Operations are as follows.

1. When the ADST bit in ADCSR is set to 1 by software or an external trigger input, A conversion starts on the first channel in the group (AN0 when the CH2 bit in ADCSI AN4 when the CH2 bit in ADCSR is 1).
2. When A/D conversion for each channel is completed, the result is sequentially transf the A/D data register corresponding to each channel.
3. When conversion of all the selected channels is completed, the ADF 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.
4. The ADST bit is not automatically cleared to 0 so steps [2] to [3] are repeated as lon ADST bit remains set to 1. When the ADST bit is cleared to 0, A/D conversion stops

(fixed) when CKS = 1.

Use the conversion time of 134 state only when the system clock (ϕ) is 16 MHz or lower.

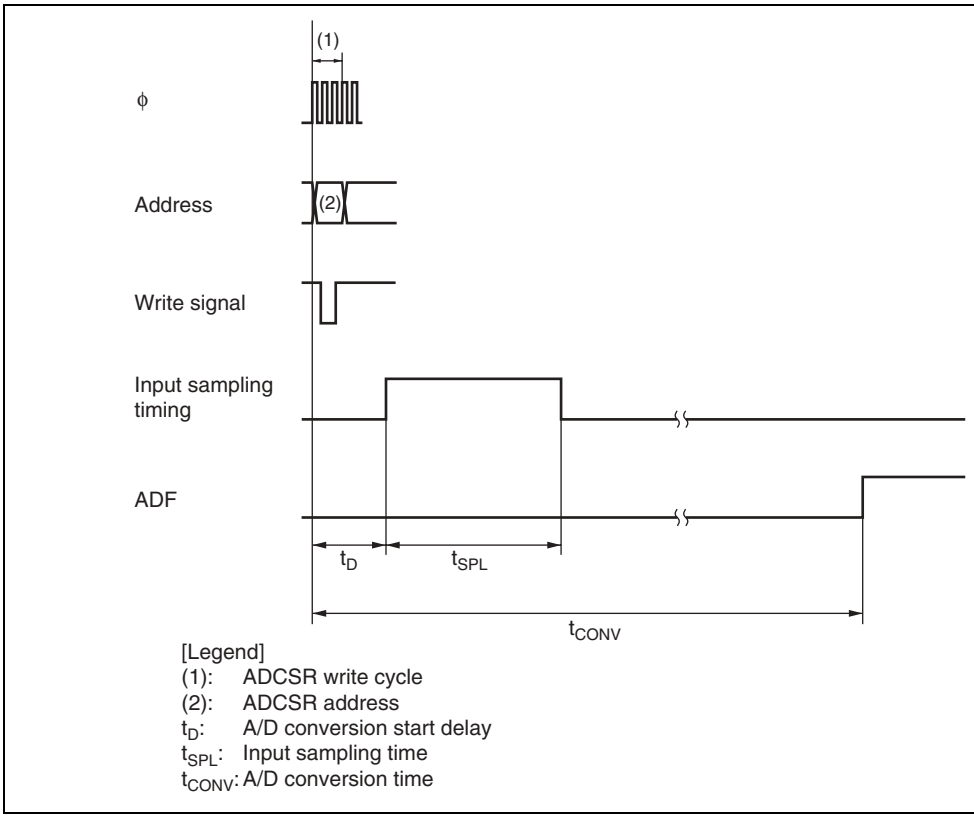


Figure 18.2 A/D Conversion Timing

A/D conversion can be externally triggered. When the TRGS1 and TRGS0 bits are set to
ADCR, external trigger input is enabled at the $\overline{\text{ADTRG}}$ pin. A falling edge at the $\overline{\text{ADTR}}$
the ADST bit to 1 in ADCSR, starting A/D conversion. Other operations, in both single
modes, are the same as when the ADST bit has been set to 1 by software. Figure 18.3 sh
timing.

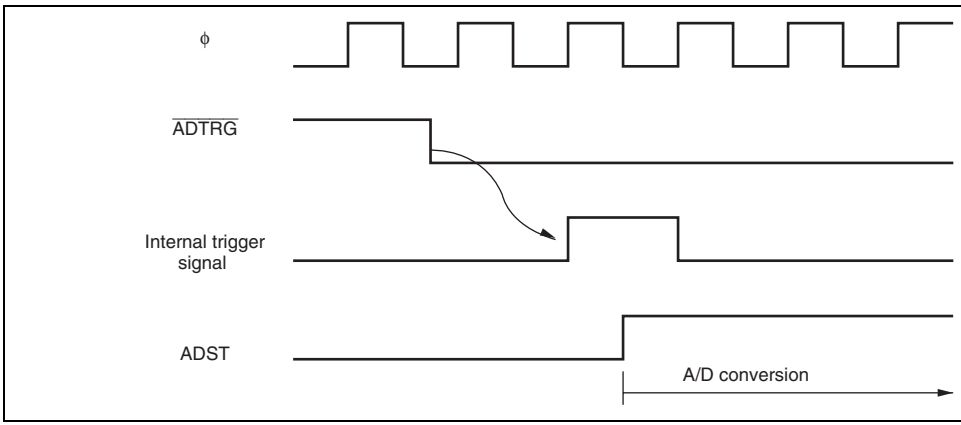


Figure 18.3 External Trigger Input Timing

18.6 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 figure 18.4).
- Offset error

  The deviation of the analog input voltage value from the ideal A/D conversion charac
  when the digital output changes from the minimum voltage value B'00 0000 0000 (H'
  B'00 0000 0001 (H'001) (see figure 18.5).
- Full-scale error

  The deviation of the analog input voltage value from the ideal A/D conversion charac
  when the digital output changes from B'11 1111 1110 (H'3FE) to B'11 1111 1111 (H'
  figure 18.5).
- Nonlinearity error

  The error with respect to the ideal A/D conversion characteristics between the zero vo
  the full-scale voltage. Does not include the offset error, full-scale error, or quantizatio
  (see figure 18.5).
- Absolute accuracy

  The deviation between the digital value and the analog input value. Includes the offse
  full-scale error, quantization error, and nonlinearity error.

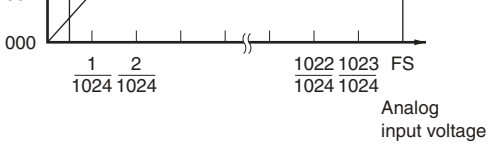


Figure 18.4 A/D Conversion Accuracy Definitions

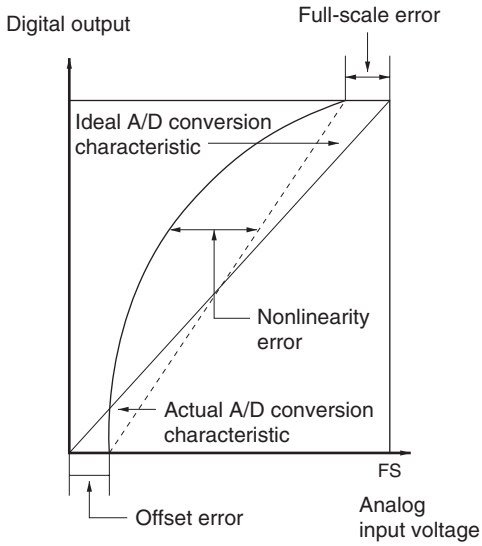


Figure 18.5 A/D Conversion Accuracy Definitions

differential coefficient (e.g., voltage fluctuation ratio of 5 mV/µs or greater) (see figure 1
When converting a high-speed analog signal or converting in scan mode, a low-impedanc
should be inserted.

18.7.2 Influences on Absolute Accuracy

Adding capacitance results in coupling with GND, and therefore noise in GND may adve
affect the absolute accuracy. Be sure to make the connection to an electrically stable GNI
AVSS.

Care is also required to insure that filter circuits do not communicate with digital signals
mounting board, so acting as antennas.

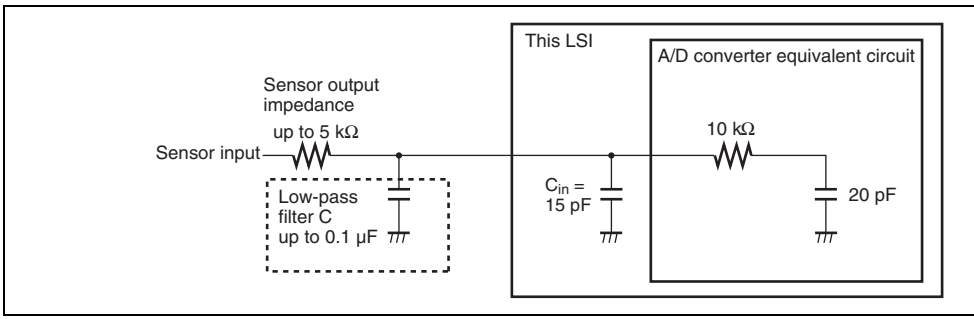


Figure 18.6 Example of Analog Input Circuit

The reference voltage of the AVref pin should be in the range AVref ≤ AVCC.

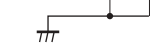
18.7.4 Notes on Board Design

In board design, digital circuitry and analog circuitry should be as mutually isolated as p
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 o
of the analog circuitry due to inductance, adversely affecting A/D conversion values. Al
circuitry must be isolated from the analog input signals (AN0 to AN7), and analog powe
(AVCC) by the analog ground (AVSS). Also, the analog ground (AVSS) should be conn
one point to a stable digital ground (VSS) on the board.

18.7.5 Notes on Noise Countermeasures

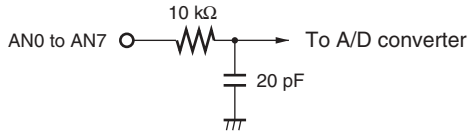
A protection circuit connected to prevent damage due to an abnormal voltage such as an
surge at the analog input pins (AN0 to AN7) should be connected between AVCC and A
shown in figure 18.7. Also, the bypass capacitors connected to AVCC and AVref, and th
capacitors connected to AN0 to AN7 must be connected to AVSS.

If a filter capacitor is connected, the input currents at the analog input pins (AN0 to AN7
averaged, and so an error may arise. Also, when A/D conversion is performed frequently
scan mode, if the current charged and discharged by the capacitance of the sample-and-h
in the A/D converter exceeds the current input via the input impedance (R_{in}), an error wi
the analog input pin voltage. Careful consideration is therefore required when deciding t
constants.



*2 R_{in}: Input impedance

Figure 18.7 Example of Analog Input Protection Circuit

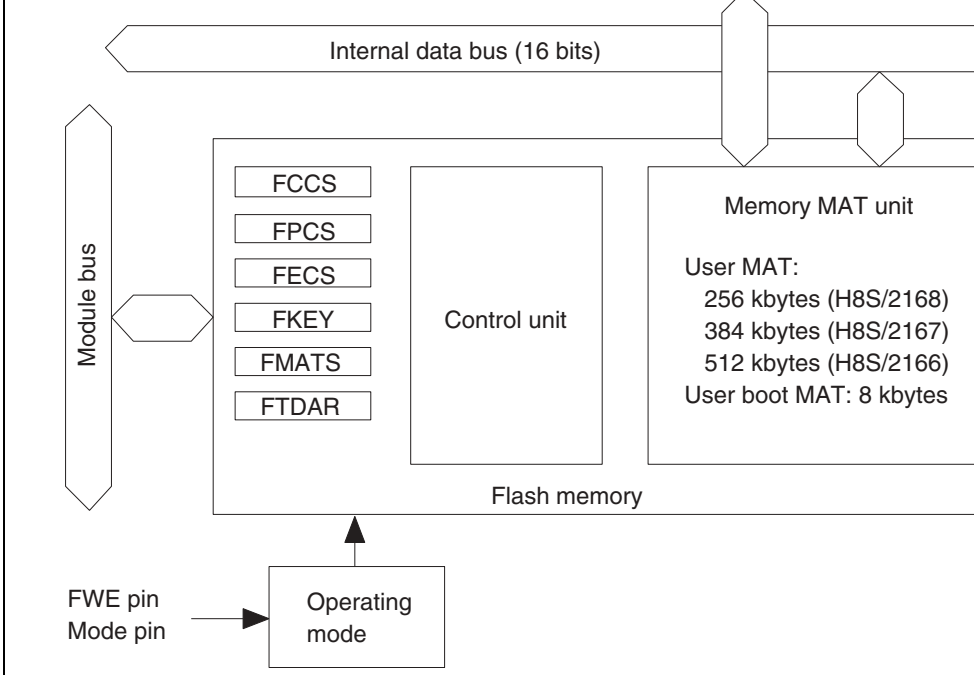


Note: Values are reference values.

Figure 18.8 Analog Input Pin Equivalent Circuit

H8S/2166	HD64F2166	512 kbytes	H'000000 to H'07FFFF

- Two flash-memory MATs according to LSI initiation mode
  The on-chip flash memory has two memory spaces in the same address space (herea
  referred to as memory MATs). The mode setting in the initiation determines which m
  MAT is initiated first. The MAT can be switched by using the bank-switching metho
  initiation.
  - The user memory MAT is initiated at a power-on reset in user mode: 256 kbytes
    (H8S/2168), 384 kbytes (H8S/2167), 512 kbytes (H8S/2166)
  - The user boot memory MAT is initiated at a power-on reset in user boot mode: 8
- Programming/erasing interface by the download of on-chip program
  This LSI has a dedicated programming/erasing program. After downloading this pro
  the on-chip RAM, programming/erasing can be performed by setting the argument p
- Programming/erasing time
  The flash memory programming time is 3 ms (typ) in 128-byte simultaneous progra
  approximately 25 µs per byte. The erasing time is 1000 ms (typ) per 64-kbyte block.
- Number of programming
  The number of flash memory programming can be up to 100 times at the minimum.
  ranged from 1 to 100 is guaranteed.)
- Three on-board programming modes
  - Boot mode
    This mode is a program mode that uses an on-chip SCI interface. The user MAT
    boot MAT can be programmed. This mode can automatically adjust the bit rate b
    host and this LSI.
  - User program mode
    The user MAT can be programmed by using the optional interface.



[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 20.1 Block Diagram of Flash Memory

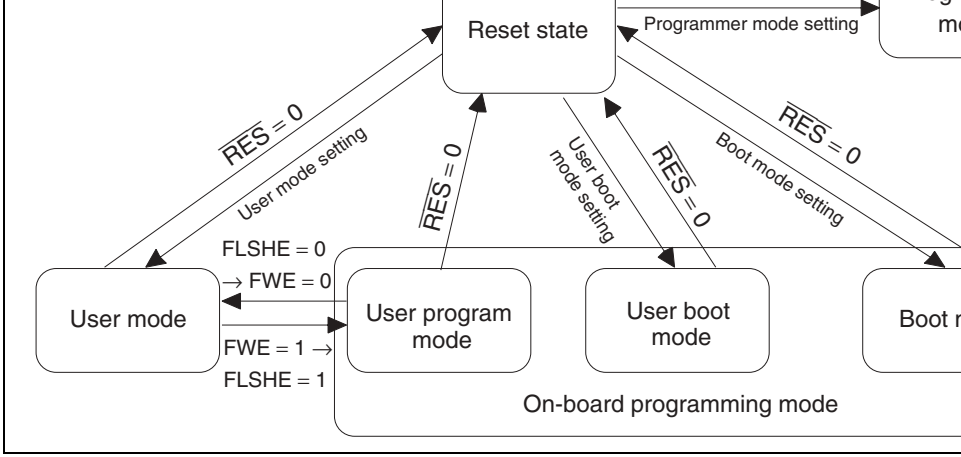


Figure 20.2 Mode Transition of Flash Memory

MAT				User boo
All erasure	○ (Automatic)	○	○	○ (Autom
Block division erasure	○ *1	○	○	×
Program data transfer	From host via SCI	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.

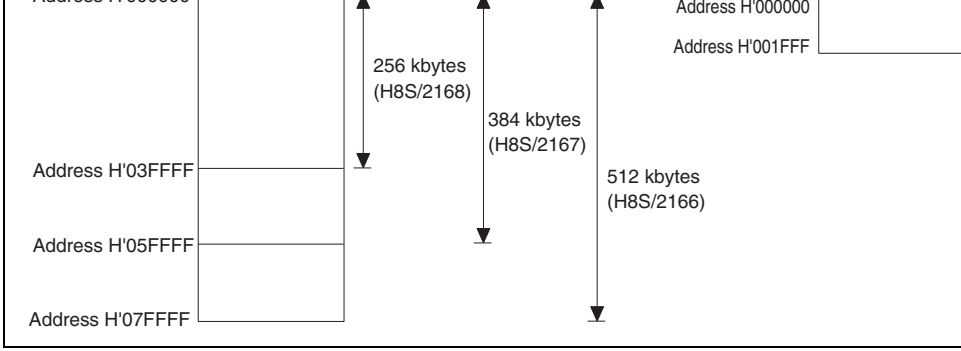
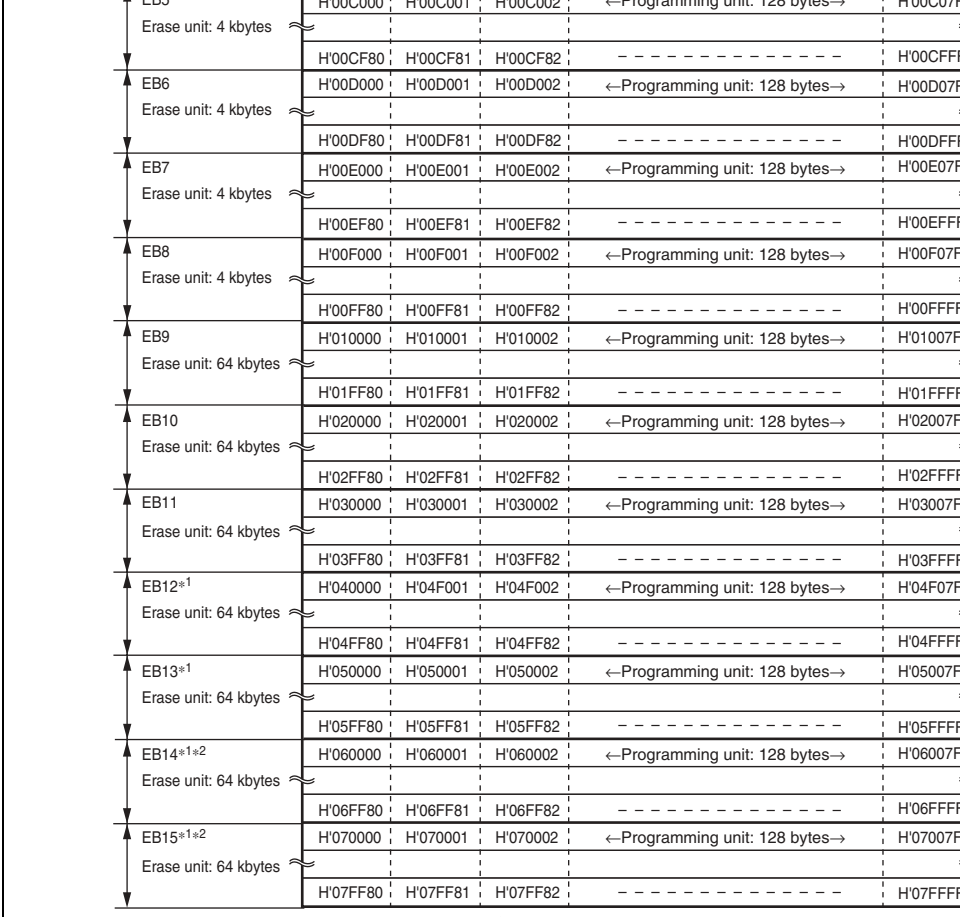


Figure 20.3 Flash Memory Configuration

The size of the user MAT is different from that of the user boot MAT. An address which
the size of the 8-kbyte user boot MAT should not be accessed. If the attempt is made, da
as undefined value.



Notes: 1. EB12 to EB15 are not available in the H8S/2168.
2. EB14 and EB15 are not available in the H8S/2167.

Figure 20.4 Block Division of User MAT

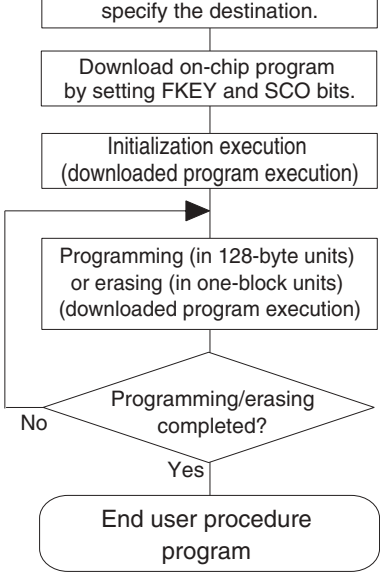


Figure 20.5 Overview of User Procedure Program

1. Selection of on-chip program to be downloaded

   For programming/erasing execution, the FLSHE bit in STCR must be set to 1 to trans
   user program mode.

   This LSI has programming/erasing programs which can be downloaded to the on-chip
   The on-chip program to be downloaded is selected by setting the corresponding bits in
   programming/erasing interface register. The address of the programming destination i
   specified by the flash transfer destination address register (FTDAR).

The operating frequency is set before execution of programming/erasing. This settin
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
   to transition to user program mode.

   The program data/programming destination address is specified in 128-byte units wh
   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 a
   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 mer

   All interrupts are prohibited during programming and erasing. Interrupts must be ma
   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 erasur
   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 processi
   download and initialization are not required when the same processing is executed
   consecutively.

TxD1	Output	Serial transmit data output (used in boot mode)
RxD1	Input	Serial receive data input (used in boot mode)

20.3 Register Descriptions

The registers/parameters which control flash memory are shown in the following. To read write to these registers/parameters, the FLSHE bit in the serial timer control register (STC be set to 1. For details on STCR, see section 3.2.3, Serial Timer Control Register (STCR)

- Flash code control status register (FCCS)
- Flash program code select register (FPCS)
- Flash erase code select register (FECS)
- Flash key code register (FKEY)
- 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 20.3.

Parameter					
	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.

20.3.1 Programming/Erasing Interface Register

The programming/erasing interface registers are as described below. They are all 8-bit re that can be accessed in byte. These registers are initialized at a reset or in hardware stand

				The initial value should not be changed.
4	FLER	0	R	Flash Memory Error Indicates an error occurs during programming a erasing flash memory. When FLER is set to 1, f memory enters the error protection state. When FLER is set to 1, high voltage is applied t internal flash memory. To reduce the damage to memory, the reset must be released after the re period of 100 μs which is longer than normal. 0: Flash memory operates normally. Programming/erasing protection for flash mer (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 mer (error protection) is valid. [Setting conditions] • When an interrupt, such as NMI, occurs duri programming/erasing flash memory. • When the flash memory is read during programming/erasing flash memory (includir vector read or an instruction fetch). • When the SLEEP instruction is executed du programming/erasing flash memory (includir software-standby mode) • When a bus master other than the CPU, suc DTC, gets bus mastership during programming/erasing flash memory.

				vector is not read, resulting in the CPU runawa 0: The space for the interrupt vector table is r modified. When interrupt vector data is not read suc the operation for the interrupt exception ha cannot be guaranteed. An occurrence of a interrupts should be masked. 1: The space for the interrupt vector table is r Even when interrupt vector data is not rea successfully, the interrupt exception handli vector number 31 is enabled.
2, 1	—	All 0	R/W	Reserved The initial value should not be changed.
0	SCO	0	(R)/W*	Source Program Copy Operation Requests the on-chip programming/erasing pr be downloaded to the on-chip RAM. When this bit is set to 1, the on-chip program w selected by FPCS/FECS is automatically down the on-chip RAM specified by FTDAR. In order to set this bit to 1, H′A5 must be writte and this operation must be executed in the on- Four NOP instructions must be executed imme 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/erasi program to the on-chip RAM is not executed

- Flash Program Code Select Register (FPCS)

FPCS selects the on-chip programming program to be downloaded.

Bit	Bit Name	Initial Value	R/W	Description
7 to 1	—	All 0	R/W	Reserved The initial value should not be changed.
0	PPVS	0	R/W	Program Pulse Verify Selects the programming program. 0: On-chip programming program is not selecte [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.

1	K1	0	R/W	Only when H'5A is written, programming/erasi
0	K0	0	R/W	executed. Even if the on-chip programming/era
				program is executed, the flash memory cannot
				programmed or erased when the value other th
				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

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

Note: * Set to 1 when in user boot mode, otherwise set to 0.

				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	
4	TDA4	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.
3	TDA3	0	R/W	
2	TDA2	0	R/W	
1	TDA1	0	R/W	H'00: H'FFE080 is specified as a start address download an on-chip program.
0	TDA0	0	R/W	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.

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 20.4. T
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 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

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)
0	SF	—	R/W	Success/Fail Returns the result whether download is ended r or not. The determination result whether progra downloaded to the on-chip RAM is read back ar transferred to the on-chip RAM is returned. 0: Downloading on-chip program is ended norm error) 1: Downloading on-chip program is ended abno (error occurs)

frequency in this LSI is 5 to 33 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 t multiplication function, set the frequency multip setting value must be calculated as the followi methods. 1. The operating frequency which is shown in must be rounded in a number to three deci and be shown in a number of two decimal 2. The value multiplied by 100 is converted to digit and is written to the FPEFEQ paramet (general register ER0). For example, when the operating frequency of is 33.000 MHz, the value is as follows. 1. The number to three decimal places of 33. rounded and the value is thus 33.00. 2. The formula that 33.00 × 100 = 3300 is cor the binary digit and B'0000,1100,1110,010 (H'0CE4) is set to ER0.

				1: Setting of operating frequency is abnormal
0	SF	—	R/W	Success/Fail Indicates whether initialization is completed nor 0: Initialization is ended normally (no error) 1: Initialization is ended abnormally (error occur

(3) Programming Execution

When flash memory is programmed, the programming destination address on the user MA 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 a register ER1. This parameter is called as flash multipurpose address area parameter (F

   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 pro data must be in the consecutive space which can be accessed by using the MOV.B ins of the CPU and in other than the flash memory space.

   When data to be programmed does not satisfy 128 bytes, the 128-byte program data n prepared by filling with the dummy code H'FF.

   The start address of the area in which the prepared program data is stored must be sto general register ER0. This parameter is called as flash multipurpose data destination a parameter (FMPDR).

For details on the program processing procedure, see section 20.4.2, User Program Mode

				programming is executed starting from the spe address of the user MAT. Therefore, the speci programming start address becomes a 128-by boundary and MOA6 to MOA0 are always 0.

(b) Flash multipurpose data destination parameter (FMPDR: general register ER0 of CP

This parameter stores the start address in the area which stores the data to be programme user MAT. When the storage destination of the program data is in flash memory, an erro 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 store program data for the user MAT. The consecuti byte data is programmed to the user MAT star 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.

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 erase bit is set to 1, there is a high possibility that the is partially rewritten. In this case, after removing factor, erase the user MAT. If FMATS is set to H'AA and the user boot MAT selected, an error occurs when programming is performed. In this case, both the user MAT and MAT are not rewritten. Programming of the user MAT should be performed in boot mode or prog mode. 0: Programming has ended normally 1: Programming has ended abnormally (progra result is not guaranteed)
4	FK	—	R/W	Flash Key Register Error Detect Returns the check result of the value of FKEY b start of the programming processing. 0: FKEY setting is normal (FKEY = H'5A) 1: FKEY setting is error (FKEY = value other tha
3	—	—	—	Unused Returns 0.
2	WD	—	R/W	Write Data Address Detect When the address in the flash memory area is s as the start address of the storage destination o program data, an error occurs. 0: Setting of write data address is normal 1: Setting of write data address is abnormal

				abnormal
0	SF	—	R/W	Success/Fail Indicates whether the program processing is e normally or not. 0: Programming is ended normally (no error) 1: Programming is ended abnormally (error oc

Bit	Bit Name	Value	R/W	Description
31 to 8	—	—	—	Unused These bits should be cleared to H′0.
7	EB7	—	R/W	Erase Block Set the erase-block number in the range from 0 corresponds to the EB0 block and 15 correspon EB15 block. The number other than 0 to 11, 0 t 0 to 15 should not be set in the H8S/2168, H8S and H8S/2166, respectively.
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	

				this bit. The state can be confirmed with the F FLER bits in FCCS. For conditions to enter the protection state, see section 20.5.3, Error Prot 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 cou erased or when flash-memory related register are partially changed. If this bit is set to 1, ther 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 bo selected, an error occurs when erasure is perf this case, both the user MAT and user boot MA erased. Erasing of the user boot MAT should b performed in boot mode or programmer mode. 0: Erasure has ended normally 1: Erasure has ended abnormally (erasure re guaranteed)
4	FK	—	R/W	Flash Key Register Error Detect Returns the check result of FKEY value before the erasing processing. 0: FKEY setting is normal (FKEY = H'5A) 1: FKEY setting is error (FKEY = value other th
3	EB	—	R/W	Erase Block Select Error Detect Returns the check result whether the specified block number is in the block range of the user 0: Setting of erase-block number is normal 1: Setting of erase-block number is abnormal

When the pin is set in on-board programming mode and the reset start is executed, the on programming state that can program/erase the on-chip flash memory is entered. On-board programming mode has three operating modes: boot mode, user program mode, and user mode.

For details of the pin setting for entering each mode, see table 20.5. For details of the stat transition of each mode for flash memory, see figure 20.2.

Table 20.5 Setting On-Board Programming Mode

Mode Setting	FWE	$\overline{\text{MD2}}$	MD1	MD0	NM
Boot mode	1	0	0	0	1
User program mode	1*	1	1	0	0/
User boot mode	1	0	0	0	0

Note: * Before downloading the programming/erasing programs, the FLSHE bit must b 1 to transition to user program mode.

20.4.1 Boot Mode

Boot mode executes programming/erasing user MAT and user boot MAT by means of th command and program data transmitted from the host using the on-chip SCI. The tool for transmitting the control command and program data must be prepared in the host. The SC communication mode is set to asynchronous mode. When reset start is executed after this pin is set in boot mode, the boot program in the microcomputer is initiated. After the SCI is automatically adjusted, the communication with the host is executed by means of the c command method.

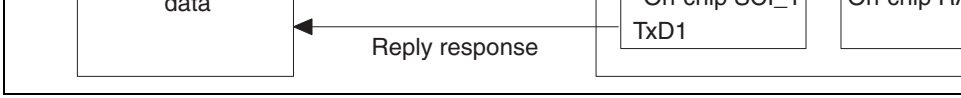


Figure 20.6 System Configuration in Boot Mode

(1) SCI Interface Setting by Host

When boot mode is initiated, this LSI measures the low period of asynchronous SCI-cor tion data (H'00), which is transmitted consecutively by the host. The SCI transmit/receiv is set to 8-bit data, 1 stop bit, and no parity. This LSI calculates the bit rate of transmissi host by means of the measured low period and transmits the bit adjustment end sign (1 b H'00) to the host. The host must confirm that this bit adjustment end sign (H'00) has bee normally and transmits 1 byte of H'55 to this LSI. When reception is not executed norma mode is initiated again (reset) and the operation described above must be executed. The between the host and this LSI is not matched by the bit rate of transmission by the host a clock frequency of this LSI. To operate the SCI normally, the transfer bit rate of the hos set to 4,800 bps, 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 20.6. Boot mode must be initiated in the range system clock.

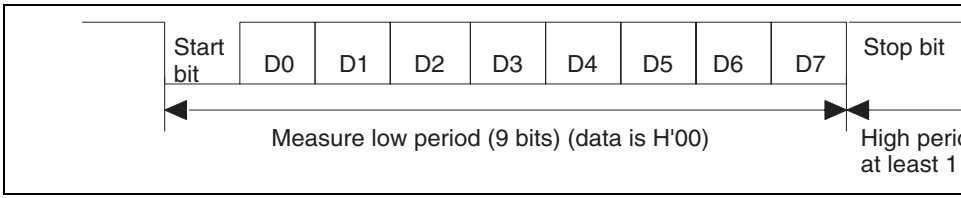


Figure 20.7 Automatic-Bit-Rate Adjustment Operation of SCI

After boot mode is initiated, the bit rate of the SCI interface is adjusted with that of th

2. Waiting for inquiry set command

   For inquiries about user-MAT size and configuration, MAT start address, and support required information is transmitted to the host.

3. Automatic erasure of all user MAT and user boot MAT

   After inquiries have finished, all user MAT and user boot MAT are automatically eras

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.

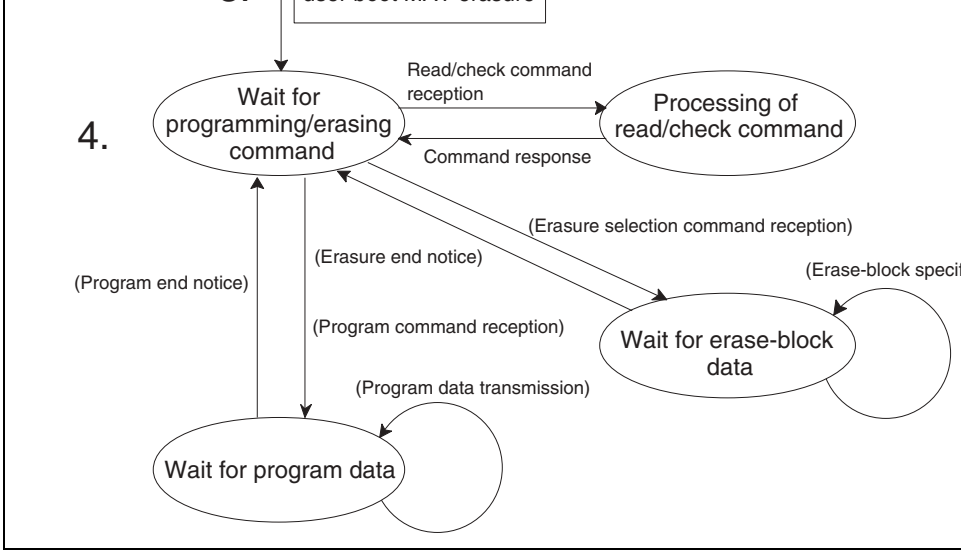


Figure 20.8 Overview of Boot Mode State Transition Diagram

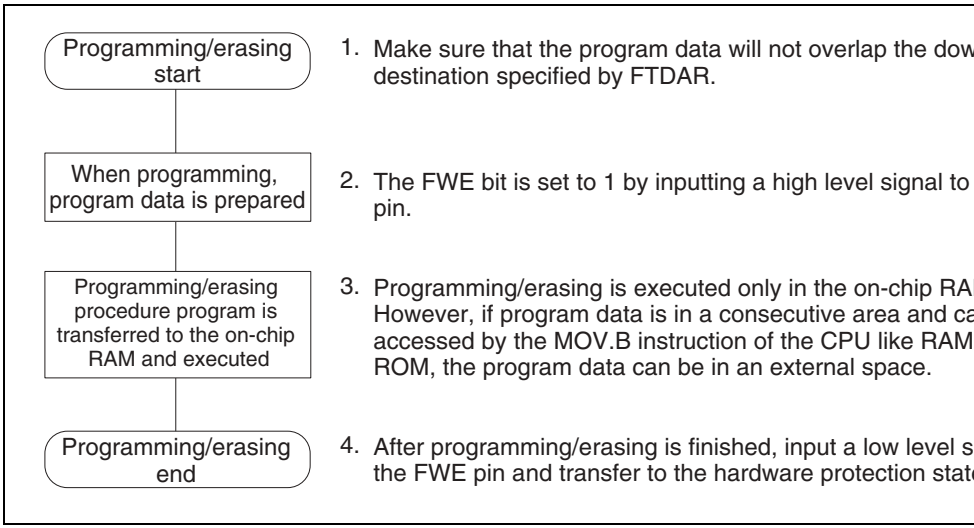


Figure 20.9 Programming/Erasing Overview Flow

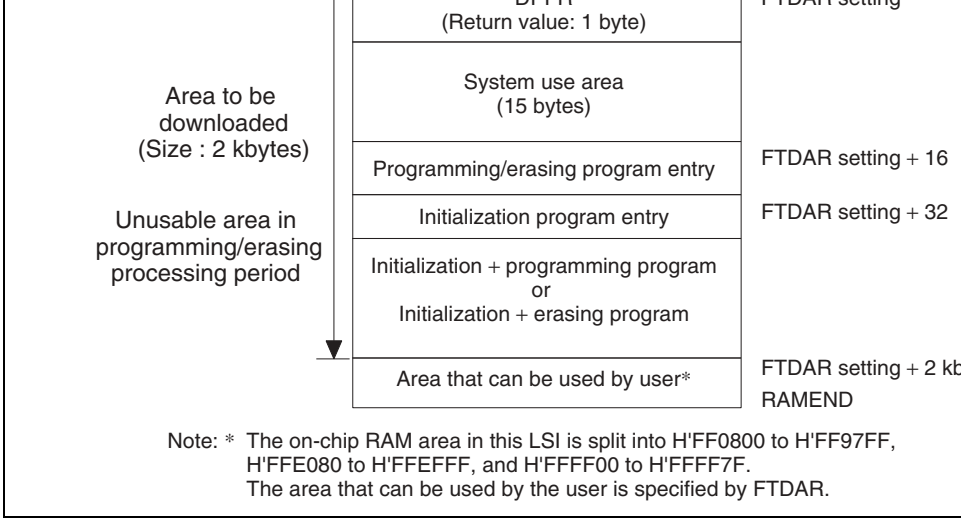


Figure 20.10 RAM Map When Programming/Erasing is Executed

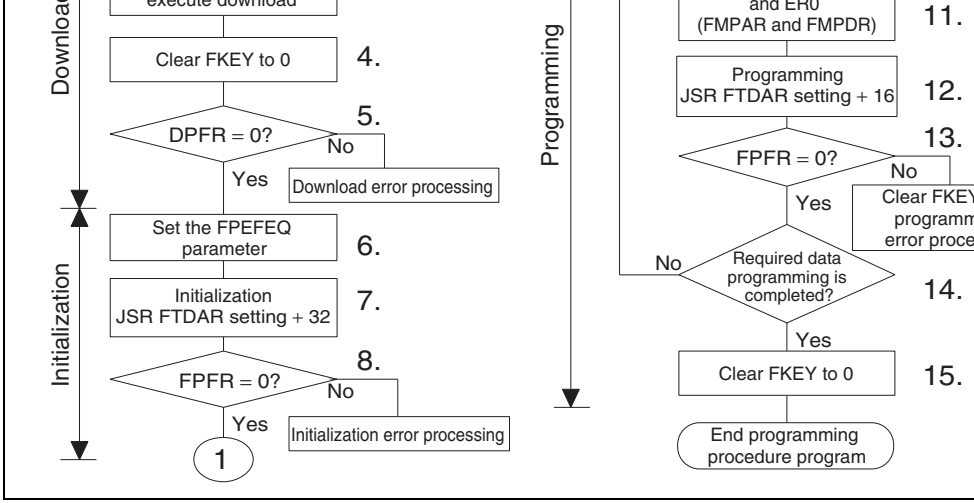


Figure 20.11 Programming Procedure

The procedure program must be executed in an area other than the flash memory to be programmed. Especially the part where the SCO bit in FCCS is set to 1 for downloading executed in the on-chip RAM.

The area that can be executed in the steps of the user procedure program (on-chip RAM, MAT, and external space) is shown in section 20.4.4, Procedure Program and Storable 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.

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
   set to 1, incorrect determination must be prevented by setting the one byte of the star
   (to be used as DPFR) specified by FTDAR to a value other than the return value (e.g

   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.
   — After the on-chip program storage area is returned to the user-MAT space, the us
     procedure program is returned.
   — In the download processing, the values of general registers of the CPU are held.

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 downloa destination specified by FTDAR). If the value is H'00, download has been perform 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 are set to the FPEFEQ parameters for initialization.

   — The current frequency of the CPU clock is set to the FPEFEQ parameter (general ER0).

     The settable range of the FPEFEQ parameter is 5 to 33 MHz. When the frequency 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 des in 20.3.2 (2) (a), Flash programming/erasing frequency parameter (FPEFEQ).

7. Initialization

   When a programming program is downloaded, the initialization program is also down the on-chip RAM. There is an entry point of the initialization program in the area fron address specified by FTDAR + 32 bytes of the on-chip RAM. The subroutine is called initialization is executed by using the following steps.

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 er
   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.

11. The parameter which is required for programming is set.
    The start address of the programming destination of the user MAT (FMPAR) is set t
    register ER1. The start address of the program data area (FMPDR) is set to general r
    ER0.

    — Example of the FMPAR setting
      FMPAR specifies the programming destination address. When an address other t
      the user MAT area is specified, even if the programming program is executed,
      programming is not executed and an error is returned to the return value paramet
      Since the unit is 128 bytes, the lower eight bits of the address must be H'00 or H'
      boundary of 128 bytes.

```
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.

15. After programming finishes, clear FKEY and specify software protection.
    If this LSI is restarted by a reset immediately after user MAT programming has finish secure the reset period (period of $\overline{RES}$ = 0) of 100 µs which is longer than normal.

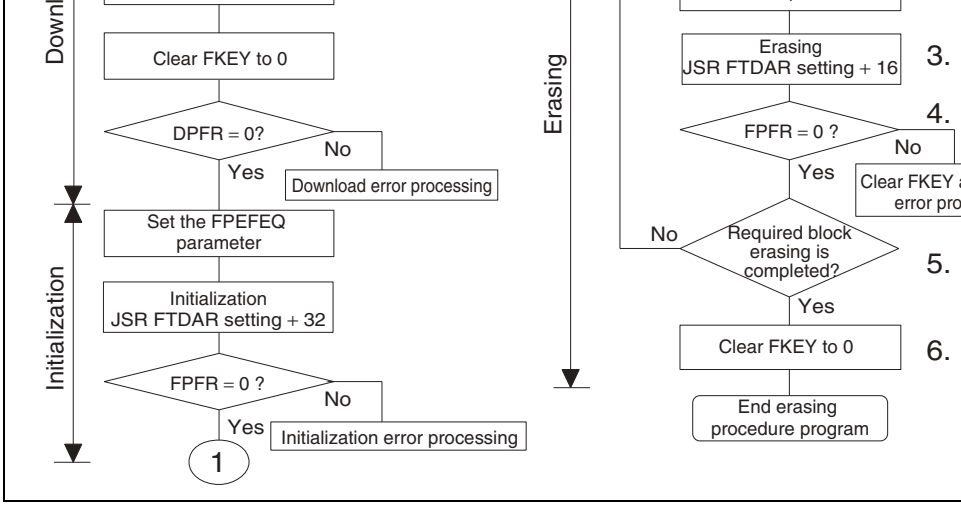


Figure 20.12 Erasing Procedure

The procedure program must be executed in an area other than the user MAT to be erase
Especially the part where the SCO bit in FCCS is set to 1 for downloading must be exec
on-chip RAM.

The area that can be executed in the steps of the user procedure program (on-chip RAM
MAT, and external space) is shown in section 20.4.4, Procedure Program and Storable A
Programming Data.

For the downloaded on-chip program area, refer to the RAM map for programming/eras
figure 20.10.

A single divided block is erased by one erasing processing. For block divisions, refer to
20.4. To erase two or more blocks, update the erase block number and perform the erasi
processing for each block.

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.

5. Determine whether erasure of the necessary blocks has completed.

   If more than one block is to be erased, update the FEBS parameter and repeat steps 2
   Blocks that have already been erased can be erased again.

6. After erasure completes, clear FKEY and specify software protection.

   If this LSI is restarted by a reset immediately after user MAT erasure has completed,
   the reset period (period of $\overline{RES}$ = 0) of 100 µs which is longer than normal.

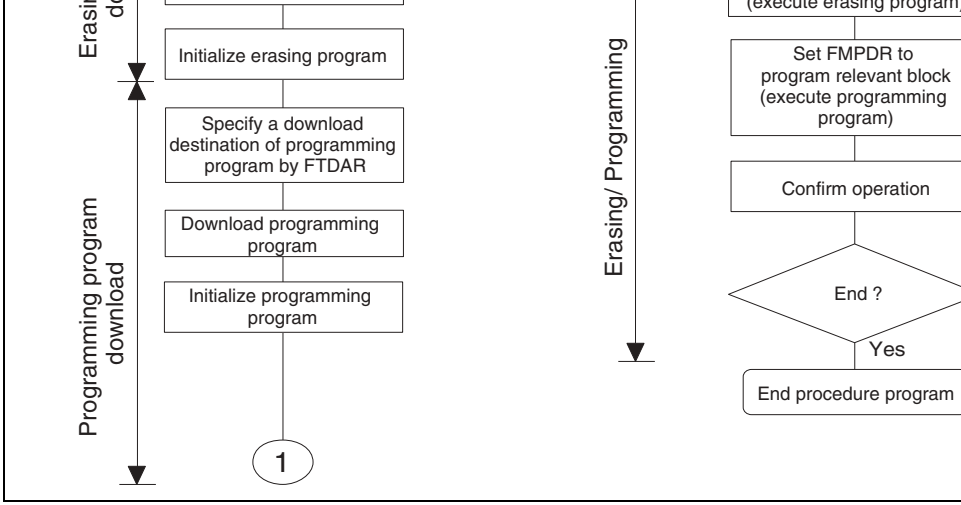


Figure 20.13 Repeating Procedure of Erasing and Programming

In the above procedure, download and initialization are performed only once at the begi

In this kind of operation, note the following:

- Be careful not to damage on-chip RAM with overlapped settings.
  In addition to the erasing program area and programming program area, areas for the procedure programs, work area, and stack area are reserved in on-chip RAM. Do not settings that will overwrite data in these areas.
- Be sure to initialize both the erasing program and programming program.
  Initialization by setting the FPEFEQ parameter must be performed for both the erasi program and the programming program. Initialization must be executed for both entr addresses: (download start address for erasing program) + 32 bytes and (download st address for programming program) + 32 bytes.

and user boot MAT states are checked by this check routine.

While the check routine is running, NMI and all other interrupts cannot be accepted.

Next, processing starts from the execution start address of the reset vector in the user boo
At this point, H′AA is set to FMATS because the execution MAT is the user boot MAT.

(2) User MAT Programming in User Boot Mode

For programming the user MAT in user boot mode, additional processing made by setting
are required: switching from user-boot-MAT selection state to user-MAT selection state,
switching back to user-boot-MAT selection state after programming completes.

Figure 20.14 shows the procedure for programming the user MAT in user boot mode.

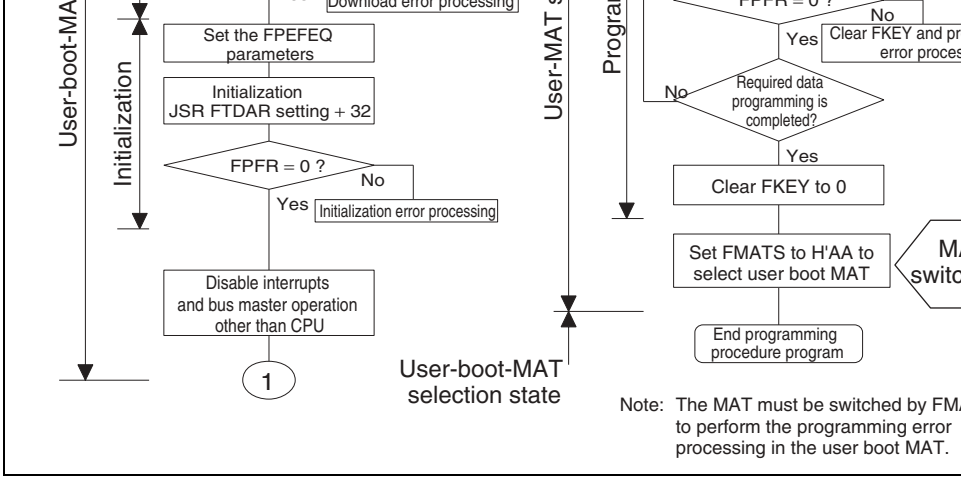


Figure 20.14 Procedure for Programming User MAT in User Boot Mode

The difference between the programming procedures in user program mode and user bo
whether the MAT is switched or not as shown in figure 20.14.

In user boot mode, the user boot MAT can be seen in the flash memory space with the u
hidden in the background. The user MAT and user boot MAT are switched only while th
MAT is being programmed. Because the user boot MAT is hidden while the user MAT
programmed, the procedure program must be located in an area other than flash memory
programming completes, switch the MATs again to return to the first state.

MAT switching is enabled by writing a specific value to FMATS. However note that wh
MATs are being switched, the LSI is in an unstable state, e.g. access to a MAT is not all
MAT switching is completed, and if an interrupt occurs, from which MAT the interrupt
read is undetermined. Perform MAT switching in accordance with the description in sec
Switching between User MAT and User Boot MAT.

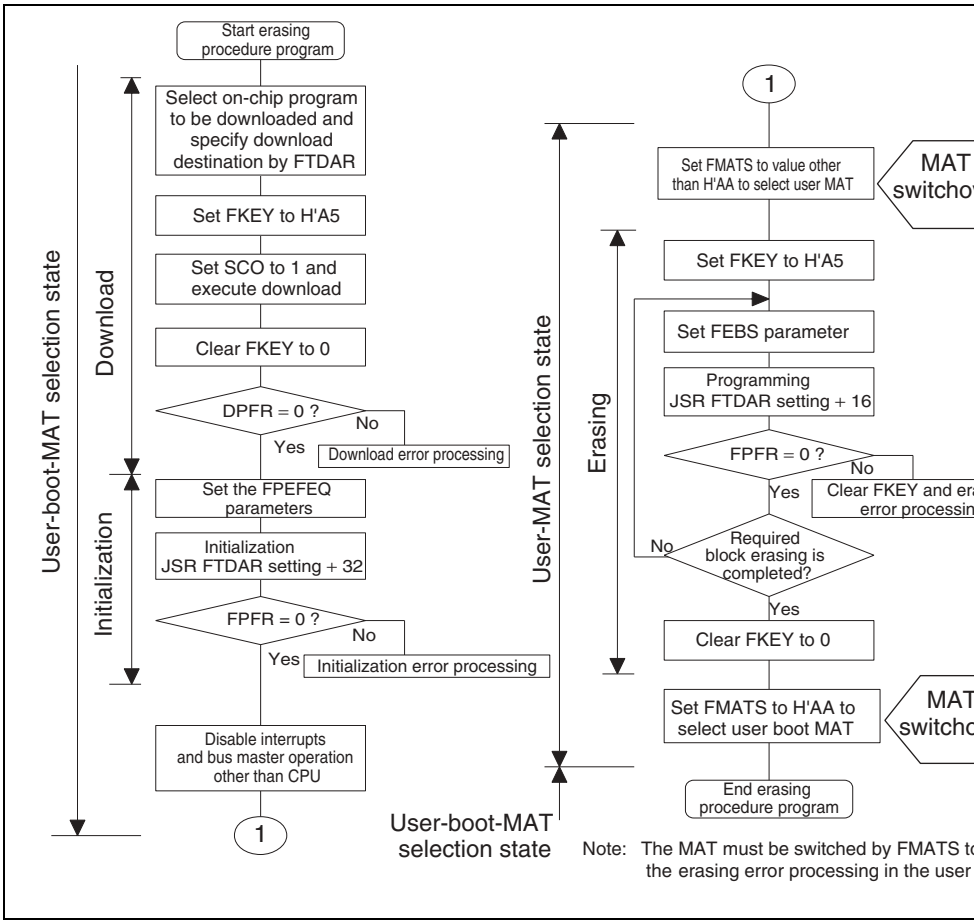


Figure 20.15 Procedure for Erasing User MAT in User Boot Mode

Programming Data.

20.4.4 Procedure Program and Storable Area for Programming Data

In the descriptions in the previous section, the programming/erasing procedure program storable areas for program data are assumed to be in the on-chip RAM. However, the pr the data can be stored in and executed from other areas, such as part of flash memory wh to be programmed or erased, or somewhere in the external address space.

(1) Conditions that Apply to Programming/Erasing

1. The on-chip programming/erasing program is downloaded from the address in the on RAM specified by FTDAR, therefore, this area is not available for use.
2. The on-chip programming/erasing program will use 128 bytes at the maximum as a s make sure that this area is secured.
3. Download by setting the SCO bit to 1 will lead to switching of the MAT. If, therefor 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, unt 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, I 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, therefor operation program is downloaded to the on-chip RAM to be executed. The NMI-han vector and programs such as that which activate the operation program, and NMI ha should thus be stored in on-chip memory other than flash memory or the external ad space.
6. After programming/erasing, the flash memory should be inhibited until FKEY is cle The reset state ($\overline{RES} = 0$) must be in place for more than 100 µs when the LSI mode to reset on completion of a programming/erasing operation.

In consideration of these conditions, there are three factors; operating mode, the bank stru
the user MAT, and operations.

The areas in which the programming data can be stored for execution are shown in tables

Table 20.7 Executable MAT

	Initiated Mode	
Operation	**User Program Mode**	**User Boot Mode***
Programming	Table 20.8 (1)	Table 20.8 (3)
Erasing	Table 20.8 (2)	Table 20.8 (4)

Note: * Programming/Erasing is possible to user MATs.

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	○	×	○	○

Note: Transferring the data to the on-chip RAM enables this area to be used.

SCO = 1 to FCOS (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	○	×	○	○

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 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 us

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 Settings of Erasure Parameter	○	×	○	○	

FMATS

Note: * Switching FMATS by a program in the on-chip RAM enables this area to be us

Table 25.5 Hardware Protection

Item	Description	Function to be Pro: Download	Function to be Pro: Progr
FWE pin protection	• When a low level signal is input to the FWE pin, the FWE bit in FCCS is cleared and the program/erase-protected state is entered.	—	○
Reset/standby protection	• The program/erase interface registers are initialized in the reset state (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.	○	○

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.	○	○

20.5.3 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 me
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.
4. When a bus master other than the CPU, such as the DTC, gets bus mastership during
   programming/erasing.

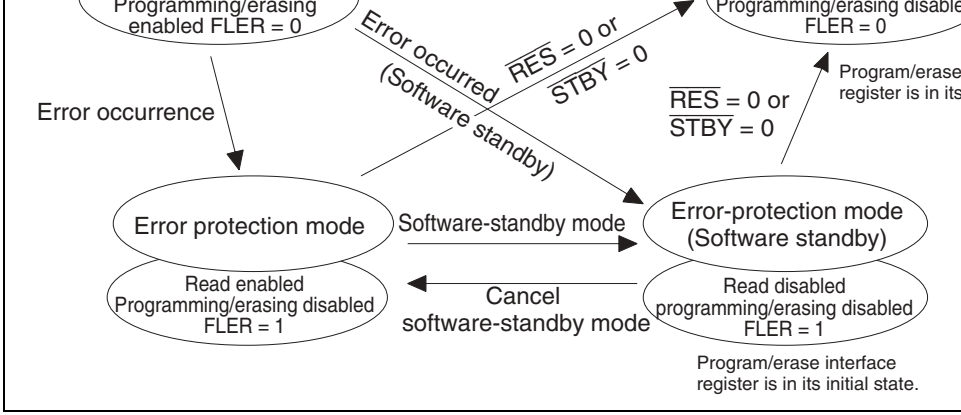


Figure 20.16 Transitions to Error-Protection State

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 8-kbyte memory space. If acc beyond the 8-kbyte space, the values read are undefined.

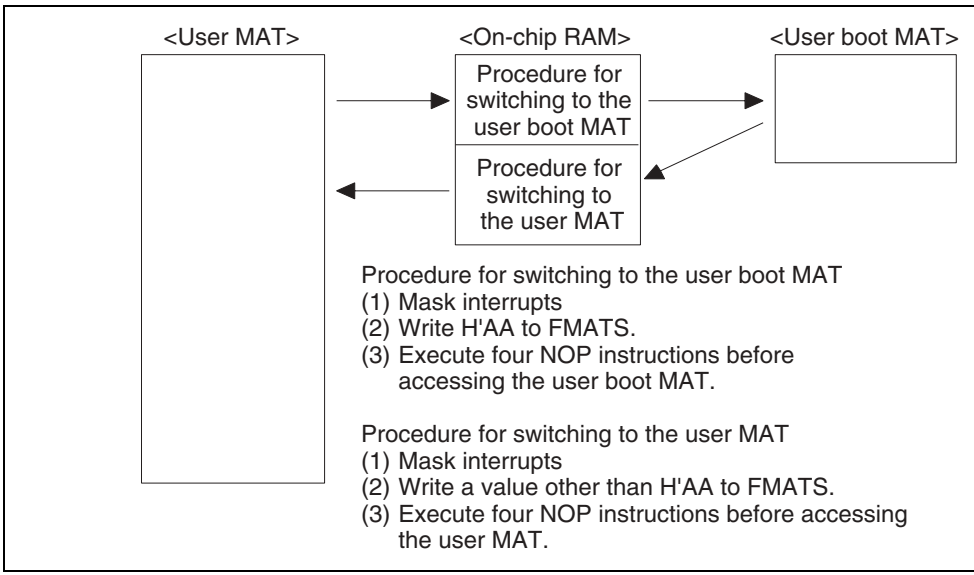


Figure 20.17 Switching between the User MAT and User Boot MAT

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.

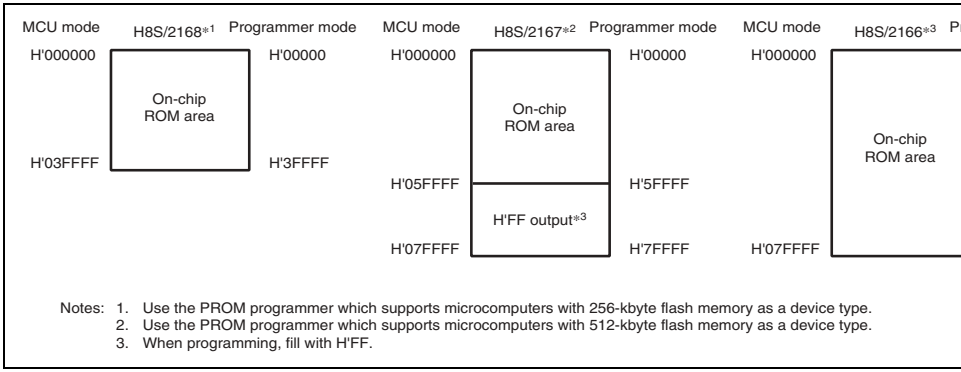


Figure 20.18 Memory Map in Programmer Mode

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 20.19.

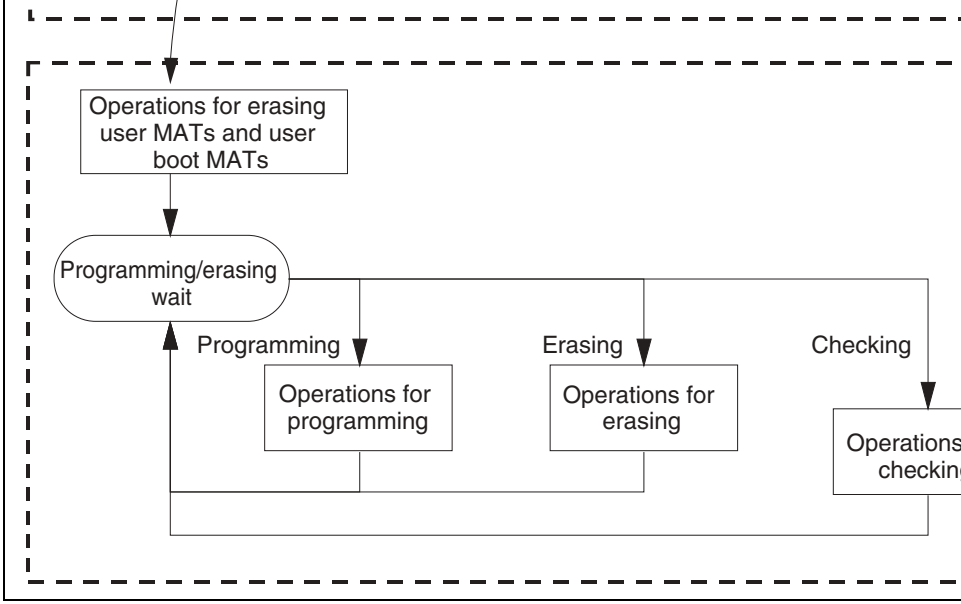


Figure 20.19 Boot Program States

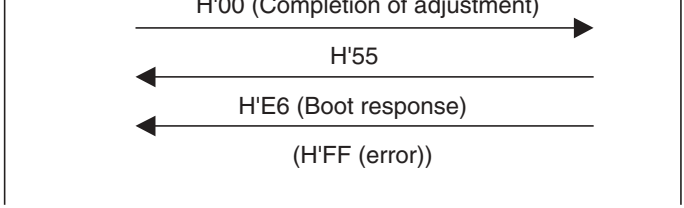


Figure 20.20 Bit-Rate-Adjustment Sequence

(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.
4. Programming of 128 bytes

   The size is not specified in commands. The size of n is indicated in response to the programming unit inquiry.
5. Memory read response

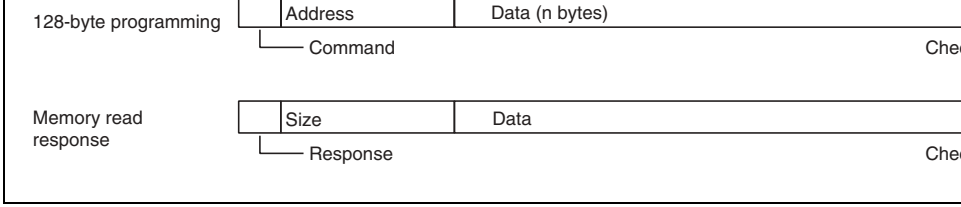


Figure 20.21 Communication Protocol Format

- 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.)
- Size (4 bytes): 4-byte response to a memory read

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

The selection commands, which are device selection (H'10), clock mode selection (H'11), bit rate selection (H'3F), should be sent from the host in that order. These commands will be needed. When two or more selection commands are sent at once, the last command wi valid.

All of these commands, except for the boot program status inquiry command (H'4F), will until the boot program receives the programming/erasing transition (H'40). The host can

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.

(b) Device Selection

The boot program will set the supported device to the specified device code. The progra
return the selected device code in response to the inquiry after this setting has been mad

Command	H'10	Size	Device code	SUM

- Command, H'10, (1 byte): Device selection
- Size (1 byte): Amount of device-code data
  This is fixed at 2
- Device code (4 bytes): Device code (ASCII code) returned in response to the suppor
  inquiry
- SUM (1 byte): Checksum

The boot program will retain 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

(d) Clock Mode Selection

The boot program will set the specified clock mode. The program will return the selected mode information after this setting has been made.

The clock-mode selection command should be sent after the device-selection commands.

Command	H'11	Size	Mode	SUM

- Command, H'11, (1 byte): Selection of clock mode
- Size (1 byte): Amount of data that represents the modes
- Mode (1 byte): A clock mode returned in reply to the supported clock mode inquiry.
- 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.

Command	H'22

- Command, H'22, (1 byte): Inquiry regarding multiplication ratio

Response	H'32	Size	Number of types				
	Number of multiplication ratios	Multiplica-tion ratio	···				
	...						
	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 cl number of types will be H'02.)
- Number of multiplication ratios (1 byte): The number of multiplication ratios for eac (e.g. the number of multiplication ratios to which the main clock can be set and the p clock can be set.)
- Multiplication ratio (1 byte)

  Multiplication ratio: The value of the multiplication ratio (e.g. when the clock-freque 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 th 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 multiplica and as many groups of data are returned as there are types.
- SUM (1 byte): Checksum

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 va the number of frequencies.
- Number of operating clock frequencies (1 byte): The number of supported operating c frequency types
  (e.g. when there are two operating clock frequency types, which are the main and peri 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 hundred of MHz, and multiplied by 100. (e.g. when the value is 20.00 MHz, it will be 2000, w H'07D0.)
- Maximum value (2 bytes): Maximum value among the multiplied or divided clock fre
  There are as many pairs of minimum and maximum values as there are operating cloc frequencies.
- SUM (1 byte): Checksum

- Size (1 byte): The number of bytes that represents the number of areas, area-start add 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 ar
- 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 add 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
- Area-last address (4 bytes): Last address of the area
  There are as many groups of data representing the start and last addresses as there ar
- SUM (1 byte): Checksum

- Size (three bytes): The number of bytes that represents the number of blocks, block-st 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 are
- 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 fixed
- Programming unit (2 bytes): A unit for programming
  This is the unit for reception of programming.
- SUM (1 byte): Checksum

- 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 i
  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 whi
  device can be set.
- Multiplication ratio 1 (1 byte) : The value of multiplication or division ratios for th
  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 th
  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.

Error Response

H'BF	ERROR

- Error response, H'BF, (1 byte): Error response to selection of new bit rate

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 o minimum to maximum frequencies which matches the clock modes of the specified devic 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 it the clock modes of the specified device. When the value is out of this range, an multiplica 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 ope 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 range minimum to maximum frequencies which are available with the clock modes of the speci device. When it is out of this range, an operating frequency error is generated.

4. Bit rate

To facilitate error checking, the value (n) of clock select (CKS) in the serial mode registe and the value (N) in the bit rate register (BRR), which are found from the peripheral oper clock frequency (ϕ) and bit rate (B), are used to calculate the error rate to ensure that it is

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 20.22.

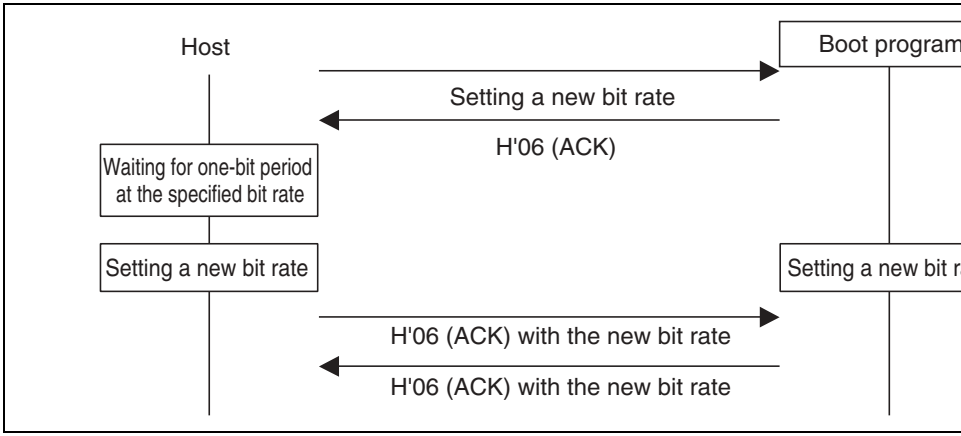


Figure 20.22 New Bit-Rate Selection Sequence

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 been
  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 inc
or a command is unacceptable. Issuing a clock-mode selection command before a device
or an inquiry command after the transition to programming/erasing state command, are ex

Error Response	H'80	H'xx

- Error response, H'80, (1 byte): Command error
- Command, H'xx, (1 byte): Received command

(8) Command Order

The order for commands in the inquiry selection state is shown below.

1. A supported device inquiry (H'20) should be made to inquire about the supported dev
2. The device should be selected from among those described by the returned informatio
   with a device-selection (H'10) command.
3. A clock-mode inquiry (H'21) should be made to inquire about the supported clock mo
4. The clock mode should be selected from among those described by the returned infor
   and set.

(3) Programming/Erasing State

A programming selection command makes the boot program select the programming me
128-byte programming command makes it program the memory with data, and an erasin
selection command and block erasing command make it erase the block. The
programming/erasing commands are listed below.

Table 20.12 Programming/Erasing Command

Command	Command Name	Description
H'42	User boot MAT programming selection	Transfers the user boot MAT progr program
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

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 20.23.

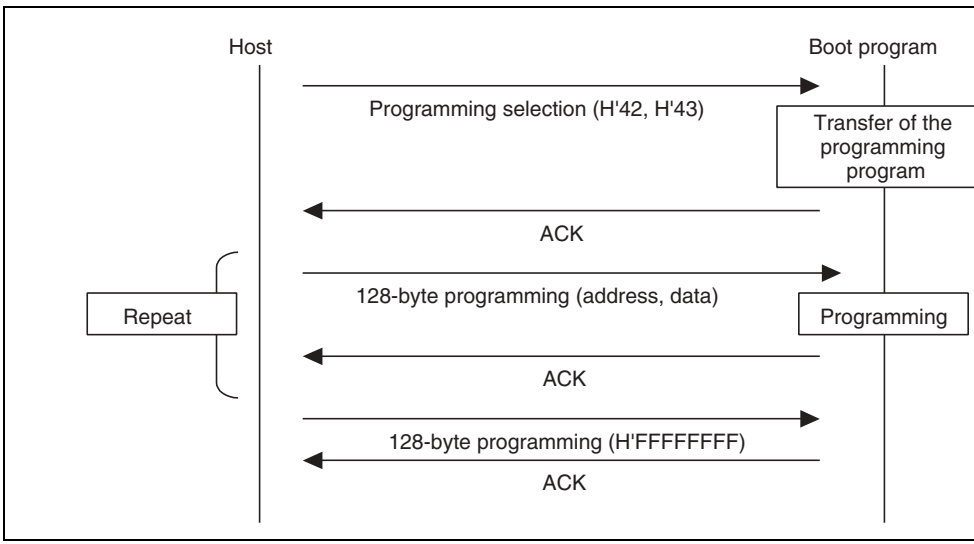


Figure 20.23 Programming Sequence

- 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 s
program the user boot MATs or user MATs in response to 128-byte programming.

Command	H'50	Address						
	Data	...						
	...							
	SUM							

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 continu

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.

Error Response	H'D0	ERROR

- Error Response, H'D0, (1 byte): Error response for 128-byte programming

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 are figure 20.24.

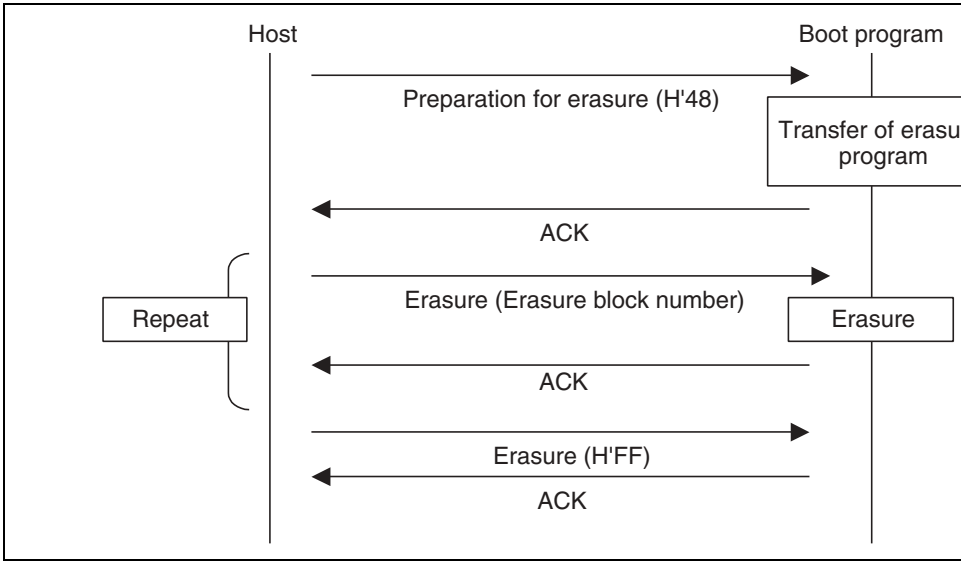


Figure 20.24 Erasure Sequence

- 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.

On receiving block number H'FF, the boot program will stop erasure and wait for a select 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

Error Response

H'D2	ERROR

- Error response: H'D2 (1 byte): Error response to memory read

- 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

(14) User Boot MAT Blank Check

The boot program will check whether or not all user boot MATs are blank and return the

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.

(16) Boot Program State Inquiry

The boot program will return indications of its present state and error condition. This inc
be made in the inquiry/selection state or the programming/erasing state.

Command	H'4F

- Command, H'4F, (1 byte): Inquiry regarding boot program's state

Response	H'5F	Size	Status	ERROR	SUM

- Response, H'5F, (1 byte): Response to boot program state inquiry
- Size (1 byte): The number of bytes. This is fixed to 2.
- 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 20.14 Error Code

Code	Description
H'00	No Error
H'11	Sum Check Error
H'12	Program Size Error
H'21	Device Code Mismatch Error
H'22	Clock Mode Mismatch Error
H'24	Bit Rate Selection Error
H'25	Input Frequency Error
H'26	Multiplication Ratio Error
H'27	Operating Frequency Error
H'29	Block Number Error
H'2A	Address Error
H'2B	Data Length Error
H'51	Erasure Error
H'52	Erasure Incomplete Error
H'53	Programming Error
H'54	Selection Processing Error
H'80	Command Error
H'FF	Bit-Rate-Adjustment Confirmation Error

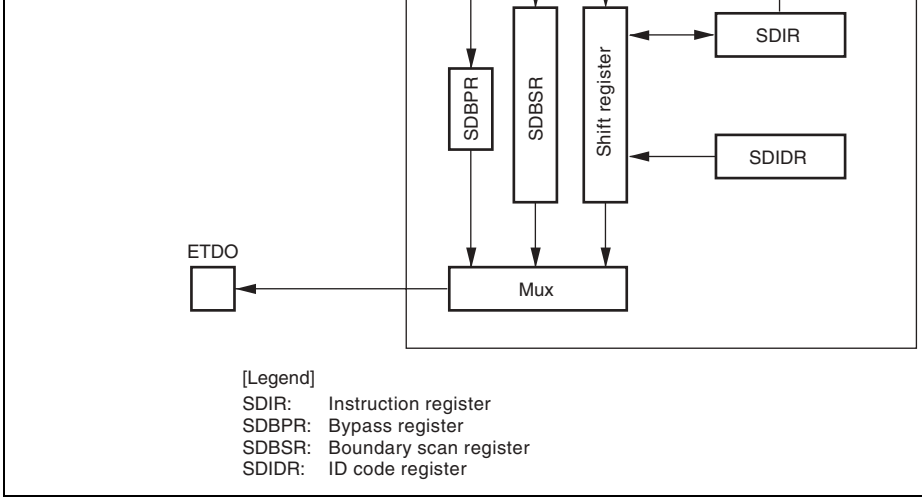
only the specified socket adapter. If other adapters are used, the product may be dam

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. executed accidentally, reset must be released after the reset input period of 100 µs w 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 finished, secure the reset period (period of $\overline{RES}$ = 0) of more than 100 µs. Though tr the reset state or hardware standby state during programming/erasing is prohibited, i executed accidentally, reset must be released after the reset input period of 100 µs w 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 pov power-on caused by a power failure and other factors.
8. Program the area with 128-byte programming-unit blocks in on-board programming 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 programmin performed after execution of automatic erasure.
10. To write data or programs to the flash memory, data or programs must be allocated t addresses higher than that of the external interrupt vector table (H'000040) and H'FF written to the areas that are reserved for the system in the exception handling vector
11. If data other than H'FFFFFFFF is written to the key code area (H'00003C to H'00003 flash memory, only H'00 can be read in programmer mode. (In this case, data is read Rewrite is possible after erasing the data.) For reading in programmer mode, make s write H'FFFFFFFF to the entire key code area. If data other than H'FF is to be writte

microcomputer which does not support download of the on-chip program by a SCO t 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 availa runaway by WDT during programming/erasing. Prepare countermeasures (e.g. use of periodic timer interrupts) for WDT with taking the programming/erasing time into consideration as required.

- 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 21.1 JTAG Block Diagram

			Characteristics. If there is no input, the is fixed to 1 by an internal pull-up.
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. If there is no input, the ETM fixed to 1 by an internal pull-up.
Test data input	ETDI	Input	Serial data input Performs serial input of instructions an JTAG registers. ETDI is sampled on th the ETCK pin. If there is no input, the E fixed to 1 by an internal pull-up.
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. If no input, the $\overline{\text{ETRST}}$ pin is fixed to 1 by internal pull-up.

CLAMP, or HIGHZ mode. The boundary scan register (SDBSR) is a 334-bit register to v
ETDI and ETDO pins are connected in SAMPLE/PRELOAD or EXTEST mode. The ID
register (SDIDR) is a 32-bit register; a fixed code can be output via the ETDO pin in IDC
mode. All registers cannot be accessed directly by the CPU.

Table 21.2 shows the kinds of serial transfer possible with each JTAG register.

Table 21.2 JTAG Register Serial Transfer

Register	Serial Input	Serial Output
SDIR	Possible	Possible
SDBPR	Possible	Possible
SDBSR	Possible	Possible
SDIDR	Impossible	Possible

29 28	TS1 TS0	1 0	R/W R/W	0000: EXTEST mode 0001: Setting prohibited 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 cann modified.
13	—	1	R	Reserved This bit is always read as 1 and cannot be
12	—	0	R	Reserved This bit is always read as 0 and cannot be
11	—	1	R	Reserved This bit is always read as 1 and cannot be
10 to 1	—	All 0	R	Reserved These bits are always read as 0 and cann modified.
0	—	1	R	Reserved This bit is always read as 1 and cannot be

				1110: IDCODE mode (Initial value) 1111: BYPASS mode
27 to 14	—	All 0	R	Reserved These bits are always read as 0 and cannot modified.
13	—	1	R	Reserved This bit is always read as 1 and cannot be r
12	—	0	R	Reserved This bit is always read as 0 and cannot be r
11, 10	—	All 1	R	Reserved These bits are always read as 1 and cannot modified.
9	—	0	R	Reserved This bit is always read as 0 and cannot be r
8	—	1	R	Reserved This bit is always read as 1 and cannot be r
7 to 1	—	All 0	R	Reserved These bits are always read as 0 and cannot modified.
0	—	1	R	Reserved This bit is always read as 1 and cannot be r

register.

5	P56	Input	324
		Enable	323
		Output	322
6	P57	Input	321
		Enable	320
		Output	319
9	MD1	Input	318
10	MD0	Input	317
11	NMI	Input	316
14	$\overline{\text{MD2}}$	Input	315
15	P51	Input	314
		Enable	313
		Output	312
16	P50	Input	311
		Enable	310
		Output	309
17	P97	Input	308
		Enable	307
		Output	306
18	P96	Input	305
		Enable	304
		Output	303

23	P91	Input	290
		Enable	289
		Output	288
24	P90	Input	287
		Enable	286
		Output	285
25	PC7	Input	284
		Enable	283
		Output	282
26	PC6	Input	281
		Enable	280
		Output	279
27	PC5	Input	278
		Enable	277
		Output	276
28	PC4	Input	275
		Enable	274
		Output	273
29	PC3	Input	272
		Enable	271
		Output	270
30	PC2	Input	269
		Enable	268
		Output	267
31	PC1	Input	266
		Enable	265
		Output	264

37	PA4	Input	251
		Enable	250
		Output	249
38	PA3	Input	248
		Enable	247
		Output	246
39	PA2	Input	245
		Enable	244
		Output	243
40	PA1	Input	242
		Enable	241
		Output	240
41	PA0	Input	239
		Enable	238
		Output	237
43	P87	Input	236
		Enable	235
		Output	234
44	P86	Input	233
		Enable	232
		Output	231
45	P85	Input	230
		Enable	229
		Output	228
46	P84	Input	227
		Enable	226
		Output	225

51	PE7	Input	212
		Enable	211
		Output	210
52	PE6	Input	209
		Enable	208
		Output	207
53	PE5	Input	206
		Enable	205
		Output	204
54	PE4	Input	203
		Enable	202
		Output	201
55	PE3	Input	200
		Enable	199
		Output	198
56	PE2	Input	197
		Enable	196
		Output	195
57	PE1	Input	194
		Enable	193
		Output	192
58	PE0	Input	191
		Enable	190
		Output	189
59	PD7	Input	188
		Enable	187
		Output	186

64	PD2	Input	173
		Enable	172
		Output	171
65	PD1	Input	170
		Enable	169
		Output	168
66	PD0	Input	167
		Enable	166
		Output	165
68	P70	Input	164
69	P71	Input	163
70	P72	Input	162
71	P73	Input	161
72	P74	Input	160
73	P75	Input	159
74	P76	Input	158
75	P77	Input	157
78	P60	Input	156
		Enable	155
		Output	154
79	P61	Input	153
		Enable	152
		Output	151
80	P62	Input	150
		Enable	149
		Output	148

85	P67	Input	135
		Enable	134
		Output	133
92	PF2	Input	132
		Enable	131
		Output	130
93	PF1	Input	129
		Enable	128
		Output	127
94	PF0	Input	126
		Enable	125
		Output	124
96	P27	Input	123
		Enable	122
		Output	121
97	P26	Input	120
		Enable	119
		Output	118
98	P25	Input	117
		Enable	116
		Output	115
99	P24	Input	114
		Enable	113
		Output	112
100	P23	Input	111
		Enable	110
		Output	109

105	P16	Input	96
		Enable	95
		Output	94
106	P15	Input	93
		Enable	92
		Output	91
107	P14	Input	90
		Enable	89
		Output	88
108	P13	Input	87
		Enable	86
		Output	85
109	P12	Input	84
		Enable	83
		Output	82
110	P11	Input	81
		Enable	80
		Output	79
112	P10	Input	78
		Enable	77
		Output	76
113	PB7	Input	75
		Enable	74
		Output	73
114	PB6	Input	72
		Enable	71
		Output	70

119	PB1	Input	57
		Enable	56
		Output	55
120	PB0	Input	54
		Enable	53
		Output	52
121	P30	Input	51
		Enable	50
		Output	49
122	P31	Input	48
		Enable	47
		Output	46
123	P32	Input	45
		Enable	44
		Output	43
124	P33	Input	42
		Enable	41
		Output	40
125	P34	Input	39
		Enable	38
		Output	37
126	P35	Input	36
		Enable	35
		Output	34
127	P36	Input	33
		Enable	32
		Output	31

132	P43	Input	18
		Enable	17
		Output	16
133	P52	Input	15
		Enable	14
		Output	13
134	P53	Input	12
		Enable	11
		Output	10
135	FWE	Input	9
136	P54	Input	8
		Enable	7
		Output	6
137	P55	Input	5
		Enable	4
		Output	3
138	P44	Input	2
		Enable	1
		Output	0
		to ETDO	

Note: The enable signals are active-high. When an enable signal is driven high, the corresponding pin is driven with the output value.

- H8S/2167, H8S/2166

31 28	27 12	11 1	
0000	0000 0011 0000 0010	0000 0000 111	
Version (4 bits)	Part Number (16 bits)	Manufacture Identify (11 bits)	F

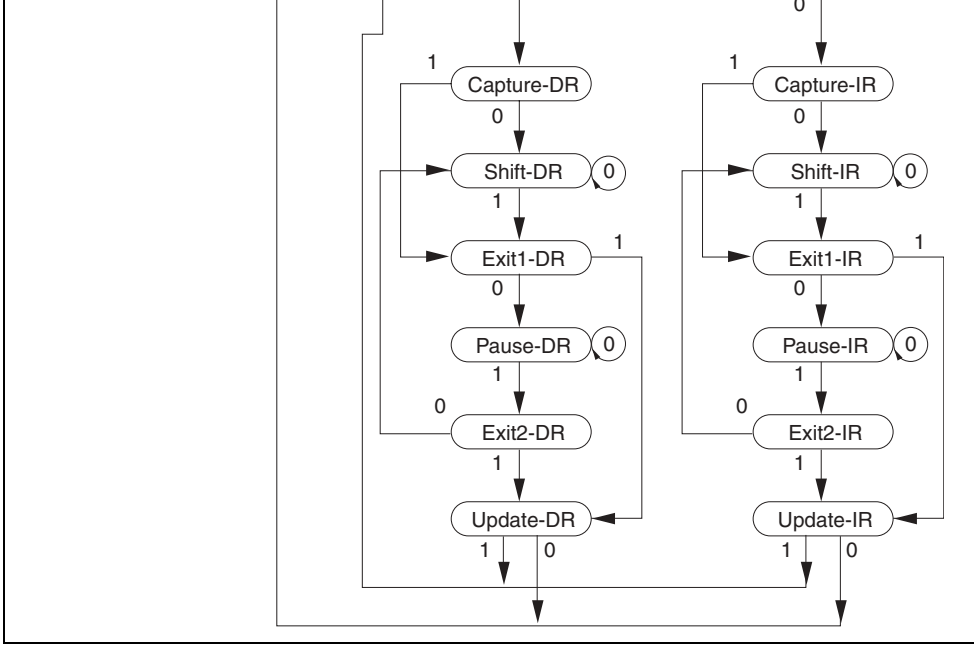


Figure 21.2 TAP Controller State Transitions

21.5.1 Supported Instructions

This LSI supports the three essential instructions defined in the IEEE1149.1 standard (B
SAMPLE/PRELOAD, and EXTEST) and optional instructions (CLAMP, HIGHZ, and I

BYPASS: Instruction code: B'1111

The BYPASS instruction is an instruction that operates the bypass register. This instruc
shortens the shift path to speed up serial data transfer involving other chips on the printe
board. While this instruction is being executed, the test circuit has no effect on the syste
circuits.

SAMPLE/PRELOAD: Instruction code: B'0100

The SAMPLE/PRELOAD instruction inputs values from this LSI internal circuitry to th
boundary scan register, outputs values from the scan path, and loads data onto the scan p
When this instruction is being executed, this LSI's input pin signals are transmitted direc
internal circuitry, and internal circuit values are directly output externally from the outpu
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
circuitry, or a value to be transferred from the internal circuitry to an output pin, is latch
boundary scan register and read from the scan path. Snapshot latching does not affect no
operation of this LSI.

In a PRELOAD operation, an initial value is set in the parallel output latch of the bound
register from the scan path prior to the EXTEST instruction. Without a PRELOAD ope
when the EXTEST instruction was executed an undefined value would be output from th
pin until completion of the initial scan sequence (transfer to the output latch) (with the E
instruction, the parallel output latch value is constantly output to the output pin).

When the CLAMP instruction is enabled, the output pin outputs the value of the boundary register that has been previously set by the SAMPLE/PRELOAD instruction. While the C instruction is enabled, the state of the boundary scan register maintains the previous state regardless of the state of the TAP controller.

A bypass register is connected between the ETDI and ETDO pins. The related circuit ope the same way when the BYPASS instruction is enabled.

HIGHZ: Instruction code: B'0011

When the HIGHZ instruction is enabled, all output pins enter a high-impedance state. Wh HIGHZ instruction is enabled, the state of the boundary scan register maintains the previo regardless of the state of the TAP controller.

A bypass register is connected between the ETDI and ETDO pins. The related circuit ope the same way when the BYPASS instruction is enabled.

IDCODE: Instruction code: B'1110

When the IDCODE instruction is enabled, the value of the ID code register is output from ETDO pin with LSB first when the TAP controller is in the Shift-DR state. While the IDC instruction is being executed, the test circuit does not affect the system circuit.

When the TAP controller is in the Test-Logic-Reset state, the instruction register is initial the IDCODE instruction.

— Alternatively, to prevent the $\overline{\text{ETRST}}$ pin of the board tester from being affected by system reset, circuits must be separated.

Figure 21.3 shows a design example of the reset signal circuit wherein no reset signal interference occurs.

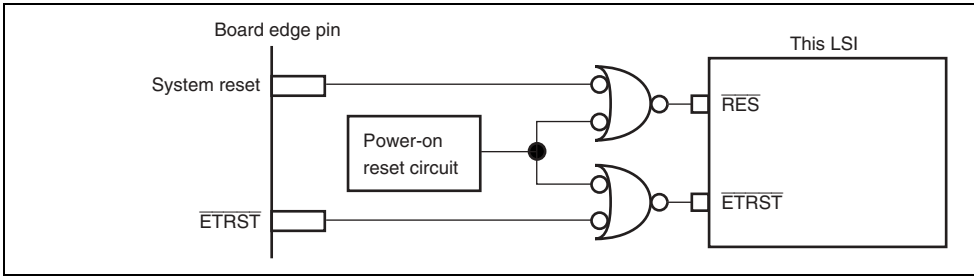


Figure 21.3 Reset Signal Circuit Without Reset Signal Interference

3. The registers are not initialized in standby mode. If the $\overline{\text{ETRST}}$ pin is set to 0 in stand IDCODE mode will be entered.
4. The frequency of the ETCK pin must be lower than that of the system clock. For deta section 25, Electrical Characteristics.
5. Data input/output in serial data transfer starts from the LSB. Figure 21.4 and 21.5 sho examples of serial data input/output.
6. When data that exceeds the number of bits of the register connected between the ETD ETDO pins is serially transferred, the serial data that exceeds the number of register b output from the ETDO pin is the same as that input from the ETDI pin.
7. If the JTAG serial transfer sequence is disrupted, the $\overline{\text{ETRST}}$ pin must be reset. Trans 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.

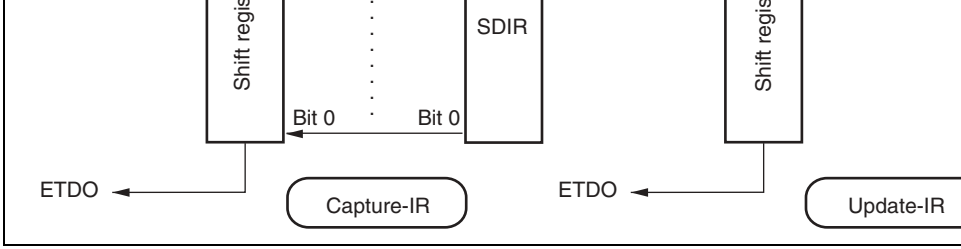


Figure 21.4 Serial Data Input/Output (1)

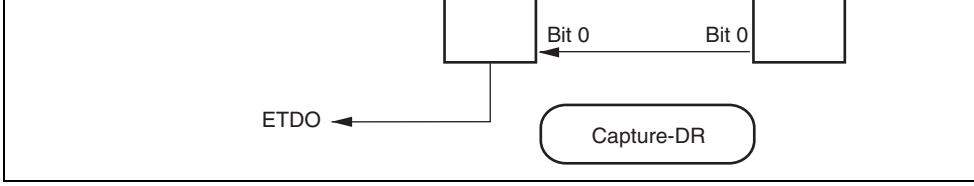


Figure 21.5 Serial Data Input/Output (2)

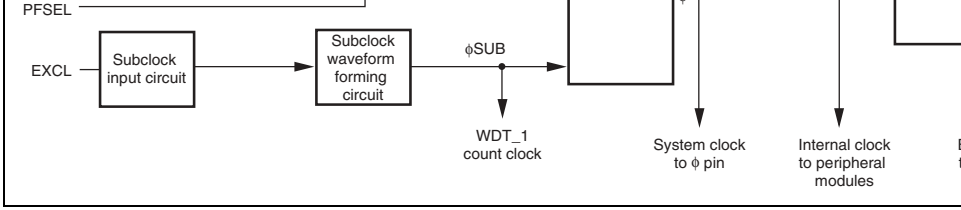


Figure 22.1 Block Diagram of Clock Pulse Generator

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 c
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 23.1.1, 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 23.1.2, Low-P
Control Register (LPWRCR).

characteristics given in table 22.2 should be used.

When PFSEL is high, the system clock (ϕ) frequency should be no more than 25 MHz an
crystal resonator with frequency identical to that of the system clock (ϕ) should be used.
PFSEL is low, a crystal resonator with ¼ times the frequency of the system clock (ϕ) sho
used.

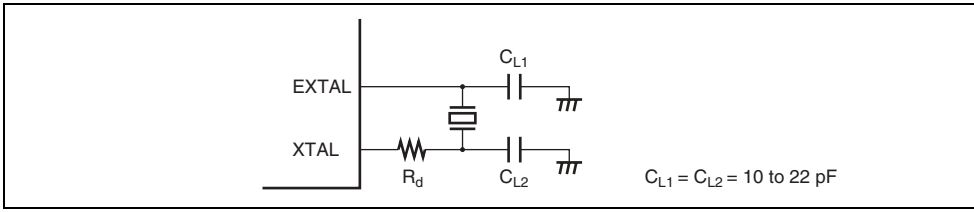


Figure 22.2 Typical Connection to Crystal Resonator

Table 22.1 Damping Resistance Values

Frequency (MHz)	5	8	10	12	16	20	
R_d (Ω)	300	200	0	0	0	0	

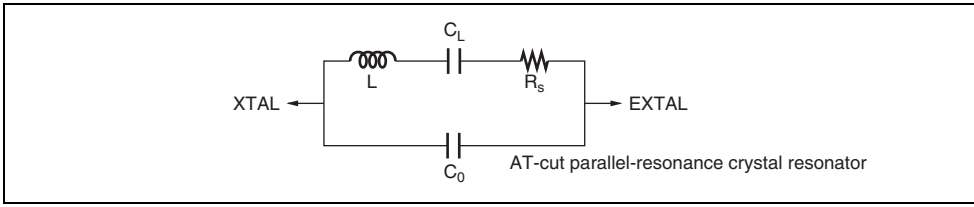


Figure 22.3 Equivalent Circuit of Crystal Resonator

mode, subactive mode, subsleep mode, and watch mode. The frequency of the external c
should be the same as that of the system clock (ϕ) when PFSEL is high. When PFSEL is
external clock of 1/4 times the frequency of the system clock (ϕ) should be used.

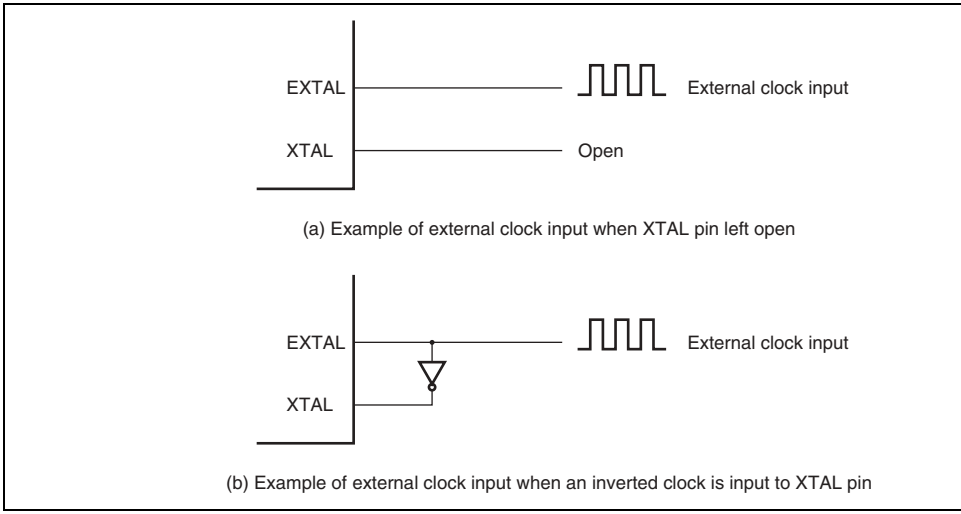


Figure 22.4 Example of External Clock Input

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 table 25.5 ar
25.8.

22.3 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.

22.4 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.

22.5 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 P96DDR P9DDR 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.

22.6 Subclock Waveform Forming 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.

The subclock is not sampled in subactive mode, subsleep mode, or watch mode.

22.8 Usage Notes

22.8.1 Note on Resonator

Since all kinds of characteristics of the resonator are closely related to the board design l
user, use the example of resonator connection in this document for only reference; be su
an resonator that has been sufficiently evaluated by the user. Consult with the resonator
manufacturer about the resonator circuit ratings which vary depending on the stray capa
the resonator and installation circuit. Make sure the voltage applied to the oscillation pin
exceed the maximum rating.

22.8.2 Notes on Board Design

When using a crystal resonator, the crystal resonator and its load capacitors should be pl
close as possible to the EXTAL and XTAL pins. Other signal lines should be routed aw
the oscillation circuit to prevent inductive interference with the correct oscillation as sho
figure 22.5.

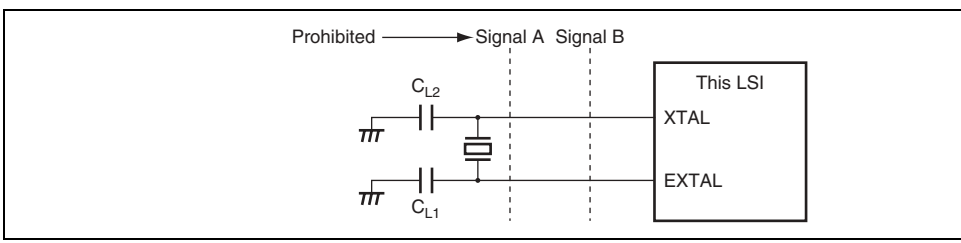


Figure 22.5 Note on Board Design of Oscillation Circuit Section

- Sleep mode

  The CPU stops but on-chip peripheral modules continue operating.
- Subsleep mode

  The CPU and on-chip peripheral modules other than TMR_0, TMR_1, WDT_0, and
  stop operating.
- Watch mode

  The CPU and on-chip peripheral modules other than WDT_1 stop 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.

- Sub-chip module stop control register AH, AL (SUBMSTPAH, SUBMSTPAL)

23.1.1 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 e 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, subactive mod watch mode When the SLEEP instruction is executed in subact mode: 0: Shifts to subsleep mode 1: Shifts to watch mode or high-speed mode Note that the SSBY bit is not changed even if a m transition occurs by an interrupt.

				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 clock The operating clock is changed when a DTC tran requested even if the CPU operates based on the speed clock.
2	SCK2	0	R/W	System Clock Select 2 to 0 Select a clock for the bus master in high-speed m medium-speed mode. When making a transition to subactive mode or w mode, SCK2 to SCK0 must be cleared to 0. 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 11*: Must not be set.
1	SCK1	0	R/W	
0	SCK0	0	R/W	

[Legend]

*: Don't care

Recommended specification

Note: Setting prohibited.

23.1.2 Low-Power Control Register (LPWRCR)

LPWRCR controls power-down modes and signals in the multiplex bus extended mode.

Bit	Bit Name	Initial Value	R/W	Description
7	DTON	0	R/W	Direct Transfer On Flag Specifies the operating mode to be entered after execut SLEEP instruction. When the SLEEP instruction is executed in high-speed r medium-speed mode: 0: Shifts to sleep mode, software standby mode, or wat 1: Shifts directly to subactive mode, or shifts to sleep m software standby mode When the SLEEP instruction is executed in subactive m 0: Shifts to subsleep mode or watch mode 1: Shifts directly to high-speed mode, or shifts to subsl

				1: Shifts to subsleep mode or watch mode When watch mode is cancelled: 0: Shifts to high-speed mode 1: Shifts to subactive mode
5	NESEL	0	R/W	Noise Elimination Sampling Frequency Select Selects the frequency by which the subclock (ϕSUB) inp the EXCL pin is sampled using the clock (ϕ) generated b 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 pin. 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 ($\overline{\text{CS256}}$ $\overline{\text{IOS}}$) in the address multiplex extended mode. 0: Outputs $\overline{\text{CS256}}$, $\overline{\text{CPCS}}$, and $\overline{\text{IOS}}$ 1: Outputs CS256, CPCS, and IOS
1	PNCAH	0	R/W	Address Multiplex Address Hold Controls the output polarity of the address hold signal ($\overline{\text{A}}$ address multiplex extended mode. 0: Outputs $\overline{\text{AH}}$ 1: Outputs AH

7	MSTP15	0	R/W	Reserved The initial value should not be changed.
6	MSTP14	0	R/W	Data transfer controller (DTC)
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	8-bit PWM timer (PWM), 14-bit PWM timer (PWMX)
2	MSTP10	1	R/W	D/A converter
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 0 (SCI_0)
6	MSTP6	1	R/W	Serial communication interface 1 (SCI_1)
5	MSTP5	1	R/W	Serial communication interface 2 (SCI_2)
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)

MSTP11	MSTPA2	Function
0	0	14-bit PWM timer (PWMX_1) operates.
0	1	14-bit PWM timer (PWMX_1) stops.
1	0	14-bit PWM timer (PWMX_1) stops.
1	1	14-bit PWM timer (PWMX_1) stops.

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	0	14-bit PWM timer (PWMX_0) stops.
1	1	14-bit PWM timer (PWMX_0) stops.

MSTPCRH (bit 3) MSTP11	MSTPCRA (bit 0) MSTPA0	Function
0	0	8-bit PWM timer (PWM) operates.
0	1	8-bit PWM timer (PWM) stops.
1	0	8-bit PWM timer (PWM) stops.
1	1	8-bit PWM timer (PWM) stops.

Note: Bit 3 of MSTPCRH is the module stop bit of PWM, PWMX_0, and PWMX_1.

Bit	Bit Name	Initial Value	R/W	Corresponding Module
7 to 1	SMSTPB7 to SMSTPB1	All 1	R/W	Reserved The initial values should not be changed.
0	SMSTPB0	1	R/W	LPC interface (LPC)

23.1.5 Sub-Chip Module Stop Control Registers AH, AL (SUBMSTPAH, SUBMS

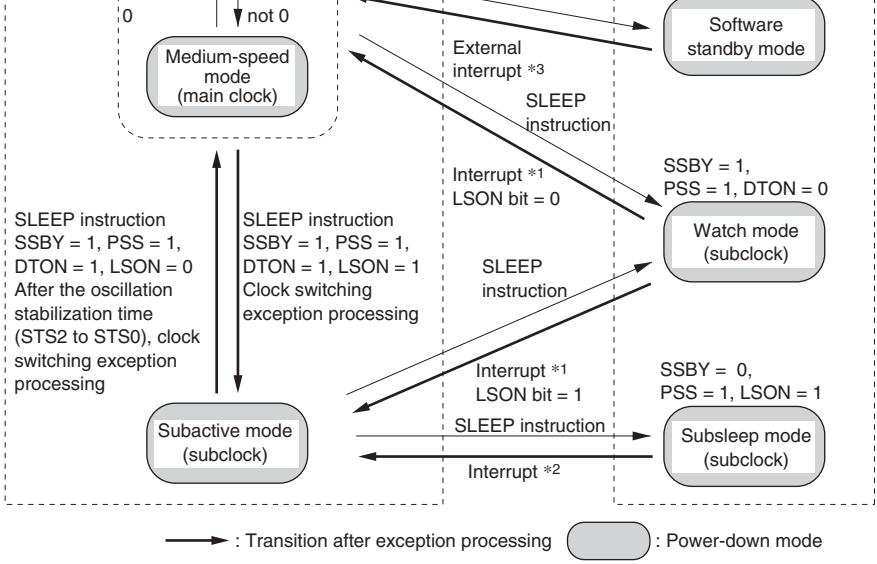
Set the values in SUBMSTPAH and SUBMSTPAL same as in SUBMSTPBH and SUBM

- SUBMSTPAH

Bit	Bit Name	Initial Value	R/W	Corresponding Module
7 to 0	SMSTPA15 to SMSTPA8	All 1	W	Reserved The initial values should not be changed.

- SUBMSTPAL

Bit	Bit Name	Initial Value	R/W	Corresponding Module
7 to 1	SMSTPA7 to SMSTPA1	All 1	W	Reserved The initial values should not be changed.
0	SMSTPA0	1	W	LPC interface (LPC)



Notes: • When a transition is made between modes by means of an interrupt, the transition cannot be ma
on interrupt source generation alone. Ensure that interrupt handling is performed after accepting
interrupt request.
• Always select high-speed mode before making a transition to watch mode or sub-active mode.

1. NMI, IRQ0 to IRQ15, KIN0 to KIN15, WUE8 to WUE15, and WDT_1 interrupts
2. NMI, IRQ0 to IRQ15, KIN0 to KIN15, WUE8 to WUE15, WDT_0, WDT_1, TMR_0, and TMR_1 interrupts
3. NMI, IRQ0 to IRQ15, KIN0 to KIN15, and WUE8 to WUE15 interrupts

Figure 23.1 Mode Transition Diagram

	KIN0 to KIN15								
	WUE8 to WUE15								
Peripheral modules	DTC	Function-ing	Function-ing in medium-speed mode/ Function-ing	Function-ing	Function-ing/Halted (retained)	Halted (retained)	Halted (retained)	Halted (retained)	Halted (retained)
	WDT_1		Function-ing		Function-ing	Subclock operation	Subclock operation	Subclock operation	
	WDT_0					Halted (retained)			
	TMR_0, TMR_1				Function-ing/Halted (retained)				
	LPC						Halted (retained)	Halted (retained)	
	FRT								
	TMR_X, TMR_Y								
	IIC_0 to IIC_5								

I/O	Function-ing	Function-ing

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).

the bus master operating clock. For example, if $\phi/4$ 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, and th
bit in LPWRCR is cleared to 0, a transition is made to sleep mode. When sleep mode is
an interrupt, medium-speed mode is restored. When the SLEEP instruction is executed v
SSBY bit set to 1, the LSON bit cleared to 0, and the PSS bit in TCSR (WDT_1) cleared
operation shifts to software standby mode. When software standby mode is cleared by a
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 23.2 shows an example of medium-speed mode timing.

Figure 23.2 Medium-Speed Mode Timing

23.4 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 and the LSON bit in LPWRCR is cleared to 0. In sleep mod operation stops but the peripheral modules do not stop. The contents of the CPU's interna 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 is not exited if the interrupt is disabled, or interrupts other than NMI are masked by the C

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.

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 with an IR
IRQ15 interrupt, set the corresponding enable bit to 1. When exiting software standby m
KIN0 to KIN15 or WUE8 to WUE15 interrupt, enable the input. In these cases, ensure t
interrupt with a higher priority than interrupts IRQ0 to IRQ15 is generated. In the case o
to IRQ15 interrupt, software standby mode is not exited if the corresponding enable bit i
to 0 or if the interrupt has been masked by the CPU. In the case of a KIN0 to KIN15 or
WUE15 interrupt, software standby mode is not exited if input is disabled or if the inter
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 23.3 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.

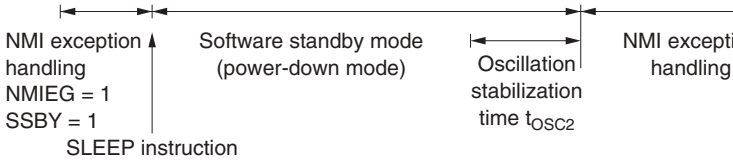


Figure 23.3 Software Standby Mode Application Example

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 23.4 shows an example of hardware standby mode timing.

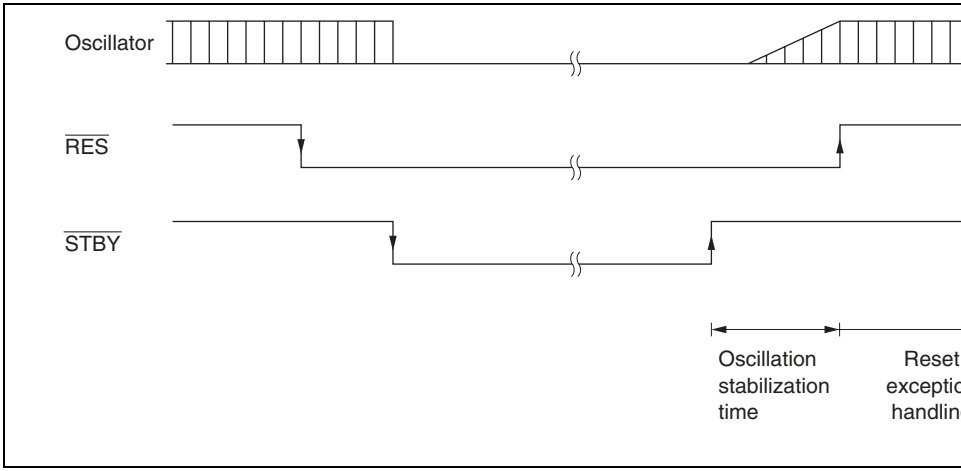


Figure 23.4 Hardware Standby Mode Timing

When an interrupt occurs, watch mode is exited and a transition is made to high-speed mo
medium-speed mode when the LSON bit in LPWRCR cleared to 0 or to subactive mode w
LSON bit is set to 1. When a transition is made to high-speed mode, a stable clock is supp
the entire LSI and interrupt exception handling starts after the time set in the STS2 to STS
SBYCR has elapsed.

In the case of an IRQ0 to IRQ15 interrupt, watch mode is not exited if the corresponding
bit has been cleared to 0 or the interrupt is masked by the CPU. In the case of a KIN0 to K
WUE8 to WUE15 interrupt, watch mode is not exited if input is disabled or the interrupt
masked by the CPU. In the case of an interrupt from the on-chip peripheral modules, wat
is not exited if the interrupt enable register has been set to disable the reception of that int
the interrupt is masked by the CPU.

When the $\overline{\text{RES}}$ pin is driven low, system clock oscillation starts. Simultaneously with the
system clock oscillation, the system clock is supplied to the entire LSI. Note that the $\overline{\text{RES}}$
must be held low until clock oscillation is settled. If the $\overline{\text{RES}}$ pin is driven high after the
oscillation settling time has passed, the CPU begins reset exception handling.

If the $\overline{\text{STBY}}$ pin is driven low, the LSI enters hardware standby mode.

When an interrupt occurs, subsleep mode is exited and interrupt exception handling star

In the case of an IRQ0 to IRQ15 interrupt, subsleep mode is not exited if the correspond bit has been cleared to 0 or the interrupt is masked by the CPU. In the case of a KIN0 to WUE8 to WUE15 interrupt, subsleep mode is not exited if input is disabled or the interr masked by the CPU. In the case of an interrupt from the on-chip peripheral modules, sul mode is not exited if the interrupt enable register has been set to disable the reception of interrupt or the interrupt is masked by the CPU.

When the $\overline{\text{RES}}$ pin is driven low, system clock oscillation starts. Simultaneously with th system clock oscillation, the system clock is supplied to the entire LSI. Note that the $\overline{\text{RE}}$ must be held low until clock oscillation is settled. If the $\overline{\text{RES}}$ pin is driven high after the oscillation settling time has passed, the CPU begins reset exception handling.

If the $\overline{\text{STBY}}$ pin is driven low, the LSI enters hardware standby mode.

0.

Subactive mode is exited by the SLEEP instruction, $\overline{\text{RES}}$ pin input, or $\overline{\text{STBY}}$ pin input.

When the SLEEP instruction is executed with the SSBY bit in SBYCR set to 1, the DTO
LPWRCR cleared to 0, and the PSS bit in TCSR (WDT_1) set to 1, the CPU exits subact
and a transition is made to watch mode. When the SLEEP instruction is executed with the
bit in SBYCR cleared to 0, the LSON bit in LPWRCR set to 1, and the PSS bit in TCSR
set to 1, a transition is made to subsleep mode. When the SLEEP instruction is executed
SSBY bit in SBYCR set to 1, the DTON bit and LSON bit in LPWRCR set to 10, and the
in TCSR (WDT_1) set to 1, a direct transition is made to high-speed mode.

For details of direct transitions, see section 23.11, Direct Transitions.

When the $\overline{\text{RES}}$ pin is driven low, system clock oscillation starts. Simultaneously with the
system clock oscillation, the system clock is supplied to the entire LSI. Note that the $\overline{\text{RES}}$
must be held low until the clock oscillation is settled. If the $\overline{\text{RES}}$ pin is driven high after
oscillation settling time has passed, the CPU begins reset exception handling.

If the $\overline{\text{STBY}}$ pin is driven low, the LSI enters hardware standby mode.

disabled.

23.11 Direct Transitions

The CPU executes programs in three modes: high-speed, medium-speed, and subactive. direct transition is made from high-speed mode to subactive mode, there is no interrupti program execution. A direct transition is enabled by setting the DTON bit in LPWRCR then executing the SLEEP instruction. After a transition, direct transition exception hand starts.

The CPU makes a transition to subactive mode when the SLEEP instruction is executed speed mode with the SSBY bit in SBYCR set to 1, the LSON bit and DTON bit in LPW to 11, and the PSS bit in TCSR (WDT_1) set to 1.

To make a direct transition to high-speed mode after the time set in the STS2 to STS0 bi SBYCR has elapsed, execute the SLEEP instruction in subactive mode with the SSBY b SBYCR set to 1, the LSON bit and DTON bit in LPWRCR set to 01, and the PSS bit in (WDT_1) set to 1.

23.12.3 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.

23.12.4 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 sublock
EXCLE bit should not be set to 1.

- 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.

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

24.1 Register Addresses (Address Order)

The data bus width indicates the numbers of bits by which the register is accessed.

The number of access states indicates the number of states based on the specified referer

Note: Access to undefined or reserved addresses is prohibited. Since operation or cont
operation is not guaranteed when these registers are accessed, do not attempt su

SMIC control status register 1	SMICCSR	8	H'FE0A	LPC	16	
SMIC data register	SMICDTR	8	H'FE0B	LPC	16	
SMIC interrupt register_0	SMICIR0	8	H'FE0C	LPC	16	
SMIC interrupt register_1	SMICIR1	8	H'FE0E	LPC	16	
Two-way data register 0MW	TWR0MW	8	H'FE10	LPC	16	
Two-way data register 0SW	TWR0SW	8	H'FE10	LPC	16	
Two-way data register 1	TWR1	8	H'FE11	LPC	16	
Two-way data register 2	TWR2	8	H'FE12	LPC	16	
Two-way data register 3	TWR3	8	H'FE13	LPC	16	
Two-way data register 4	TWR4	8	H'FE14	LPC	16	
Two-way data register 5	TWR5	8	H'FE15	LPC	16	
Two-way data register 6	TWR6	8	H'FE16	LPC	16	
Two-way data register 7	TWR7	8	H'FE17	LPC	16	
Two-way data register 8	TWR8	8	H'FE18	LPC	16	
Two-way data register 9	TWR9	8	H'FE19	LPC	16	
Two-way data register 10	TWR10	8	H'FE1A	LPC	16	
Two-way data register 11	TWR11	8	H'FE1B	LPC	16	
Two-way data register 12	TWR12	8	H'FE1C	LPC	16	
Two-way data register 13	TWR13	8	H'FE1D	LPC	16	
Two-way data register 14	TWR14	8	H'FE1E	LPC	16	
Two-way data register 15	TWR15	8	H'FE1F	LPC	16	

Output data register 1	ODR1	8	H'FE29	LPC	16
Status register 1	STR1	8	H'FE2A	LPC	16
Input data register 2	IDR2	8	H'FE2C	LPC	16
Output data register 2	ODR2	8	H'FE2D	LPC	16
Status register 2	STR2	8	H'FE2E	LPC	16
Host interface select register	HISEL	8	H'FE2F	LPC	16
Host interface control register 0	HICR0	8	H'FE30	LPC	16
Host interface control register 1	HICR1	8	H'FE31	LPC	16
Host interface control register 2	HICR2	8	H'FE32	LPC	16
Host interface control register 3	HICR3	8	H'FE33	LPC	16
SERIRQ control register 2	SIRQCR2	8	H'FE34	LPC	16
BT data buffer	BTDTR	8	H'FE35	LPC	16
BT FIFO enable size register 0	BTFVSR0	8	H'FE36	LPC	16
BT FIFO enable size register 1	BTFVSR1	8	H'FE37	LPC	16
LPC channel 1, 2 address register H	LADR12H	8	H'FE38	LPC	16
LPC channel 1, 2 address register L	LADR12L	8	H'FE39	LPC	16

Event count control register	ECCR	8	H'FE42	EVC	8	
Module stop control register A	MSTPCRA	8	H'FE43	SYSTEM	8	
Noise canceler enable register	P6NCE	8	H'FE44	PORT	8	
Noise canceler decision control register	P6NCMC	8	H'FE45	PORT	8	
Noise canceler cycle setting register	P6NCCS	8	H'FE46	PORT	8	
Port E output data register	PEODR	8	H'FE48	PORT	8	
Port F output data register	PFODR	8	H'FE49	PORT	8	
Port E input data register	PEPIN	8	H'FE4A (Read)	PORT	8	
Port E data direction register	PEDDR	8	H'FE4A (Write)	PORT	8	
Port F input data register	PFPIN	8	H'FE4B (Read)	PORT	8	
Port F data direction register	PFDDR	8	H'FE4B (Write)	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 (Read)	PORT	8	
Port C data direction register	PCDDR	8	H'FE4E (Write)	PORT	8	
Port D input data register	PDPIN	8	H'FE4F (Read)	PORT	8	

register					
I²C bus control register_4	ICCR_4	8	H'FEB0	IIC_4	8
I²C bus status register_4	ICSR_4	8	H'FEB1	IIC_4	8
I²C bus data register_4	ICDR_4	8	H'FEB2	IIC_4	8
Second slave address register_4	SARX_4	8	H'FEB2	IIC_4	8
I²C bus mode register_4	ICMR_4	8	H'FEB3	IIC_4	8
Slave address register_4	SAR_4	8	H'FEB3	IIC_4	8
I²C bus control register_5	ICCR_5	8	H'FEB4	IIC_5	8
I²C bus status register_5	ICSR_5	8	H'FEB5	IIC_5	8
I²C bus data register_5	ICDR_5	8	H'FEB6	IIC_5	8
Second slave address register_5	SARX_5	8	H'FEB6	IIC_5	8
I²C bus mode register_5	ICMR_5	8	H'FEB7	IIC_5	8
Slave address register_5	SAR_5	8	H'FEB7	IIC_5	8
I²C bus control register_3	ICCR_3	8	H'FEC0	IIC_3	8
I²C bus status register_3	ICSR_3	8	H'FEC1	IIC_3	8
I²C bus data register_3	ICDR_3	8	H'FEC2	IIC_3	8
Second slave address register_3	SARX_3	8	H'FEC2	IIC_3	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

Serial extended mode register_2	SEMR_2	8	H'FED2	SCI_2	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	
I^2C bus control extended register_0	ICXR_0	8	H'FED8	IIC_0	8	
I^2C bus control extended register_1	ICXR_1	8	H'FED9	IIC_1	8	
I^2C SMBus control register	ICSMBCR	8	H'FEDB	IIC	8	
I^2C bus control extended register_2	ICXR_2	8	H'FEDC	IIC_2	8	
I^2C bus control extended register_3	ICXR_3	8	H'FEDD	IIC_3	8	
I^2C bus transfer select register	IICX3	8	H'FEDF	IIC	8	
I^2C bus control extended register_4	ICXR_4	8	H'FEE0	IIC_4	8	
I^2C bus control extended register_5	ICXR_5	8	H'FEE1	IIC_5	8	
Keyboard comparator control register	KBCOMP	8	H'FEE4	EVC	8	
Serial interface control register	SCICR	8	H'FEE5	SCI_1	8	
Interrupt control register D	ICRD	8	H'FEE7	INT	8	
Interrupt control register A	ICRA	8	H'FEE8	INT	8	
Interrupt control register B	ICRB	8	H'FEE9	INT	8	
Interrupt control register C	ICRC	8	H'FEEA	INT	8	
IRQ status register	ISR	8	H'FEEB	INT	8	
IRQ sense control register H	ISCRH	8	H'FEEC	INT	8	
IRQ sense control register L	ISCRL	8	H'FEED	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'FEEA	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	PWM	8
System control register 2	SYSCR2	8	H'FF83	SYSTEM	8
Standby control register	SBYCR	8	H'FF84	SYSTEM	8
Low power control register	LPWRCR	8	H'FF85	SYSTEM	8
Module stop control register H	MSTPCRH	8	H'FF86	SYSTEM	8
Module stop control register L	MSTPCRL	8	H'FF87	SYSTEM	8
Serial mode register_1	SMR_1	8	H'FF88	SCI_1	8
I^2C bus control register_1	ICCR_1	8	H'FF88	IIC_1	8
Bit rate register_1	BRR_1	8	H'FF89	SCI_1	8
I^2C bus status register_1	ICSR_1	8	H'FF89	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	
Input capture register A	ICRA	16	H'FF98	FRT	16	
Output Compare register AR	OCRAR	16	H'FF98	FRT	16	
Input capture register B	ICRB	16	H'FF9A	FRT	16	
Output Compare register AF	OCRAF	16	H'FF9A	FRT	16	
Input capture register C	ICRC	16	H'FF9C	FRT	16	
Output compare register DM	OCRDM	16	H'FF9C	FRT	16	
Input capture register D	ICRD	16	H'FF9E	FRT	16	
Serial mode register_2	SMR_2	8	H'FFA0	SCI_2	8	
PWMX (D/A) control register_0	DACR_0	8	H'FFA0	PWMX_0	8	
PWMX (D/A) data register A_0	DADRA_0	16	H'FFA0	PWMX_0	8	
Bit rate register_2	BRR_2	8	H'FFA1	SCI_2	8	
Serial control register_2	SCR_2	8	H'FFA2	SCI_2	8	
Transmit data register_2	TDR_2	8	H'FFA3	SCI_2	8	

			(write)		
Timer counter_0	TCNT_0	8	H'FFA9 (read)	WDT_0	16
Timer counter_0	TCNT_0	16	H'FFA8 (write)	WDT_0	16
Port A output data register	PAODR	8	H'FFAA	PORT	8
Port A input data register	PAPIN	8	H'FFAB (read)	PORT	8
Port A data direction register	PADDR	8	H'FFAB (write)	PORT	8
Port 1 pull-up MOS control register	P1PCR	8	H'FFAC	PORT	8
Port 2 pull-up MOS control register	P2PCR	8	H'FFAD	PORT	8
Port 3 pull-up MOS control register	P3PCR	8	H'FFAE	PORT	8
Port 1 data direction register	P1DDR	8	H'FFB0	PORT	8
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

			(Write)			
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	
Bus control register	BCR	8	H'FFC6	BSC	8	
Wait state control register	WSCR	8	H'FFC7	BSC	8	
Timer control register_0	TCR_0	8	H'FFC8	TMR_0	16	
Timer control register_1	TCR_1	8	H'FFC9	TMR_1	16	
Timer control/status register_0	TCSR_0	8	H'FFCA	TMR_0	16	
Timer control/status register_1	TCSR_1	8	H'FFCB	TMR_1	16	
Time constant register A_0	TCORA_0	8	H'FFCC	TMR_0	16	
Time constant register A_1	TCORA_1	8	H'FFCD	TMR_1	16	
Time constant register B_0	TCORB_0	8	H'FFCE	TMR_0	16	
Time constant register B_1	TCORB_1	8	H'FFCF	TMR_1	16	
Timer counter_0	TCNT_0	8	H'FFD0	TMR_0	16	
Timer counter_1	TCNT_1	8	H'FFD1	TMR_1	16	
PWM output enable register B	PWOERB	8	H'FFD2	PWM	8	
PWM output enable register A	PWOERA	8	H'FFD3	PWM	8	

Serial control register_0	SCR_0	8	H'FFDA	SCI_0	8
Transmit data register_0	TDR_0	8	H'FFDB	SCI_0	8
Serial status register_0	SSR_0	8	H'FFDC	SCI_0	8
Receive data register_0	RDR_0	8	H'FFDD	SCI_0	8
Smart card mode register_0	SCMR_0	8	H'FFDE	SCI_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
A/D data register AH	ADDRAH	8	H'FFE0	A/D converter	8
A/D data register AL	ADDRAL	8	H'FFE1	A/D converter	8
A/D data register BH	ADDRBH	8	H'FFE2	A/D converter	8
A/D data register BL	ADDRBL	8	H'FFE3	A/D converter	8
A/D data register CH	ADDRCH	8	H'FFE4	A/D converter	8
A/D data register CL	ADDRCL	8	H'FFE5	A/D converter	8

			(read)			
Timer control/status register_1	TCSR_1	16	H'FFEA (write)	WDT_1	16	
Timer counter_1	TCNT_1	8	H’FFEB (read)	WDT_1	16	
Timer counter_1	TCNT_1	16	H'FFEA (write)	WDT_1	16	
Timer control register_X	TCR_X	8	H'FFF0	TMR_X	8	
Timer control register_Y	TCR_Y	8	H'FFF0	TMR_Y	8	
Keyboard matrix interrupt mask register 6	KMIMR6	8	H'FFF1	INT	8	
Timer control/status register_X	TCSR_X	8	H'FFF1	TMR_X	8	
Timer control/status register_Y	TCSR_Y	8	H'FFF1	TMR_Y	8	
Port 6 pull-up MOS control register	KMPCR6	8	H'FFF2	PORT	8	
Input capture register R	TICRR	8	H'FFF2	TMR_X	8	
Time constant register A_Y	TCORA_Y	8	H'FFF2	TMR_Y	8	
Keyboard matrix interrupt mask register A	KMIMRA	8	H'FFF3	INT	8	
Input capture register F	TICRF	8	H'FFF3	TMR_X	8	
Time constant register B_Y	TCORB_Y	8	H'FFF3	TMR_Y	8	
Wakeup event interrupt mask register 3	WUEMR3	8	H'FFF4	INT	8	

D/A data register 1	DADR1	8	H'FFF9	D/A converter	8
D/A control register	DACR	8	H'FFFA	D/A converter	8
Timer connection register I	TCONRI	8	H'FFFC	TMR	8
Timer connection register S	TCONRS	8	H'FFFE	TMR	8

BTCSR1	RSTRENBL	HRSTIE	IRQCRIE	BEVTIE	B2HIE	H2BIE	CRRPIE	CRWPIE
BTCR	B_BUSY	H_BUSY	OEM0	BEVT_ATN	B2H_ATN	H2B_ATN	CLR_RD_PTR	CLR_WR_PTR
BTIMSR	BMC_HWRST	—	—	OME3	OME2	OME1	B2H_IRQ	B2H_IRQ_EN
SMICFLG	RX_DATA_RDY	TX_DATA_RDY	—	SMI	SEVT_ATN	SMS_ATN	—	BUSY
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
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
TWR8	Bit 7	Bit 6	Bit 5	Bit 4	Bit 3	Bit 2	Bit 1	Bit 0
TWR9	Bit 7	Bit 6	Bit 5	Bit 4	Bit 3	Bit 2	Bit 1	Bit 0
TWR10	Bit 7	Bit 6	Bit 5	Bit 4	Bit 3	Bit 2	Bit 1	Bit 0
TWR11	Bit 7	Bit 6	Bit 5	Bit 4	Bit 3	Bit 2	Bit 1	Bit 0
TWR12	Bit 7	Bit 6	Bit 5	Bit 4	Bit 3	Bit 2	Bit 1	Bit 0
TWR13	Bit 7	Bit 6	Bit 5	Bit 4	Bit 3	Bit 2	Bit 1	Bit 0

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}$1	DBU12	IBF1	OBF1
IDR2	Bit 7	Bit 6	Bit 5	Bit 4	Bit 3	Bit 2	Bit 1	Bit 0
ODR2	Bit 7	Bit 6	Bit 5	Bit 4	Bit 3	Bit 2	Bit 1	Bit 0
STR2	DBU27	DBU26	DBU25	DBU24	C/$\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
SIRQCR2	IEDIR3	—	—	—	—	—	—	—
BTDTR	Bit 7	Bit 6	Bit 5	Bit 4	Bit 3	Bit 2	Bit 1	Bit 0
BTFVSR0	N7	N6	N5	N4	N3	N2	N1	N0
BTFVSR1	N7	N6	N5	N4	N3	N2	N1	N0
LADR12H	Bit 15	Bit 14	Bit 13	Bit 12	Bit 11	Bit 10	Bit 9	Bit 8
LADR12L	Bit 7	Bit 6	Bit 5	Bit 4	Bit 3	—	Bit 1	Bit 0
SUBMSTPAH	SMSTPA15	SMSTPA14	SMSTPA13	SMSTPA12	SMSTPA11	SMSTPA10	SMSTPA9	SMSTPA8
SUBMSTPAL	SMSTPA7	SMSTPA6	SMSTPA5	SMSTPA4	SMSTPA3	SMSTPA2	SMSTPA1	SMSTPA0
SUBMSTPBH	SMSTPB15	SMSTPB14	SMSTPB13	SMSTPB12	SMSTPB11	SMSTPB10	SMSTPB9	SMSTPB8
SUBMSTPBL	SMSTPB7	SMSTPB6	SMSTPB5	SMSTPB4	SMSTPB3	SMSTPB2	SMSTPB1	SMSTPB0

PEPIN	PE7PIN	PE6PIN	PE5PIN	PE4PIN	PE3PIN	PE2PIN	PE1PIN	PE0PIN
PEDDR	PE7DDR	PE6DDR	PE5DDR	PE4DDR	PE3DDR	PE2DDR	PE1DDR	PE0DDR
PFPIN	—	—	—	—	—	PF2PIN	PF1PIN	PF0PIN
PFDDR	—	—	—	—	—	PF2DDR	PF1DDR	PF0DDR
PCODR	PC7ODR	PC6ODR	PC5ODR	PC4ODR	PC3ODR	PC2ODR	PC1ODR	PC0ODR
PDODR	PD7ODR	PD6ODR	PD5ODR	PD4ODR	PD3ODR	PD2ODR	PD1ODR	PD0ODR
PCPIN	PC7PIN	PC6PIN	PC5PIN	PC4PIN	PC3PIN	PC2PIN	PC1PIN	PC0PIN
PCDDR	PC7DDR	PC6DDR	PC5DDR	PC4DDR	PC3DDR	PC2DDR	PC1DDR	PC0DDR
PDPIN	PD7PIN	PD6PIN	PD5PIN	PD4PIN	PD3PIN	PD2PIN	PD1PIN	PD0PIN
PDDDR	PD7DDR	PD6DDR	PD5DDR	PD4DDR	PD3DDR	PD2DDR	PD1DDR	PD0DDR
FCCS	FWE	—	—	FLER	WEINTE	—	—	SC0
FPCS	—	—	—	—	—	—	—	PPVS
FECS	—	—	—	—	—	—	—	EPVB
FKEY	K7	K6	K5	K4	K3	K2	K1	K0
FMATS	MS7	MS6	MS5	MS4	MS3	MS2	MS1	MS0
FTDAR	TDER	TDA6	TDA5	TDA4	TDA3	TDA2	TDA1	TDA0
ICCR_4	ICE	IEIC	MST	TRS	ACKE	BBSY	IRIC	SCP
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

SARX_3	SVAX6	SVAX5	SVAX4	SVAX3	SVAX2	SVAX1	SVAX0	FSX
ICMR_3	MLS	WAIT	CKS2	CKS1	CKS0	BC2	BC1	BC0
SAR_3	SVA6	SVA5	SVA4	SVA3	SVA2	SVA1	SVA0	FS
ICCR_2	ICE	IEIC	MST	TRS	ACKE	BBSY	IRIC	SCP
ICSR_2	ESTP	STOP	IRTR	AASX	AL	AAS	ADZ	ACKB
ICDR_2	Bit 7	Bit 6	Bit 5	Bit 4	Bit 3	Bit 2	Bit 1	Bit 0
SARX_2	SVAX6	SVAX5	SVAX4	SVAX3	SVAX2	SVAX1	SVAX0	FSX
ICMR_2	MLS	WAIT	CKS2	CKS1	CKS0	BC2	BC1	BC0
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
SEMR_0	SSE	—	—	ACS4	ABCS	ACS2	ACS1	ACS0
SEMR_2	SSE	—	—	ACS4	ABCS	ACS2	ACS1	ACS0
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

ICRA	ICRA7	ICRA6	ICRA5	ICRA4	ICRA3	ICRA2	ICRA1	ICRA0
ICRB	ICRB7	ICRB6	—	ICRB4	ICRB3	ICRB2	ICRB1	ICRB0
ICRC	ICRC7	ICRC6	ICRC5	ICRC4	ICRC3	ICRC2	ICRC1	—
ISR	IRQ7F	IRQ6F	IRQ5F	IRQ4F	IRQ3F	IRQ2F	IRQ1F	IRQ0F
ISCRH	IRQ7SCB	IRQ7SCA	IRQ6SCB	IRQ6SCA	IRQ5SCB	IRQ5SCA	IRQ4SCB	IRQ4SCA
ISCRL	IRQ3SCB	IRQ3SCA	IRQ2SCB	IRQ2SCA	IRQ1SCB	IRQ1SCA	IRQ0SCB	IRQ0SCA
DTCERA	DTCEA7	DTCEA6	DTCEA5	DTCEA4	DTCEA3	DTCEA2	DTCEA1	DTCEA0
DTCERB	DTCEB7	DTCEB6	DTCEB5	—	—	DTCEB2	DTCEB1	DTCEB0
DTCERC	DTCEC7	DTCEC6	DTCEC5	DTCEC4	DTCEC3	DTCEC2	DTCEC1	DTCEC0
DTCERD	DTCED7	DTCED6	DTCED5	DTCED4	DTCED3	—	—	DTCED0
DTCERE	—	—	—	—	DTCEE3	DTCEE2	DTCEE1	DTCEE0
DTVECR	SWDTE	DTVEC6	DTVEC5	DTVEC4	DTVEC3	DTVEC2	DTVEC1	DTVEC0
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	IRQ12SCA
ISCR16L	IRQ11SCB	IRQ11SCA	IRQ10SCB	IRQ10SCA	IRQ9SCB	IRQ9SCA	IRQ8SCB	IRQ8SCA
ISSR16	ISS15	ISS14	ISS13	ISS12	ISS11	ISS0	ISS9	ISS8
ISSR	ISS7	ISS6	ISS5	ISS4	ISS3	ISS2	ISS1	ISS0
PTCNT0	TMI0S	TMI1S	TMIXS	TMIYS	—	PWMS	—	—

	(GM)	(BLK)	(PE)	(O/E)	(BCP1)	(BCP0)	(CKS1)	(CKS0)
ICCR_1	ICE	IEIC	MST	TRS	ACKE	BBSY	IRIC	SCP
BRR_1	Bit 7	Bit 6	Bit 5	Bit 4	Bit 3	Bit 2	Bit 1	Bit 0
ICSR_1	ESTP	STOP	IRTR	AASX	AL	AAS	ADZ	ACKB
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
ICDR_1	Bit 7	Bit 6	Bit 5	Bit 4	Bit 3	Bit 2	Bit 1	Bit 0
SARX_1	SVAX6	SVAX5	SVAX4	SVAX3	SVAX2	SVAX1	SVAX0	FSX
ICMR_1	MLS	WAIT	CKS2	CKS1	CKS0	BC2	BC1	BC0
SAR_1	SVA6	SVA5	SVA4	SVA3	SVA2	SVA1	SVA0	FS
TIER	ICIAE	ICIBE	ICICE	ICIDE	OCIAE	OCIBE	OVIE	—
TCSR	ICFA	ICFB	ICFC	ICFD	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	IEDGA	IEDGB	IEDGC	IEDGD	BUFEA	BUFEB	CKS1	CKS0
TOCR	ICRDMS	OCRAMS	ICRS	OCRS	OEA	OEB	OLVLA	OLVLB

	Bit 7	Bit 6	Bit 5	Bit 4	Bit 3	Bit 2	Bit 1	Bit 0
OCRDM	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
ICRD	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
SMR_2*[3]	C/$\overline{A}$ (GM)	CHR (BLK)	PE (PE)	O/$\overline{E}$ (O/$\overline{E}$)	STOP (BCP1)	MP (BCP0)	CKS1 (CKS1)	CKS0 (CKS0)
DACR_0	—	PWME	—	—	OEB	OEA	OS	CKS
DADRA_0	DA13	DA12	DA11	DA10	DA9	DA8	DA7	DA6
	DA5	DA4	DA3	DA2	DA1	DA0	CFS	—
BRR_2	Bit 7	Bit 6	Bit 5	Bit 4	Bit 3	Bit 2	Bit 1	Bit 0
SCR_2	TIE	RIE	TE	RE	MPIE	TEIE	CKE1	CKE0
TDR_2	Bit 7	Bit 6	Bit 5	Bit 4	Bit 3	Bit 2	Bit 1	Bit 0
SSR_2*[3]	TDRE (TDRE)	RDRF (RDRF)	ORER (ORER)	FER (ERS)	PER (PER)	TEND (TEND)	MPB (MPB)	MPBT (MPBT)
RDR_2	Bit 7	Bit 6	Bit 5	Bit 4	Bit 3	Bit 2	Bit 1	Bit 0
SCMR_2	—	—	—	—	SDIR	SINV	—	SMIF
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

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	P82ODR	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	P82DDR	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	IICE	FLSHE	—	ICKS1	ICKS0
SYSCR	CS256E	IOSE	INTM1	INTM0	XRST	NMIEG	KINWUE	RAME
MDCR	EXPE	—	—	—	—	MDS2	MDS1	MDS0
BCR	—	ICIS	BRSTRM	BRSTS1	BRSTS0	—	IOS1	IOS0
WSCR	ABW256	AST256	ABW	AST	WMS1	WMS0	WC1	WC0

TCNT_1	Bit 7	Bit 6	Bit 5	Bit 4	Bit 3	Bit 2	Bit 1	Bit 0
PWOERB	OE15	OE14	OE13	OE12	OE11	OE10	OE9	OE8
PWOERA	OE7	OE6	OE5	OE4	OE3	OE2	OE1	OE0
PWDPRB	OS15	OS14	OS13	OS12	OS11	OS10	OS9	OS8
PWDPRA	OS7	OS6	OS5	OS4	OS3	OS2	OS1	OS0
PWSL	PWCKE	PWCKS	—	—	RS3	RS2	RS1	RS0
PWDR15-0	Bit 7	Bit 6	Bit 5	Bit 4	Bit 3	Bit 2	Bit 1	Bit 0
SMR_0*[3]	C/$\overline{A}$ (GM)	CHR (BLK)	PE (PE)	O/$\overline{E}$ (O/$\overline{E}$)	STOP (BCP1)	MP (BCP0)	CKS1 (CKS1)	CKS0 (CKS0)
ICCR_0	ICE	IEIC	MST	TRS	ACKE	BBSY	IRIC	SCP
BRR_0	Bit 7	Bit 6	Bit 5	Bit 4	Bit 3	Bit 2	Bit 1	Bit 0
ICSR_0	ESTP	STOP	IRTR	AASX	AL	AAS	ADZ	ACKB
SCR_0	TIE	RIE	TE	RE	MPIE	TEIE	CKE1	CKE0
TDR_0	Bit 7	Bit 6	Bit 5	Bit 4	Bit 3	Bit 2	Bit 1	Bit 0
SSR_0*[3]	TDRE (TDRE)	RDRF (RDRF)	ORER (ORER)	FER (ERS)	PER (PER)	TEND (TEND)	MPB (MPB)	MPBT (MPBT)
RDR_0	Bit 7	Bit 6	Bit 5	Bit 4	Bit 3	Bit 2	Bit 1	Bit 0
SCMR_0	—	—	—	—	SDIR	SINV	—	SMIF
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

ADCR	TRGS1	TRGS0	—	—	—	—	—	—
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	CCLR1	CCLR0	CKS2	CKS1	CKS0
TCR_Y	CMIEB	CMIEA	OVIE	CCLR1	CCLR0	CKS2	CKS1	CKS0
KMIMR6	KMIM7	KMIM6	KMIM5	KMIM4	KMIM3	KMIM2	KMIM1	KMIM0
TCSR_X	CMFB	CMFA	OVF	ICF	OS3	OS2	OS1	OS0
TCSR_Y	CMFB	CMFA	OVF	ICIE	OS3	OS2	OS1	OS0
KMPCR6	KM7PCR	KM6PCR	KM5PCR	KM4PCR	KM3PCR	KM2PCR	KM1PCR	KM0PCR
TICRR	Bit 7	Bit 6	Bit 5	Bit 4	Bit 3	Bit 2	Bit 1	Bit 0
TCORA_Y	Bit 7	Bit 6	Bit 5	Bit 4	Bit 3	Bit 2	Bit 1	Bit 0
KMIMRA	KMIM15	KMIM14	KMIM13	KMIM12	KMIM11	KMIM10	KMIM9	KMIM8
TICRF	Bit 7	Bit 6	Bit 5	Bit 4	Bit 3	Bit 2	Bit 1	Bit 0
TCORB_Y	Bit 7	Bit 6	Bit 5	Bit 4	Bit 3	Bit 2	Bit 1	Bit 0
WUEMR3	WUEM15	WUEM14	WUEM13	WUEM12	WUEM11	WUEM10	WUEM9	WUEM8
TCNT_X	Bit 7	Bit 6	Bit 5	Bit 4	Bit 3	Bit 2	Bit 1	Bit 0
TCNT_Y	Bit 7	Bit 6	Bit 5	Bit 4	Bit 3	Bit 2	Bit 1	Bit 0
TCORC	Bit 7	Bit 6	Bit 5	Bit 4	Bit 3	Bit 2	Bit 1	Bit 0
TISR	—	—	—	—	—	—	—	IS
TCORA_X	Bit 7	Bit 6	Bit 5	Bit 4	Bit 3	Bit 2	Bit 1	Bit 0
TCORB_X	Bit 7	Bit 6	Bit 5	Bit 4	Bit 3	Bit 2	Bit 1	Bit 0
DADR0	Bit 7	Bit 6	Bit 5	Bit 4	Bit 3	Bit 2	Bit 1	Bit 0
DADR1	Bit 7	Bit 6	Bit 5	Bit 4	Bit 3	Bit 2	Bit 1	Bit 0
DACR	DAOE1	DAOE0	DAE	—	—	—	—	—

BTIMSR	Initialized	—	—	—	—	—	—	—	Initiali
SMICFLG	Initialized	—	—	—	—	—	—	—	Initiali
SMICCSR	—	—	—	—	—	—	—	—	—
SMICDTR	—	—	—	—	—	—	—	—	—
SMICIR0	Initialized	—	—	—	—	—	—	—	Initiali
SMICIR1	Initialized	—	—	—	—	—	—	—	Initiali
TWR0MW	—	—	—	—	—	—	—	—	—
TWR0SW	—	—	—	—	—	—	—	—	—
TWR1	—	—	—	—	—	—	—	—	—
TWR2	—	—	—	—	—	—	—	—	—
TWR3	—	—	—	—	—	—	—	—	—
TWR4	—	—	—	—	—	—	—	—	—
TWR5	—	—	—	—	—	—	—	—	—
TWR6	—	—	—	—	—	—	—	—	—
TWR7	—	—	—	—	—	—	—	—	—
TWR8	—	—	—	—	—	—	—	—	—
TWR9	—	—	—	—	—	—	—	—	—
TWR10	—	—	—	—	—	—	—	—	—
TWR11	—	—	—	—	—	—	—	—	—
TWR12	—	—	—	—	—	—	—	—	—
TWR13	—	—	—	—	—	—	—	—	—
TWR14	—	—	—	—	—	—	—	—	—
TWR15	—	—	—	—	—	—	—	—	—
IDR3	—	—	—	—	—	—	—	—	—
ODR3	—	—	—	—	—	—	—	—	—

IDR2	—	—	—	—	—	—	—	—	—
ODR2	—	—	—	—	—	—	—	—	—
STR2	Initialized	—	—	—	—	—	—	—	Initialize
HISEL	Initialized	—	—	—	—	—	—	—	Initialize
HICR0	Initialized	—	—	—	—	—	—	—	Initialize
HICR1	Initialized	—	—	—	—	—	—	—	Initialize
HICR2	Initialized	—	—	—	—	—	—	—	Initialize
HICR3	—	—	—	—	—	—	—	—	—
SIRQCR2	Initialized	—	—	—	—	—	—	—	Initialize
BTDTR	—	—	—	—	—	—	—	—	—
BTFVSR0	Initialized	—	—	—	—	—	—	—	Initialize
BTFVSR1	Initialized	—	—	—	—	—	—	—	Initialize
LADR12H	Initialized	—	—	—	—	—	—	—	Initialize
LADR12L	Initialized	—	—	—	—	—	—	—	Initialize
SUBMSTPAH	Initialized	—	—	—	—	—	—	—	Initialize
SUBMSTPAL	Initialized	—	—	—	—	—	—	—	Initialize
SUBMSTPBH	Initialized	—	—	—	—	—	—	—	Initialize
SUBMSTPBL	Initialized	—	—	—	—	—	—	—	Initialize
ECS	Initialized	—	—	—	—	—	—	—	Initialize
ECCR	Initialized	—	—	—	—	—	—	—	Initialize
MSTPCRA	Initialized	—	—	—	—	—	—	—	Initialize

PFDDR	Initialized	—	—	—	—	—	—	—	Initiali
PCODR	Initialized	—	—	—	—	—	—	—	Initiali
PDODR	Initialized	—	—	—	—	—	—	—	Initiali
PCPIN	—	—	—	—	—	—	—	—	—
PCDDR	Initialized	—	—	—	—	—	—	—	Initiali
PDPIN	—	—	—	—	—	—	—	—	—
PDDDR	Initialized	—	—	—	—	—	—	—	Initiali
FCCS	Initialized	—	—	—	—	—	—	—	Initiali
FPCS	Initialized	—	—	—	—	—	—	—	Initiali
FECS	Initialized	—	—	—	—	—	—	—	Initiali
FKEY	Initialized	—	—	—	—	—	—	—	Initiali
FMATS	Initialized	—	—	—	—	—	—	—	Initiali
FTDAR	Initialized	—	—	—	—	—	—	—	Initiali
ICCR_4	Initialized	—	—	—	—	—	—	—	Initiali
ICSR_4	Initialized	—	—	—	—	—	—	—	Initiali
ICDR_4	—	—	—	—	—	—	—	—	—
SARX_4	Initialized	—	—	—	—	—	—	—	Initiali
ICMR_4	Initialized	—	—	—	—	—	—	—	Initiali
SAR_4	Initialized	—	—	—	—	—	—	—	Initiali
ICCR_5	Initialized	—	—	—	—	—	—	—	Initiali
ICSR_5	Initialized	—	—	—	—	—	—	—	Initiali
ICDR_5	—	—	—	—	—	—	—	—	—
SARX_5	Initialized	—	—	—	—	—	—	—	Initiali
ICMR_5	Initialized	—	—	—	—	—	—	—	Initiali
SAR_5	Initialized	—	—	—	—	—	—	—	Initiali

ICDR_2	—	—	—	—	—	—	—	—	—
SARX_2	Initialized	—	—	—	—	—	—	—	Initialize
ICMR_2	Initialized	—	—	—	—	—	—	—	Initialize
SAR_2	Initialized	—	—	—	—	—	—	—	Initialize
DADRA_1	Initialized	—	Initialized	—	Initialized	Initialized	Initialized	Initialized	Initialize
DACR_1	Initialized	—	Initialized	—	Initialized	Initialized	Initialized	Initialized	Initialize
DADRB_1	Initialized	—	Initialized	—	Initialized	Initialized	Initialized	Initialized	Initialize
DACNT_1	Initialized	—	Initialized	—	Initialized	Initialized	Initialized	Initialized	Initialize
SEMR_0	Initialized	—	—	—	—	—	—	—	Initialize
SEMR_2	Initialized	—	—	—	—	—	—	—	Initialize
CRCCR	Initialized	—	—	—	—	—	—	—	Initialize
CRCDIR	Initialized	—	—	—	—	—	—	—	Initialize
CRCDOR	Initialized	—	—	—	—	—	—	—	Initialize
ICXR_0	Initialized	—	—	—	—	—	—	—	Initialize
ICXR_1	Initialized	—	—	—	—	—	—	—	Initialize
ICSMBCR	initialized	—	—	—	—	—	—	—	Initialize
ICXR_2	Initialized	—	—	—	—	—	—	—	Initialize
ICXR_3	Initialized	—	—	—	—	—	—	—	Initialize
IICX3	Initialized	—	—	—	—	—	—	—	Initialize
ICXR_4	Initialized	—	—	—	—	—	—	—	Initialize
ICXR_5	Initialized	—	—	—	—	—	—	—	Initialize
KBCOMP	Initialized	—	—	—	—	—	—	—	Initialize
SCICR	Initialized	—	—	—	—	—	—	—	Initialize
ICRD	Initialized	—	—	—	—	—	—	—	Initialize

DTCERC	Initialized	—	—	—	—	—	—	—	Initiali
DTCERD	Initialized	—	—	—	—	—	—	—	Initiali
DTCERE	Initialized	—	—	—	—	—	—	—	Initiali
DTVECR	Initialized	—	—	—	—	—	—	—	Initiali
ABRKCR	Initialized	—	—	—	—	—	—	—	Initiali
BARA	Initialized	—	—	—	—	—	—	—	Initiali
BARB	Initialized	—	—	—	—	—	—	—	Initiali
BARC	Initialized	—	—	—	—	—	—	—	Initiali
IER16	Initialized	—	—	—	—	—	—	—	Initiali
ISR16	Initialized	—	—	—	—	—	—	—	Initiali
ISCR16H	Initialized	—	—	—	—	—	—	—	Initiali
ISCR16L	Initialized	—	—	—	—	—	—	—	Initiali
ISSR16	Initialized	—	—	—	—	—	—	—	Initiali
ISSR	Initialized	—	—	—	—	—	—	—	Initiali
PTCNT0	Initialized	—	—	—	—	—	—	—	Initiali
BCR2	Initialized	—	—	—	—	—	—	—	Initiali
WSCR2	Initialized	—	—	—	—	—	—	—	Initiali
PCSR	Initialized	—	Initialized	—	Initialized	Initialized	Initialized	Initialized	Initiali
SYSCR2	Initialized	—	—	—	—	—	—	—	Initiali
SBYCR	Initialized	—	—	—	—	—	—	—	Initiali
LPWRCR	Initialized	—	—	—	—	—	—	—	Initiali
MSTPCRH	Initialized	—	—	—	—	—	—	—	Initiali
MSTPCRL	Initialized	—	—	—	—	—	—	—	Initiali
SMR_1	Initialized	—	—	—	—	—	—	—	Initiali
ICCR_1	Initialized	—	—	—	—	—	—	—	Initiali

SARX_1	Initialized	—	—	—	—	—	—	—	Initialize
ICMR_1	Initialized	—	—	—	—	—	—	—	Initialize
SAR_1	Initialized	—	—	—	—	—	—	—	Initialize
TIER	Initialized	—	—	—	—	—	—	—	Initialize
TCSR	Initialized	—	—	—	—	—	—	—	Initialize
FRC	Initialized	—	—	—	—	—	—	—	Initialize
OCRA	Initialized	—	—	—	—	—	—	—	Initialize
OCRB	Initialized	—	—	—	—	—	—	—	Initialize
TCR	Initialized	—	—	—	—	—	—	—	Initialize
TOCR	Initialized	—	—	—	—	—	—	—	Initialize
ICRA	Initialized	—	—	—	—	—	—	—	Initialize
OCRAR	Initialized	—	—	—	—	—	—	—	Initialize
ICRB	Initialized	—	—	—	—	—	—	—	Initialize
OCRAF	Initialized	—	—	—	—	—	—	—	Initialize
ICRC	Initialized	—	—	—	—	—	—	—	Initialize
OCRDM	Initialized	—	—	—	—	—	—	—	Initialize
ICRD	Initialized	—	—	—	—	—	—	—	Initialize
SMR_2	Initialized	—	—	—	—	—	—	—	Initialize
DACR_0	Initialized	—	Initialized	—	Initialized	Initialized	Initialized	Initialized	Initialize
DADRA_0	Initialized	—	Initialized	—	Initialized	Initialized	Initialized	Initialized	Initialize
BRR_2	Initialized	—	—	—	—	—	—	—	Initialize
SCR_2	Initialized	—	—	—	—	—	—	—	Initialize
TDR_2	Initialized	—	Initialized	—	Initialized	Initialized	Initialized	Initialized	Initialize
SSR_2	Initialized	—	Initialized	—	Initialized	Initialized	Initialized	Initialized	Initialize
RDR_2	Initialized	—	Initialized	—	Initialized	Initialized	Initialized	Initialized	Initialize

P1PCR	Initialized	—	—	—	—	—	—	—	Initiali
P2PCR	Initialized	—	—	—	—	—	—	—	Initiali
P3PCR	Initialized	—	—	—	—	—	—	—	Initiali
P1DDR	Initialized	—	—	—	—	—	—	—	Initiali
P2DDR	Initialized	—	—	—	—	—	—	—	Initiali
P1DR	Initialized	—	—	—	—	—	—	—	Initiali
P2DR	Initialized	—	—	—	—	—	—	—	Initiali
P3DDR	Initialized	—	—	—	—	—	—	—	Initiali
P4DDR	Initialized	—	—	—	—	—	—	—	Initiali
P3DR	Initialized	—	—	—	—	—	—	—	Initiali
P4DR	Initialized	—	—	—	—	—	—	—	Initiali
P5DDR	Initialized	—	—	—	—	—	—	—	Initiali
P6DDR	Initialized	—	—	—	—	—	—	—	Initiali
P5DR	Initialized	—	—	—	—	—	—	—	Initiali
P6DR	Initialized	—	—	—	—	—	—	—	Initiali
PBODR	Initialized	—	—	—	—	—	—	—	Initiali
PBPIN	—	—	—	—	—	—	—	—	—
P8DDR	Initialized	—	—	—	—	—	—	—	Initiali
P7PIN	—	—	—	—	—	—	—	—	—
PBDDR	Initialized	—	—	—	—	—	—	—	Initiali
P8DR	Initialized	—	—	—	—	—	—	—	Initiali
P9DDR	Initialized	—	—	—	—	—	—	—	Initiali
P9DR	Initialized	—	—	—	—	—	—	—	Initiali
IER	Initialized	—	—	—	—	—	—	—	Initiali
STCR	Initialized	—	—	—	—	—	—	—	Initiali

TCORA_1	Initialized	—	—	—	—	—	—	—	Initialize
TCORB_0	Initialized	—	—	—	—	—	—	—	Initialize
TCORB_1	Initialized	—	—	—	—	—	—	—	Initialize
TCNT_0	Initialized	—	—	—	—	—	—	—	Initialize
TCNT_1	Initialized	—	—	—	—	—	—	—	Initialize
PWOERB	Initialized	—	Initialized	—	Initialized	Initialized	Initialized	Initialized	Initialize
PWOERA	Initialized	—	Initialized	—	Initialized	Initialized	Initialized	Initialized	Initialize
PWDPRB	Initialized	—	Initialized	—	Initialized	Initialized	Initialized	Initialized	Initialize
PWDPRA	Initialized	—	Initialized	—	Initialized	Initialized	Initialized	Initialized	Initialize
PWSL	Initialized	—	Initialized	—	Initialized	Initialized	Initialized	Initialized	Initialize
PWDR15 to 0	Initialized	—	Initialized	—	Initialized	Initialized	Initialized	Initialized	Initialize
SMR_0	Initialized	—	—	—	—	—	—	—	Initialize
ICCR_0	Initialized	—	—	—	—	—	—	—	Initialize
BRR_0	Initialized	—	—	—	—	—	—	—	Initialize
ICSR_0	Initialized	—	—	—	—	—	—	—	Initialize
SCR_0	Initialized	—	—	—	—	—	—	—	Initialize
TDR_0	Initialized	—	Initialized	—	Initialized	Initialized	Initialized	Initialized	Initialize
SSR_0	Initialized	—	Initialized	—	Initialized	Initialized	Initialized	Initialized	Initialize
RDR_0	Initialized	—	Initialized	—	Initialized	Initialized	Initialized	Initialized	Initialize
SCMR_0	Initialized	—	—	—	—	—	—	—	Initialize
ICDR_0	—	—	—	—	—	—	—	—	—
SARX_0	Initialized	—	—	—	—	—	—	—	Initialize
ICMR_0	Initialized	—	—	—	—	—	—	—	Initialize
SAR_0	Initialized	—	—	—	—	—	—	—	Initialize
ADDRAH	Initialized	—	Initialized	—	Initialized	Initialized	Initialized	Initialized	Initialize
ADDRAL	Initialized	—	Initialized	—	Initialized	Initialized	Initialized	Initialized	Initialize

TCSR_1	Initialized	—	—	—	—	—	—	—	Initialized
TCNT_1	Initialized	—	—	—	—	—	—	—	Initialized
TCR_X	Initialized	—	—	—	—	—	—	—	Initialized
TCY_Y	Initialized	—	—	—	—	—	—	—	Initialized
KMIMR6	Initialized	—	—	—	—	—	—	—	Initialized
TCSR_X	Initialized	—	—	—	—	—	—	—	Initialized
TCSR_Y	Initialized	—	—	—	—	—	—	—	Initialized
KMPCR6	Initialized	—	—	—	—	—	—	—	Initialized
TICRR	Initialized	—	—	—	—	—	—	—	Initialized
TCORA_Y	Initialized	—	—	—	—	—	—	—	Initialized
KMIMRA	Initialized	—	—	—	—	—	—	—	Initialized
TICRF	Initialized	—	—	—	—	—	—	—	Initialized
TCORB_Y	Initialized	—	—	—	—	—	—	—	Initialized
WUEMR3	Initialized	—	—	—	—	—	—	—	Initialized
TCNT_X	Initialized	—	—	—	—	—	—	—	Initialized
TCNT_Y	Initialized	—	—	—	—	—	—	—	Initialized
TCORC	Initialized	—	—	—	—	—	—	—	Initialized
TISR	Initialized	—	—	—	—	—	—	—	Initialized
TCORA_X	Initialized	—	—	—	—	—	—	—	Initialized
TCORB_X	Initialized	—	—	—	—	—	—	—	Initialized
DADR0	Initialized	—	—	—	—	—	—	—	Initialized
DADR1	Initialized	—	—	—	—	—	—	—	Initialized
DACR	Initialized	—	—	—	—	—	—	—	Initialized
TCONRI	Initialized	—	—	—	—	—	—	—	Initialized
TCONRS	Initialized	—	—	—	—	—	—	—	Initialized

Input voltage (port 7)	V_{in}	–0.3 to AVCC +0.3
Input voltage (port 8, C0 to C5, D6, and D7)	V_{in}	–0.3 to +7.0
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
Operating temperature	T_{opr}	Regular specifications: –20 to +75
		Wide-range specifications: –40 to +85
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.

trigger input voltage	(Ex)TMI1, (Ex)TMI0, (Ex)$\overline{IRQ15}$ to (Ex)$\overline{IRQ2}$, $\overline{IRQ1}$, $\overline{IRQ0}$, $\overline{KIN15}$ to $\overline{KIN0}$, $\overline{WUE15}$ to $\overline{WUE8}$, $\overline{ETRST}$,XTAL, EXCL, $\overline{ADTRG}$		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	$\overline{RES}$, $\overline{STBY}$, NMI, FWE, $\overline{MD2}$, MD1 MD0	(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	
	$\overline{CLKRUN}$, GA20, $\overline{PME}$, $\overline{LSMI}$, LSCI, SERIRQ, LAD3 to LAD0, $\overline{LPCPD}$, LCLK, $\overline{LRESET}$,$\overline{LFRAME}$			VCC × 0.5	—	VCC + 0.3	
	Input pins other than (1) and (2) above			2.2	—	VCC + 0.3	
Input low voltage	$\overline{RES}$, $\overline{STBY}$, NMI, FWE, $\overline{MD2}$, MD1, MD0	(3)	V_{IL}	–0.3	—	VCC × 0.1	
	EXTAL			–0.3	—	VCC × 0.1	f >
				–0.3	—	VCC × 0.2	f ≤
	Port 7			–0.3	—	AVCC × 0.2	
	$\overline{CLKRUN}$, GA20, $\overline{PME}$, $\overline{LSMI}$, LSCI, SERIRQ, LAD3 to LAD0, $\overline{LPCPD}$, LCLK, $\overline{LRESET}$,$\overline{LFRAME}$			–0.3	—	VCC × 0.3	
	Input pins other than (1) and (3) above			–0.3	—	VCC × 0.2	

voltage	$\overline{\text{CLKRUN}}$, GA20, $\overline{\text{PME}}$, $\overline{\text{LSMI}}$, LSCI, SERIRQ, LAD3 to LAD0	—	—	VCC × 0.1	I
	Output pins other than (5) above	—	—	0.4	I
	Ports 1, 2, and 3	—	—	1.0	I

Table 25.2 DC Characteristics (2)

Conditions: VCC = 3.0 V to 3.6 V, AVCC*[1] = 3.0 V to 3.6 V,
AVref*[1] = 3.0 V to AVCC, VSS = AVSS*[1] = 0 V

Item		Symbol	Min.	Typ.	Max.	Unit	Test Conditions
Input leakage current	$\overline{\text{RES}}$, $\overline{\text{STBY}}$, NMI, FWE, $\overline{\text{MD2}}$, MD1, MD0, PFSEL	$\lvert I_{in} \rvert$	—	—	1.0	µA	V_{IN} = 0.5 to VCC – 0
	Port 7		—	—	1.0		V_{IN} = 0.5 to AVCC –
Three-state leakage current (off state)	Ports 1 to 6 Ports 8 to F	$\lvert I_{TSI} \rvert$	—	—	1.0		V_{IN} = 0.5 to VCC – 0
Input pull-up MOS current	Ports 1 to 3	$-I_P$	5	—	150		V_{IN} = 0 V
	Ports 6 (P6PUE=0), A, D		30	—	300		
	Port 6 (P6PUE=1)		3	—	100		
Current consumption *[5]	Normal operation	I_{CC}	—	43	55	mA	f = 33 MHz, high-sp All modules operat
	Sleep mode		—	30	40		f = 33 MHz
	Standby mode*[6]		—	38	90	µA	Ta ≤ 50 °C
			—	—	120		50 °C < Ta

capacitance				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 or D/A co not used.

Even if the A/D converter or D/A converter is not used, apply a value in the range from 3 V to the AVCC and AVref pins by connecting them to the power supply (VCC). The relat between these two pins should be AVref ≤ AVCC.

2. When noise cancel has been enabled.
3. An external pull-up resistor is necessary to provide high-level output from SCL5 to SCL0 SDA5 to SDA0 (ICE bit in ICCR is 1).
4. Port 8, C0 to C5, D6, and D7 are NMOS push-pull outputs.

Port 8, C0 to C5, D6, D7, and SCK0 to SCK2 (ICE bit in ICCR = 0) high levels are driver NMOS. An external pull-up resistor is necessary to provide high-level output from these they are used as an output.

5. Current consumption values are for V_{IH} min = VCC – 0.2 V and V_{IL} max = 0.2 V with all o unloaded and the on-chip pull-up MOSs in the off state.
6. When VCC = 3.0 V, V_{IH} min = VCC – 0.2 V, and V_{IL} max = 0.2 V.

current (per pin)					
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 25.3.
2. When driving a Darlington transistor or LED, always insert a current-limiting resistor in t line, as show in figures 25.1 and 25.2.

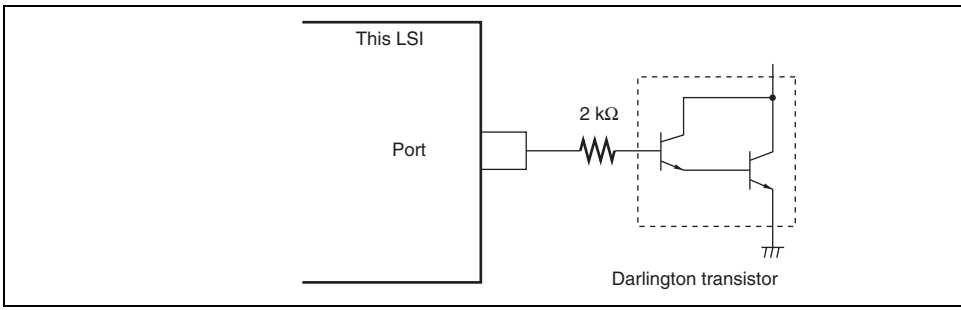


Figure 25.1 Darlington Transistor Drive Circuit (Example)

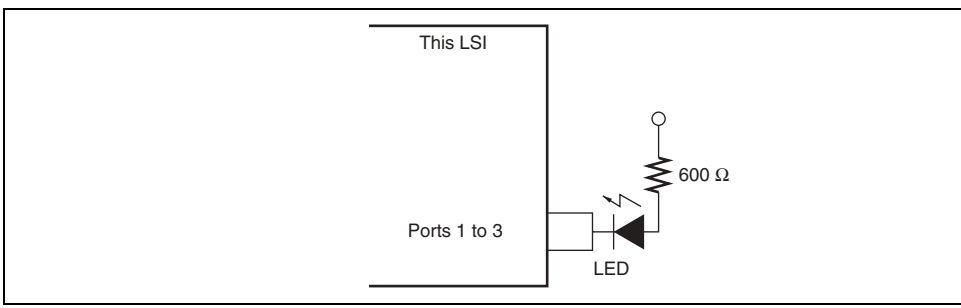


Figure 25.2 LED Drive Circuit (Example)

Figure 25.3 Output Load Circuit

25.3.1 Clock Timing

Table 25.4 shows the clock timing. The clock timing specified here covers clock output (
clock pulse generator (crystal) and external clock input (EXTAL pin) oscillation stabiliza
times. For details of external clock input (EXTAL pin and EXCL pin) timing, see table 2
25.6.

Table 25.4 Clock Timing

Condition: VCC = 3.0 V to 3.6 V, VSS = 0 V, ϕ = 5 MHz to 33 MHz

Item	Symbol	Min.	Max.	Unit	Ref
Clock cycle time	t_{cyc}	30	200	ns	Figu
Clock high level pulse width	t_{CH}	10	—		
Clock low level pulse width	t_{CL}	10	—		
Clock rise time	t_{Cr}	—	5		
Clock fall time	t_{Cf}	—	5		
Reset oscillation stabilization (crystal)	t_{OSC1}	10	—	ms	Figu
Software standby oscillation stabilization time (crystal)	t_{OSC2}	8	—		Figu

Clock high level pulse width	t_{CH}	0.4	0.6	t_{cyc}	
External clock output stabilization delay time	t_{DEXT}*	500	—	µs	Fig

Note: * t_{DEXT} includes a $\overline{RES}$ pulse width (t_{RESW}).

Table 25.6 Subclock Input Conditions

Condition: VCC = 3.0 V to 3.6 V, VSS = 0 V, ϕ SUB = 32.768 kHz, 5 MHz to 33 MHz

Item	Symbol	Min.	Typ.	Max.	Unit	Me nt
Subclock input low level pulse width	t_{EXCLL}	—	15.26	—	µs	Fig
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}	Fig
Clock high level pulse width	t_{CH}	0.4	—	0.6	t_{cyc}	

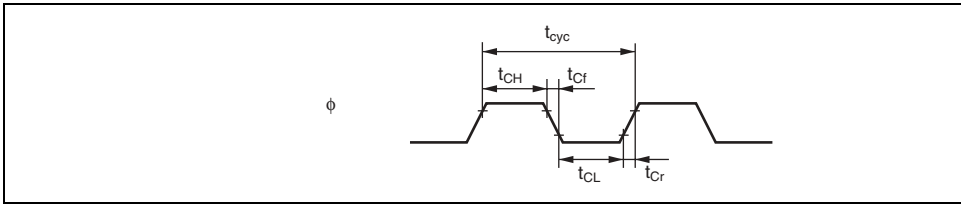


Figure 25.4 System Clock Timing

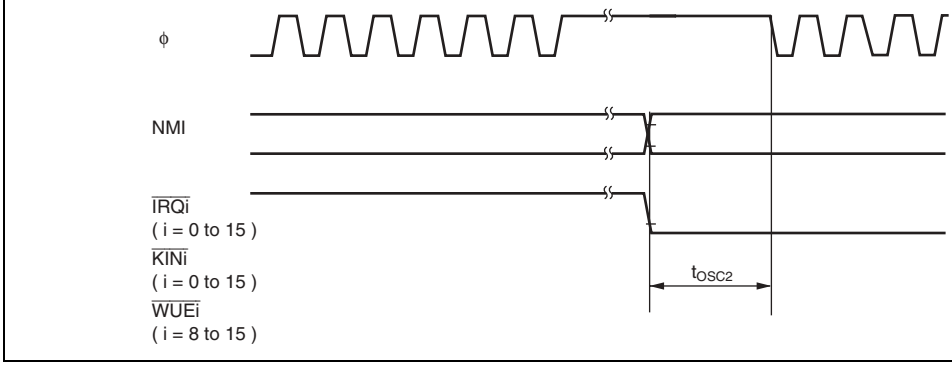


Figure 25.6 Oscillation Stabilization Timing (Exiting Software Standby Mod

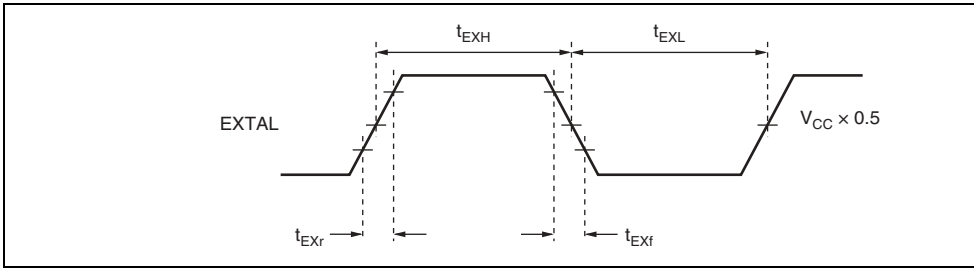
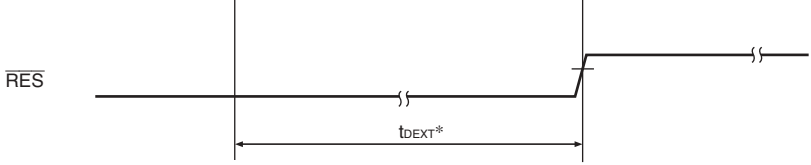


Figure 25.7 External Clock Input Timing



Note: The external clock output stabilization delay time (t_{DEXT}) includes a $\overline{RES}$ pulse width (t_{RESW}).

Figure 25.8 Timing of External Clock Output Stabilization Delay Time

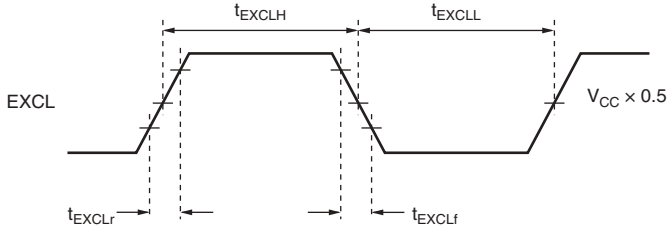


Figure 25.9 Subclock Input Timing

NMI hold time	t_{NMIH}	10	—
NMI pulse width (exiting software standby mode)	t_{NMIW}	200	—
IRQ setup time ($\overline{IRQ15}$ to $\overline{IRQ0}$, $\overline{KIN15}$ to $\overline{KIN0}$, $\overline{WUE15}$ to $\overline{WUE8}$)	t_{IRQS}	150	—
IRQ hold time ($\overline{IRQ15}$ to $\overline{IRQ0}$, $\overline{KIN15}$ to $\overline{KIN0}$, $\overline{WUE15}$ to $\overline{WUE8}$)	t_{IRQH}	10	—
IRQ pulse width ($\overline{IRQ15}$ to $\overline{IRQ0}$, $\overline{KIN15}$ to $\overline{KIN0}$, $\overline{WUE15}$ to $\overline{WUE8}$) (exiting software standby mode)	t_{IRQW}	200	—

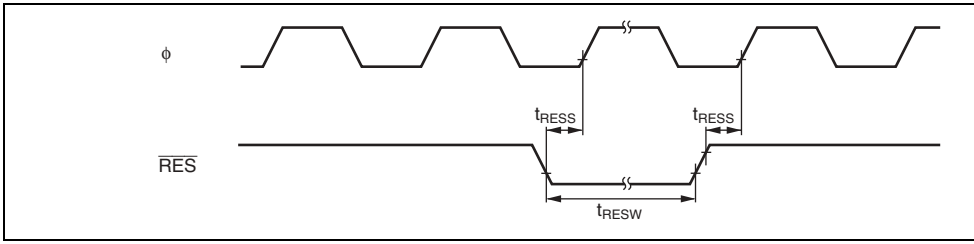


Figure 25.10 Reset Input Timing

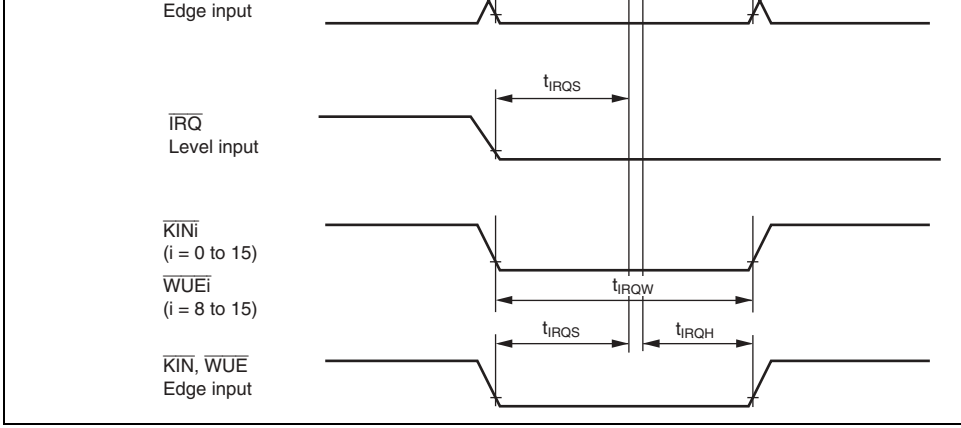
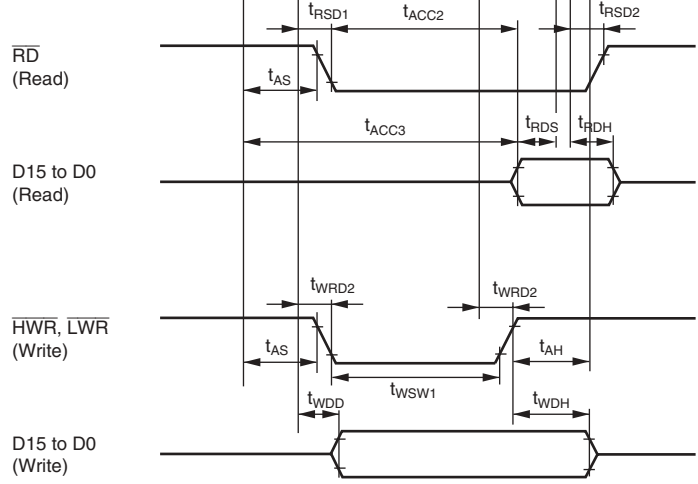


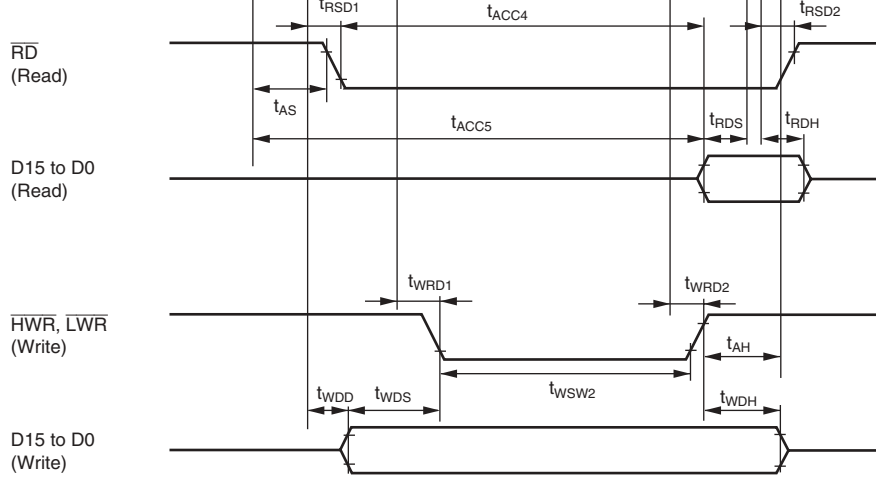
Figure 25.11 Interrupt Input Timing

	t_{AH}	10	
$\overline{CS}$ delay time ($\overline{IOS}$, $\overline{CS256}$, $\overline{CPCS1}$)	t_{CSD}	—	15
$\overline{AS}$ delay time	t_{ASD}	—	15
$\overline{RD}$ delay time 1	t_{RSD1}	—	15
$\overline{RD}$ delay time 2	t_{RSD2}	—	15
Read data setup time	t_{RDS}	15	—
Read data hold time	t_{RDH}	0	—
Read data access time 1	t_{ACC1}	—	$1.0 \times t_{cyc} - 30$
Read data access time 2	t_{ACC2}	—	$1.5 \times t_{cyc} - 25$
Read data access time 3	t_{ACC3}	—	$2.0 \times t_{cyc} - 30$
Read data access time 4	t_{ACC4}	—	$2.5 \times t_{cyc} - 25$
Read data access time 5	t_{ACC5}	—	$3.0 \times t_{cyc} - 30$
$\overline{WR}$ delay time 1	t_{WRD1}	—	15
$\overline{WR}$ delay time 2	t_{WRD2}	—	15
$\overline{WR}$ pulse width 1	t_{WSW1}	$1.0 \times t_{cyc} - 20$	—
$\overline{WR}$ pulse width 2	t_{WSW2}	$1.5 \times t_{cyc} - 20$	—
Write data delay time	t_{WDD}	—	25
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}	25	—
$\overline{WAIT}$ hold time	t_{WTH}	5	—



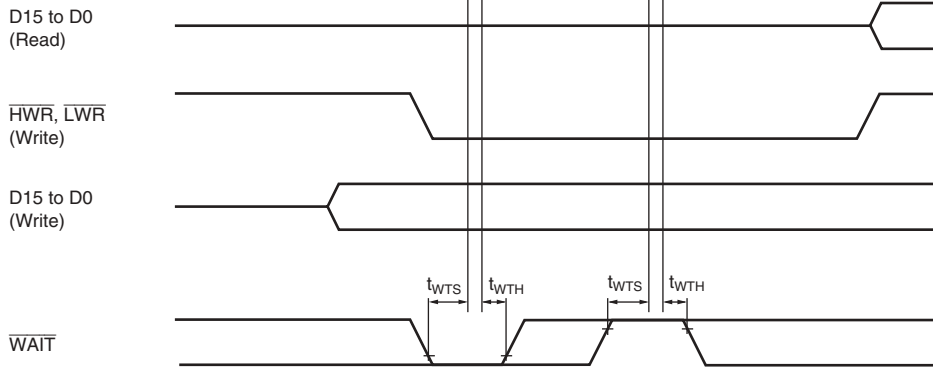
Note: * AS is multiplexed with IOS. Either the AS or IOS function can be selected by the IOSE bit of SYS

Figure 25.12 Basic Bus Timing/2-State Access



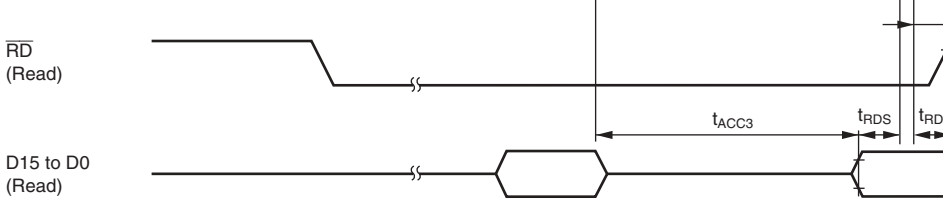
Note: * AS is multiplexed with IOS. Either the AS or IOS function can be selected by the IOSE bit of SYS

Figure 25.13 Basic Bus Timing/3-State Access



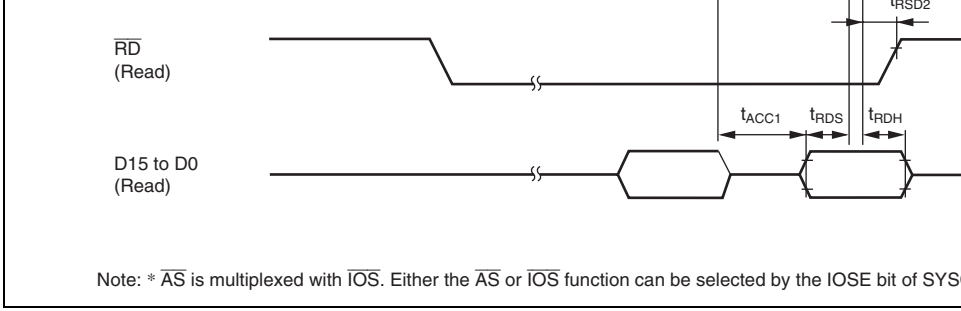
Note: * AS is multiplexed with IOS. Either the AS or IOS function can be selected by the IOSE bit of SYSCR.

Figure 25.14 Basic Bus Timing/3-State Access with One Wait State



Note: * AS is multiplexed with IOS. Either the AS or IOS function can be selected by the IOSE bit of SYSCR.

Figure 25.15 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 SYS

Figure 25.16 Burst ROM Access Timing/1-State Access

$\overline{AH}$	t_{AH2}	t_{cyc}	
$\overline{CS}$ delay time ($\overline{IOS}$, $\overline{CS256}$, $\overline{CPCS1}$)	t_{CSD}	—	15
$\overline{AH}$ delay time	t_{AHD}	—	15
$\overline{RD}$ delay time 1	t_{RSD1}	—	15
$\overline{RD}$ delay time 2	t_{RSD2}	—	15
Read data setup time	t_{RDS}	15	—
Read data hold time	t_{RDH}	0	—
Read data access time 2	t_{ACC2}	—	$1.5 \times t_{cyc} - 25$
Read data access time 4	t_{ACC4}	—	$2.5 \times t_{cyc} - 25$
Read data access time 6	t_{ACC6}	—	$3.5 \times t_{cyc} - 25$
Read data access time 7	t_{ACC7}	—	$4.5 \times t_{cyc} - 25$
$\overline{WR}$ delay time 1	t_{WRD1}	—	15
$\overline{WR}$ delay time 2	t_{WRD2}	—	15
$\overline{WR}$ pulse width time 1	t_{WSW1}	$1.0 \times t_{cyc} - 20$	—
$\overline{WR}$ pulse width time 2	t_{WSW2}	$1.5 \times t_{cyc} - 20$	—
Write data delay time	t_{WDD}	—	25
Write data setup time	t_{WDS}	0	—
Write data hold time	t_{WDH}	$0.5 \times t_{cyc} - 5$	—

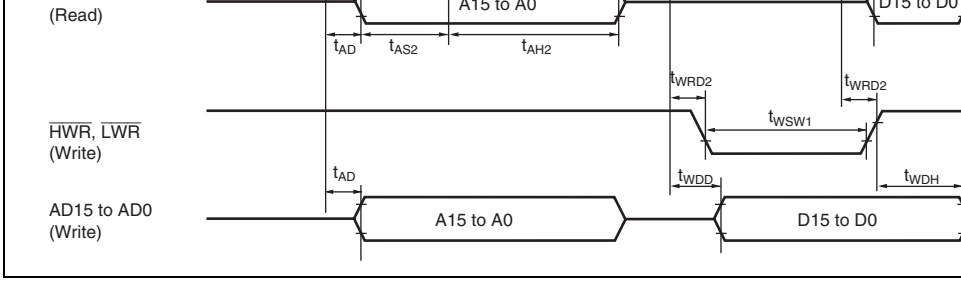


Figure 25.17 Multiplex Bus Timing/Data 2-State Access

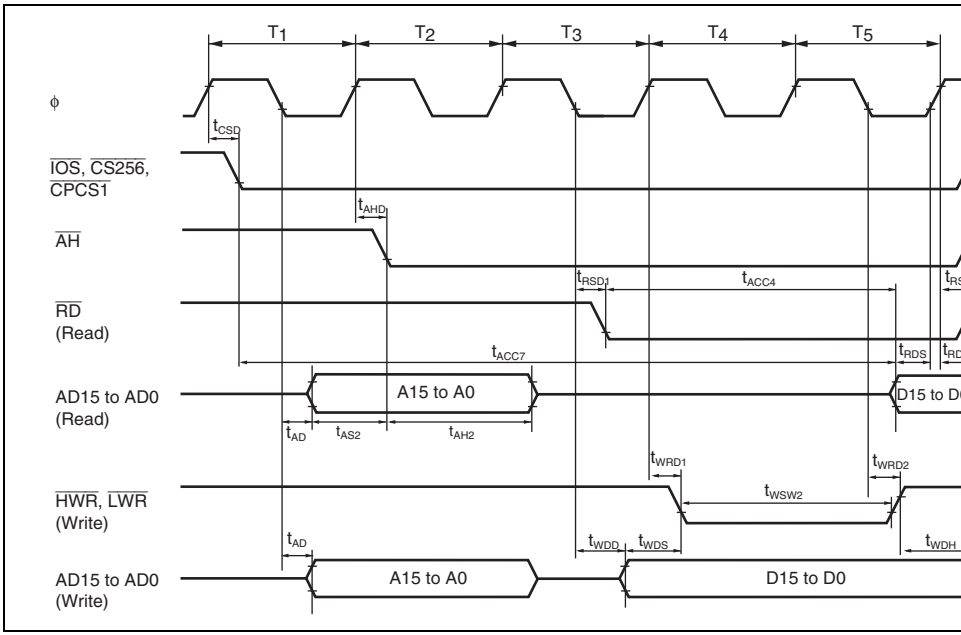


Figure 25.18 Multiplex Bus Timing/Data 3-State Access

	pulse width	Both edges	t_{FTCWL}	2.5	—		
TMR	Timer output delay time		t_{TMOD}	—	30	ns	Figure 25.2
	Timer reset input setup time		t_{TMRS}	20	—		Figure 25.2
	Timer clock input setup time		t_{TMCS}	20	—		Figure 25.2
	Timer clock pulse width	Single edge	t_{TMCWH}	1.5	—	t_{cyc}	
		Both edges	t_{TMCWL}	2.5	—		
PWM, PWMX	Timer output delay time		t_{PWOD}	—	30	ns	Figure 25.2
SCI	Input clock cycle	Asynchronous	t_{Scyc}	4	—	t_{cyc}	Figure 25.2
		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}	—	30	ns	Figure 25.2
	Receive data setup time (synchronous)		t_{RXS}	20	—		
	Receive data hold time (synchronous)		t_{RXH}	20	—		
A/D converter	Trigger input setup time		t_{TRGS}	20	—	ns	Figure 25.2
WDT	$\overline{\text{RESO}}$ output delay time		t_{RESD}	—	200	ns	Figure 25.2
	$\overline{\text{RESO}}$ output pulse width		t_{RESOW}	132	—	t_{cyc}	

Note: * Only the peripheral modules that can be used in subclock operation.

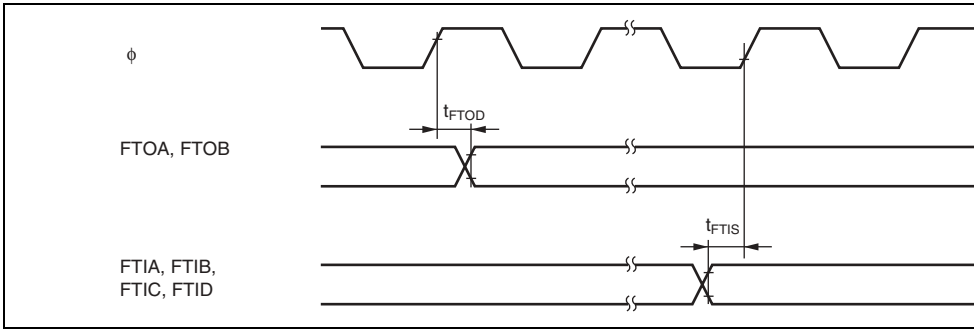


Figure 25.20 FRT Input/Output Timing

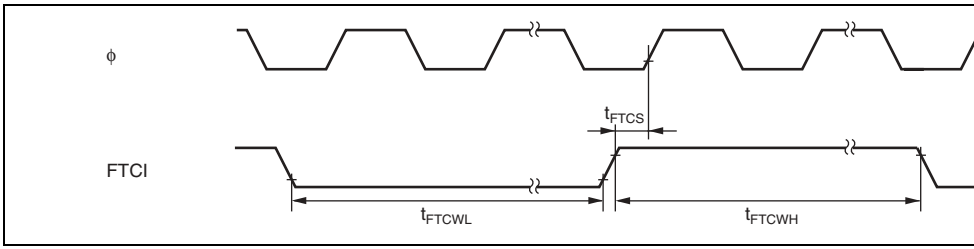


Figure 25.21 FRT Clock Input Timing

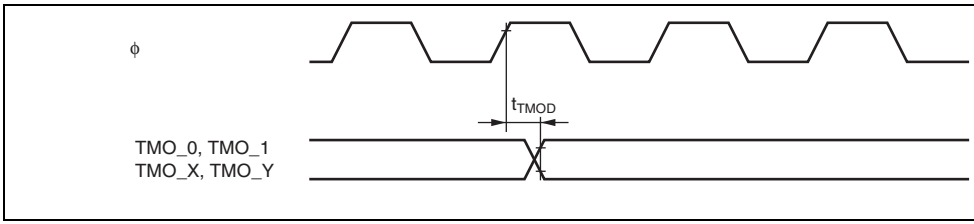


Figure 25.22 8-Bit Timer Output Timing

Figure 25.24 8-Bit Timer Reset Input Timing

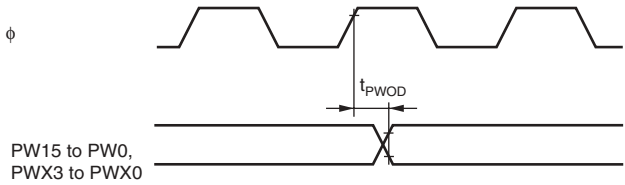


Figure 25.25 PWM, PWMX Output Timing

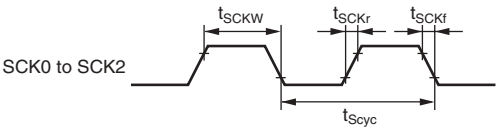


Figure 25.26 SCK Clock Input Timing

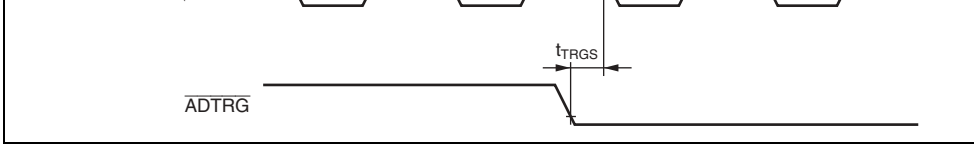


Figure 25.28 A/D Converter External Trigger Input Timing

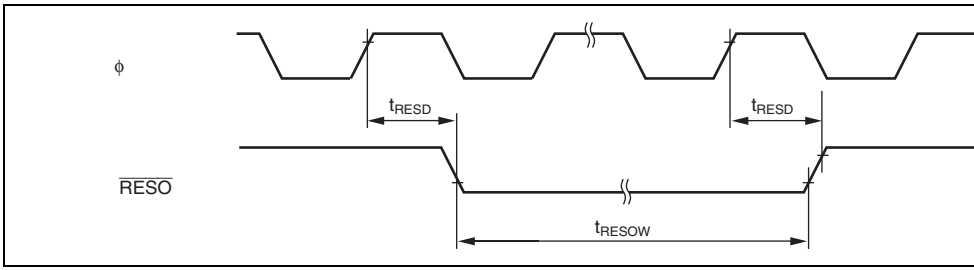


Figure 25.29 WDT Output Timing ($\overline{\text{RESO}}$)

pulse elimination time					
SDA input bus free time	t_{BUF}	5	—	—	
Start condition input hold time	t_{STAH}	3	—	—	
Retransmission 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: Retransmission start condition

Figure 25.30 I²C Bus Interface Input/Output Timing

Table 25.12 LPC Module Timing

Conditions: VCC = 3.0 V to 3.6V, VSS = 0 V, ϕ = 5 MHz to 33 MHz

Item	Symbol	Min.	Typ.	Max.	Unit	Test Conditions
Input clock cycle	t_{Lcyc}	30	—	—	ns	Figure 25.31
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	—	—		

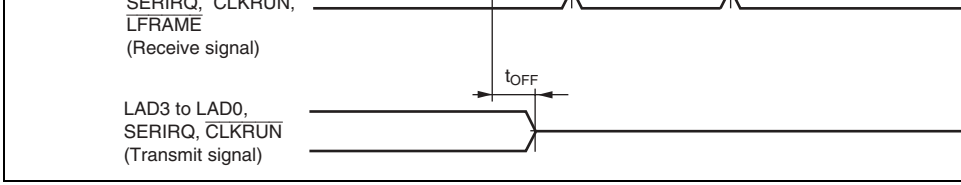


Figure 25.31 LPC Interface (LPC) Timing

Table 25.13 JTAG Timing

Condition: VCC = 3.0 V to 3.6 V, VSS = 0 V, ϕ = 5 MHz to 33 MHz

Item	Symbol	Min.	Max.	Unit	Test Condi
ETCK clock cycle time	t_{TCKcyc}	40*	200*	ns	Figure 25.3
ETCK clock high pulse width	t_{TCKH}	15	—		
ETCK clock low pulse width	t_{TCKL}	15	—		
ETCK clock rise time	t_{TCKr}	—	5		
ETCK clock fall time	t_{TCKf}	—	5		
$\overline{\text{ETRST}}$ pulse width	t_{TRSTW}	20	—	t_{cyc}	Figure 25.3
Reset hold transition pulse width	t_{RSTHW}	3	—		
ETMS setup time	t_{TMSS}	20	—	ns	Figure 25.3
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 $t_{cyc} \leq t_{TCKcyc}$

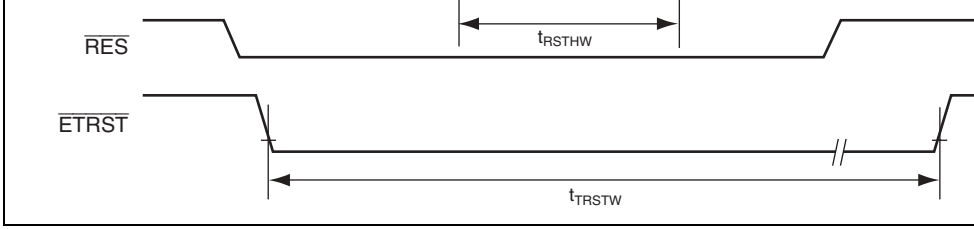


Figure 25.33 Reset Hold Timing

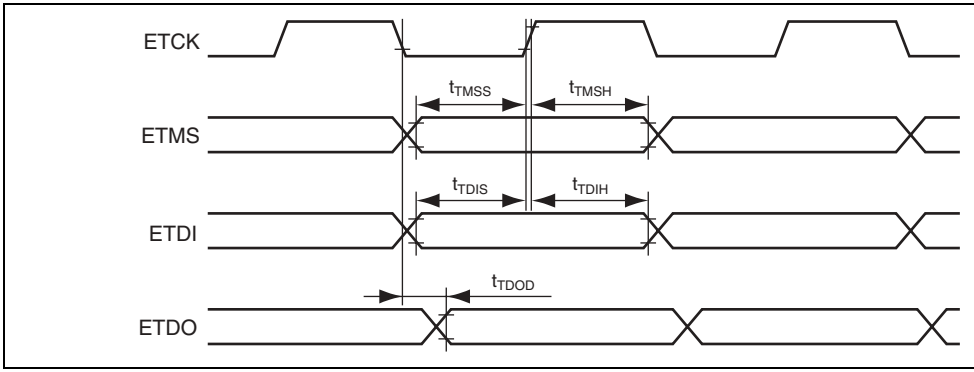


Figure 25.34 JTAG Input/Output Timing

	Condition A			Condition B			
Item	**Min.**	**Typ.**	**Max.**	**Min.**	**Typ.**	**Max.**	**Unit**
Resolution		10			10		Bits
Conversion time	—	—	8.38*[1]	—	—	8.06*[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 134 st
2. Value when using the maximum operating frequency in single mode of 266 st

Load resistance 4 MΩ	—	—	±2.0

Item	Symbol	Min.	Typ.	Max.	Unit	Test Cond
Programming time*1*2*4	t_P	—	3	30	ms/128 bytes	
Erase time*1*2*4	t_E	—	80	800	ms/4-kbyte block	
		—	500	5000	ms/32-kbyte block	
		—	1000	10000	ms/64-kbyte block	
Programming time (total)*1*2*4	Σt_P	—	5	15	s/256 kbytes	Ta =
Erase time (total)*1*2*4	Σt_E	—	5	15	s/256 kbytes	Ta =
Programming and Erase time (total)*1*2*4	Σt_{PE}	—	10	30	s/256 kbytes	Ta =
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 ran 1 to the minimum number.)
4. This value indicates the characteristics while the flash memory is reprogramm the specified range (including the minimum number).
5. Reprogramming count in each erase block.

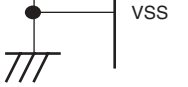
Reprogramming count*[5]	N_{WEC}	100*[3]	1000	—	Times
Data retention time*[4]	t_{DRP}	10	—	—	Years

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.

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 ran 1 to the minimum number.)
4. This value indicates the characteristics while the flash memory is reprogramm 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 VC
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 25.35 Connection of VCL Capacitor

	(EXPE = 0)							I/O port
Port 3 D15 to D8	(EXPE = 1)	T	T	T	T	T	T	D15 to D8
	(EXPE = 0)			kept	kept	kept	kept	I/O port
Port 4	(EXPE = 1)	T	T	kept	kept	kept	kept	I/O port
	(EXPE = 0)							
Port 5	(EXPE = 1)	T	T	kept	kept	kept	kept	I/O port
	(EXPE = 0)							
Port 6 D7 to D0	(EXPE = 1)	T	T	kept	kept	kept	kept	D7 to D0/ I/O port
	(EXPE = 0)							I/O port
Port 7	(EXPE = 1)	T	T	T	T	T	T	Input port
	(EXPE = 0)							
Port 8	(EXPE = 1)	T	T	kept	kept	kept	kept	I/O port
	(EXPE = 0)							
Port 97 $\overline{\text{WAIT}}$, $\overline{\text{CS256}}$	(EXPE = 1)	T	T	T/kept	T/kept	T/kept	T/kept	$\overline{\text{WAIT}}$/$\overline{\text{CS256}}$/ I/O port
	(EXPE = 0)			kept	kept	kept	kept	I/O port
Port 96 ϕ, EXCL	(EXPE = 1)	T	T	[DDR = 1] H [DDR = 0] T	EXCL input	[DDR = 1] Clock output [DDR = 0] T	EXCL input	EXCL input
	(EXPE = 0)							

	(EXPE = 0)			kept	kept	kept	kept	I/O port	I
Port A A23 to A16	(EXPE = 1)	T	T	kept*	kept*	kept*	kept*	Address output/ I/O port	A o I/
	(EXPE = 0)							I/O port	I/
Port B	(EXPE = 1)	T	T	kept	kept	kept	kept	I/O port	I/
	(EXPE = 0)								
Port C	(EXPE = 1)	T	T	kept	kept	kept	kept	I/O port	I/
	(EXPE = 0)								
Port D	(EXPE = 1)	T	T	kept	kept	kept	kept	I/O port	I/
	(EXPE = 0)								
Port E	(EXPE = 1)	T	T	kept	kept	kept	kept	I/O port	I/
	(EXPE = 0)								
Port F	(EXPE = 1)	T	T	kept	kept	kept	kept	I/O port	I/
	(EXPE = 0)								

Legend

H: High level

L: Low level

T: High impedance

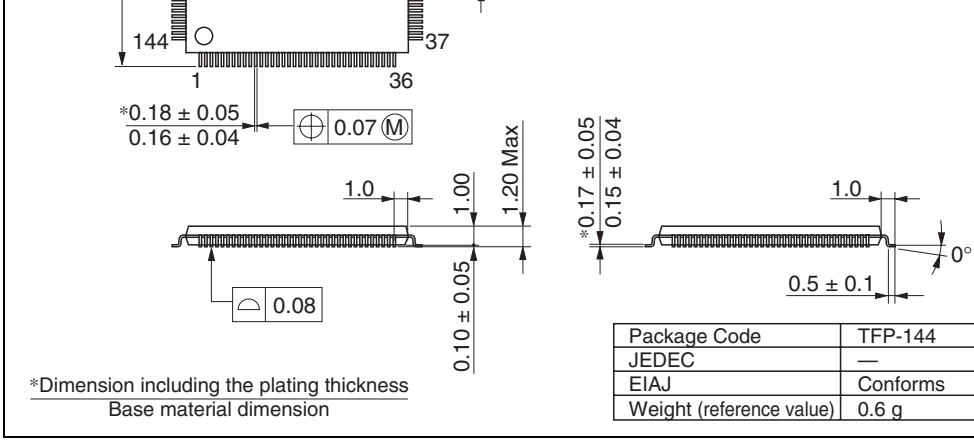
kept: Input ports are in the high-impedance state (when DDR = 0 and PCR = 1, the inpu MOS remains on).

Output ports maintain their previous state.

Depending on the pins, the on-chip peripheral modules may be initialized and the function determined by DDR and DR.

DDR: Data direction register

Note: * In the case of address output, the last address accessed is retained.



Package Code	TFP-144
JEDEC	—
EIAJ	Conforms
Weight (reference value)	0.6 g

Figure C.1 Package Dimensions (TFP-144)

TMR_0 compare-match occurrences (compa
count mode)

Section 12.3.4 Timer Control register (TCR) 312, 313

Description amended.

Table 12.2 Clock Input to TCNT and Count Condition

Channel	TCR CKS2	CKS1	CKS0	Description
TMR_Y	0	0	0	Disables clock inp
	0	0	1	Increments at falli of internal clock φ/
	0	1	0	Increments at falli of internal clock φ/
	0	1	1	Increments at falli of internal clock φ/
	1	0	0	Setting prohibited
TMR_X	0	0	0	Disables clock inp
	0	0	1	Increments at falli of internal clock φ
	0	1	0	Increments at falli of internal clock φ/
	0	1	1	Increments at falli of internal clock φ/
	1	0	0	Setting prohibited

Note: * If the TMR_0 clock input is set as the TCN overflow signal and the TMR_1 clock inpu the TCNT_0 compare-match signal simult a count-up clock cannot be generated. Se these conditions should therefore be avoi

3, 2	—	All 0	R/W	Reserved The initial value should changed.

14.3.11 Serial Enhanced Mode Register_0 and 2 (SEMR_0 and SEMR_2) — 377

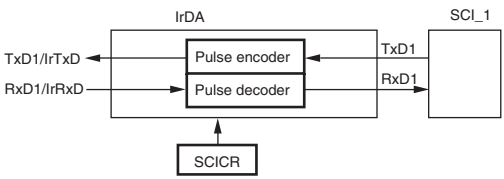
Description amended.

Bit	Bit Name	Initial Value	R/W	Description
4	ACS4	0	R/W	0011: Average transfer
2	ACS2	0	R/W	operation at 720
1	ACS1	0	R/W	when the system
0	ACS0	0	R/W	frequency is 32 M (operated using t basic clock with a frequency 16 time transfer clock freq

14.8 IrDA Operation — 417

Figure 14.36 IrDA Block Diagram

Figure amended



21.5.1 Supported Instructions

IDCODE: Instruction Code: B'1110

Section 24 List of Registers

24.2 Register Bits

766

Description amended.

Register Abbreviation	Bit 7	Bit 6	Bit 5	Bit 4	Bit 3	Bit 2	B
SCICR	IrE	IrCKS2	IrCKS1	IrCKS0	—	—	—

Section 25 Electrical Characteristics

25.2 DC Characteristics

Table 25.2 DC Characteristics (1)

784

Amended

Item			Symbol	Min.	Typ.	Max.	T C
Input low voltage	$\overline{RES}$, $\overline{STBY}$, NMI, FWE, $\overline{MD2}$, MD1, MD0	(3)	V_{IL}	–0.3	—	VCC × 0.1	
	EXTAL			–0.3	—	VCC × 0.1	f
				–0.3	—	VCC × 0.2	f
	Port 7			–0.3	—	AVCC × 0.2	

25.3.3 Bus Timing Table 25.8 Bus Timing 25.3.4 Multiplex Bus Timing Table 25.9 Multiplex Bus Timing		**Item** **Symbol** **Min.** **Max.** Write data hold time t_{WDH} $0.5 \times t_{cyc} - 5$ —
25.6 Flash Memory Characteristics Table 25.16 Flash Memory Characteristics	813 to 815	Description amended and added. Ta = 0°C to +75°C (operating temperature range for programming/erasing in regular specifications) Ta = 0°C to +85°C (operating temperature range for programming/erasing in wide-range specifications) **Item** **Symbol** **Min.** **Typ.** **Max.** **Unit** **Tes Co** Reprogramming count*5 N_{WEC} 100*3 1000 — Times Notes: 5. Reprogramming count in each erase block.

RENESAS

Renesas 16-Bit Single-Chip Microcomputer
Hardware Manual
H8S/2168 Group

Publication Date: 1st Edition, Dec, 2002
Rev.3.00, Mar 12, 2004
Published by: Sales Strategic Planning Div.
Renesas Technology Corp.
Edited by: Technical Documentation & Information Department
Renesas Kodaira Semiconductor Co., Ltd.

Renesas Technology Corp. Sales Strategic Planning Div. Nippon Bldg., 2-6-2, Ohte-machi, Chiyoda-ku, Tokyo 100-0004, J

RENESAS SALES OFFICES

http://www.re

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, United Kingdom
Tel: <44> (1628) 585 100, Fax: <44> (1628) 585 900

Renesas Technology Europe GmbH
Dornacher Str. 3, D-85622 Feldkirchen, Germany
Tel: <49> (89) 380 70 0, Fax: <49> (89) 929 30 11

Renesas Technology Hong Kong Ltd.
7/F., North Tower, World Finance Centre, Harbour City, Canton Road, Hong Kong
Tel: <852> 2265-6688, Fax: <852> 2375-6836

Renesas Technology Taiwan Co., Ltd.
FL 10, #99, Fu-Hsing N. Rd., Taipei, Taiwan
Tel: <886> (2) 2715-2888, Fax: <886> (2) 2713-2999

Renesas Technology (Shanghai) Co., Ltd.
26/F., Ruijin Building, No.205 Maoming Road (S), Shanghai 200020, China
Tel: <86> (21) 6472-1001, Fax: <86> (21) 6415-2952

Renesas Technology Singapore Pte. Ltd.
1, Harbour Front Avenue, #06-10, Keppel Bay Tower, Singapore 098632
Tel: <65> 6213-0200, Fax: <65> 6278-8001

H8S/2168Group
Hardware Manual

Renesas Electronics Corporation
1753, Shimonumabe, Nakahara-ku, Kawasaki-shi, Kanagawa 211-8668 Japan

REJ09B0078-0300Z

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 :

MB90F036APMC-GSE1 MB90F342CASPMC-GSE1 MB90F345CESPMC-GE1 MB90F349CAPFR-GSE1 MB90F428GCPFR-GSE1 MB90F462APFM-GE1 MB90F462APMC-G-SNE1 MB90F497GPF-GE1 MB90F546GSPFR-GE1 MB90F947APFR-GS-SPE1 MB96F346RSBPMC-GS-N2E2 MB96F683RBPMC-GSAE1 R5F11BGEAFB#30 DF3026XBL25V S912ZVFP64F1VLL R4F24268NVRFQV R5F107DEGSP#X0 R5F11B7EANA#U0 R5F21172DSP#U0 M30622F8PGP#U3C MB90092PF-G-BNDE1 MB90F335APMC1-G-SPE1 MB90F342CASPFR-GS-N2E1 MB90F345CAPFR-GSE1 MB90F543GPF-GE1 MB90F546GSPF-GE1 MB90F568PMCR-GE1 MB90F594APFR-GE1 MB90F882ASPMC-GE1 MB96F346RSAPQCR-GS-N2E2 MB96F387RSBPMC-GSE2 MB96F387RSBPMC-GS-N2E2 MB96F395RSAPMC-GSE2 MB96F623RBPMC1-GSE1 MB96F646RBPMC-GSE1 XE167F96F66LACFXUMA1 MB96F696RBPMC-GSAE1 MB96F018RBPMC-GSE1 MB90F962SPMCR-GE1 MB90F867ASPFR-GE1 MB90F543GPF-G-FLE1 MB90F345CESPF-GE1 M30290FCHP#U3A DF2239FA20IV HD64F3672FPV R5F104AEASP#V0 R5F100BCANA#U0 R5F100BFANA#U0 S9S12H256J2VFVER R5F100ACASP#V0