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/2164 Group

Hardware Manual

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

H8S/2164 R4F2164

Renesas Electronics
www.renesas.com

Rev.2.00 2009.09

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

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

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

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

2. Processing at Power-on

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

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

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

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

3. Prohibition of Access to Reserved Addresses

   Access to reserved addresses is prohibited.

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

4. Clock Signals

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

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

5. Differences between Products

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

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

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

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

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

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

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

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

11. Index

Objective: This manual was written to explain the hardware functions and electrical characteristics of the H8S/2164 Group to the target users.
Refer to the H8S/2600 Series, H8S/2000 Series Software Manual for a detailed description of the instruction set.

Notes on reading this manual:

- In order to understand the overall functions of the chip
  Read the manual according to the contents. This manual can be roughly categorized on the CPU, system control functions, peripheral functions and electrical characteris
- In order to understand the details of the CPU's functions
  Read the H8S/2600 Series, H8S/2000 Series Software Manual.
- In order to understand the details of a register when its name is known
  Read the index that is the final part of the manual to find the page number of the entr register. The addresses, bits, and initial values of the registers are summarized in sec List of Registers.

Examples: Register name: The following notation is used for cases when the sar similar function, e.g. 16-bit timer pulse unit or serial communication, is implemented on more than one ch XXX_N (XXX is the register name and N is the char number)

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

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

User's Manual	
H8S, H8/300 Series Simulator/Debugger User's Manual	REJ10B021
High-performance Embedded Workshop User's Manual	REJ10J200

Section 18 LPC Interface (LPC)

Section 21 Flash Memory

Appendix

Section 7 Data Transfer Controller (DTC)

Section 8 I/O Ports

Section 10 16-Bit Free-Running Timer (FRT)

Section 11 8-Bit Timer (TMR)

Section 12 Watchdog Timer (WDT)

Section 13 Serial Communication Interface (SCI)

Section 26 Electrical Characteristics

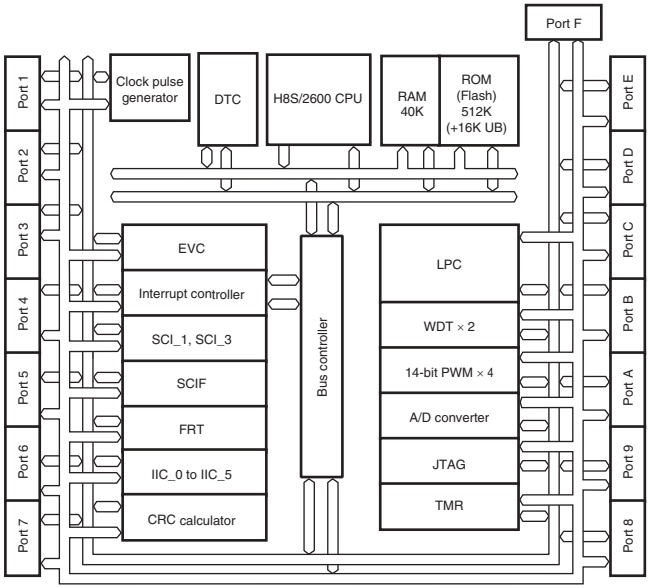
Appendix

- Various peripheral functions
  Data transfer controller (DTC)
  14-bit PWM timer (PWMX)
  16-bit free-running timer (FRT)
  8-bit timer (TMR)
  Watchdog timer (WDT)
  Asynchronous or synchronous serial communication interface (SCI)
  CRC operation circuit (CRC)
  Serial communication interface with FIFO (SCIF)
  I^2C bus interface (IIC)
  LPC interface (LPC)
  10-bit A/D converter
  Boundary scan (JTAG)
  Clock pulse generator

- On-chip memory

ROM Type	Model	ROM	RAM	Remark
Flash memory Version	R4F2164	512 Kbytes	40 Kbytes	

1.2 Block Diagram



[Legend]

CPU:	Central processing unit
DTC:	Data transfer controller
EVC:	Event counter
SCI:	Serial communication interface
SCIF:	Serial communication interface with FIFO
FRT:	16-bit free running timer
IIC:	I^2C bus interface
LPC:	LPC interface
WDT:	Watchdog timer
JTAG:	Boundary scan
TMR:	8-bit timer

Figure 1.1 Internal Block Diagram

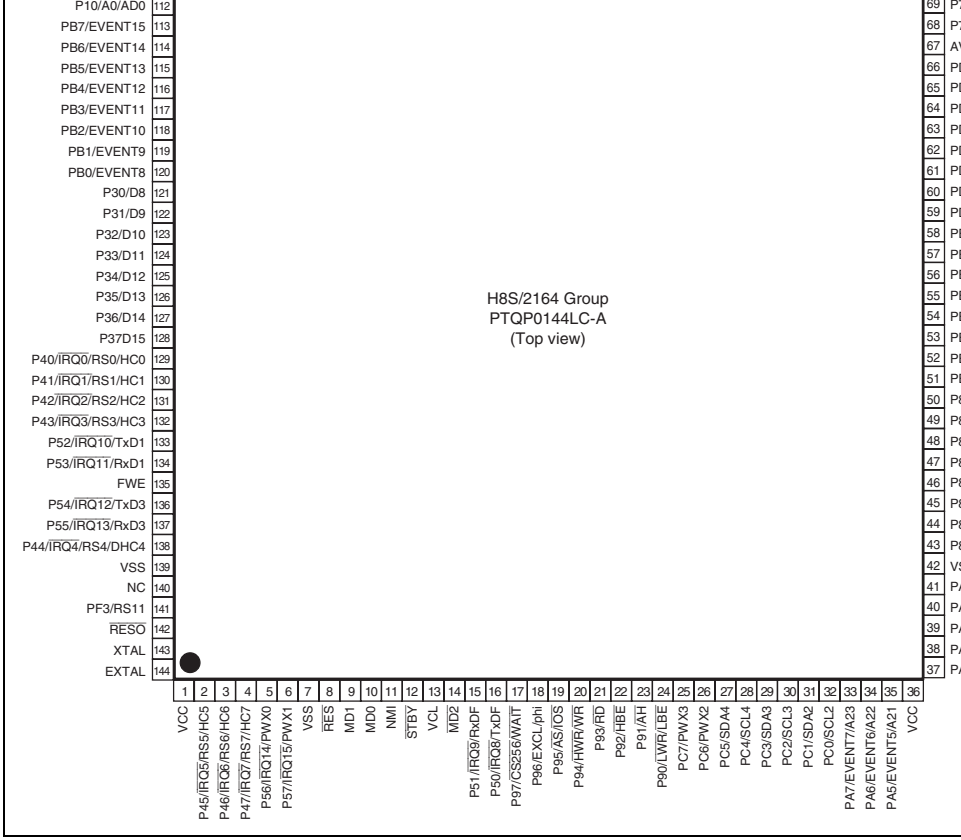


Figure 1.2 Pin Assignment

4	P47/$\overline{\text{IRQ7}}$/RS7/HC7	P47/$\overline{\text{IRQ7}}$/RS7/HC7	NC
5	P56/$\overline{\text{IRQ14}}$/PWX0	P56/$\overline{\text{IRQ14}}$/PWX0	NC
6	P57/$\overline{\text{IRQ15}}$/PWX1	P57/$\overline{\text{IRQ15}}$/PWX1	NC
7	VSS	VSS	VSS
8	$\overline{\text{RES}}$	$\overline{\text{RES}}$	$\overline{\text{RES}}$
9	MD1	MD1	VSS
10	MD0	MD0	VSS
11	NMI	NMI	FA9
12	$\overline{\text{STBY}}$	$\overline{\text{STBY}}$	VCC
13	VCL	VCL	VCL
14	$\overline{\text{MD2}}$	$\overline{\text{MD2}}$	VCC
15	P51/$\overline{\text{IRQ9}}$/RxDF	P51/$\overline{\text{IRQ9}}$/RxDF	FA17
16	P50/$\overline{\text{IRQ8}}$/TxDF	P50/$\overline{\text{IRQ8}}$/TxDF	NC
17	P97/$\overline{\text{CS256}}$/$\overline{\text{WAIT}}$	P97	VCC
18	P96/EXCL/phi	P96	NC
19	$\overline{\text{AS}}$/$\overline{\text{IOS}}$	P95	FA16
20	P94/$\overline{\text{HWR}}$/$\overline{\text{WR}}$	P94	FA15
21	P93/$\overline{\text{RD}}$	P93	$\overline{\text{WE}}$
22	P92/$\overline{\text{HBE}}$	P92	VSS
23	P91/$\overline{\text{AH}}$	P91	VCC
24	P90/$\overline{\text{LWR}}$/$\overline{\text{LBE}}$	P90	VCC
25	PC7/PWX3	PC7/PWX3	NC
26	PC6/PWX2	PC6/PWX2	NC

33	PA7/EVENT7/A23	PA7/EVENT7	NC
34	PA6/EVENT6/A22	PA6/EVENT6	NC
35	PA5/EVENT5/A21	PA5/EVENT5	NC
36	VCC	VCC	VCC
37	PA4/EVENT4/A20	PA4/EVENT4	NC
38	PA3/EVENT3/A19	PA3/EVENT3	NC
39	PA2/EVENT2/A18	PA2/EVENT2	NC
40	PA1/EVENT1/A17	PA1/EVENT1	NC
41	PA0/EVENT0/A16	PA0/EVENT0	NC
42	VSS	VSS	VSS
43	P87/$\overline{\text{ExIRQ15}}$/$\overline{\text{ADTRG}}$	P87/$\overline{\text{ExIRQ15}}$/$\overline{\text{ADTRG}}$	NC
44	P86/$\overline{\text{ExIRQ14}}$	P86/$\overline{\text{ExIRQ14}}$	NC
45	P85/$\overline{\text{ExIRQ13}}$/SCK1	P85/$\overline{\text{ExIRQ13}}$/SCK1	NC
46	P84/$\overline{\text{ExIRQ12}}$/SCK3	P84/$\overline{\text{ExIRQ12}}$/SCK3	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

62	PD4/$\overline{\text{CLKRUN}}$	PD4/$\overline{\text{CLKRUN}}$	NC
63	PD3/GA20	PD3/GA20	NC
64	PD2/$\overline{\text{PME}}$	PD2/$\overline{\text{PME}}$	NC
65	PD1/$\overline{\text{LSMI}}$	PD1/$\overline{\text{LSMI}}$	NC
66	PD0/LSCI	PD0/LSCI	NC
67	AVSS	AVSS	VSS
68	P70/$\overline{\text{ExIRQ0}}$/AN0	P70/$\overline{\text{ExIRQ0}}$/AN0	NC
69	P71/$\overline{\text{ExIRQ1}}$/AN1	P71/$\overline{\text{ExIRQ1}}$/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	P76/$\overline{\text{ExIRQ6}}$/AN6	NC
75	P77/$\overline{\text{ExIRQ7}}$/AN7	P77/$\overline{\text{ExIRQ7}}$/AN7	NC
76	AVCC	AVCC	VCC
77	AVref	AVref	VCC
78	P60/DB8/D0	P60/DB8	NC
79	P61/DB9/D1	P61/DB9	NC
80	P62/DB10/D2	P62/DB10	NC
81	P63/DB11/D3	P63/DB11	NC
82	P64/DB12/$\overline{\text{CTS}}$/D4	P64/DB12/$\overline{\text{CTS}}$	NC
83	P65/DB13/$\overline{\text{RTS}}$/D5	P65/DB13/$\overline{\text{RTS}}$	NC
84	P66/DB14/D6	P66/DB14	NC

91	$\overline{\text{ETRST}}$	$\overline{\text{ETRST}}$	$\overline{\text{RES}}$
92	PF2/RS10	PF2/RS10	NC
93	PF1/RS9	PF1/RS9	NC
94	PF0/RS8	PF0/RS8	NC
95	VSS	VSS	VSS
96	P27/A15/AD15	P27	$\overline{\text{CE}}$
97	P26/A14/AD14	P26	FA14
98	P25/A13/AD13	P25	FA13
99	P24/A12/AD12	P24	FA12
100	P23/A11/AD11	P23	FA11
101	P22/A10/AD10	P22	FA10
102	P21/A9/AD9	P21	$\overline{\text{OE}}$
103	P20/A8/AD8	P20	FA8
104	P17/A7/AD7	P17	FA7
105	P16/A6/AD6	P16	FA6
106	P15/A5/AD5	P15	FA5
107	P14/A4/AD4	P14	FA4
108	P13/A3/AD3	P13	FA3
109	P12/A2/AD2	P12	FA2
110	P11/A1/AD1	P11	FA1
111	VSS	VSS	VSS
112	P10/A0/AD0	P10	FA0
113	PB7/EVENT15	PB7/EVENT15	NC

120	PB0/EVENT8	PB0/EVENT8	NC
121	P30/D8	P30	FO0
122	P31/D9	P31	FO1
123	P32/D10	P32	FO2
124	P33/D11	P33	FO3
125	P34/D12	P34	FO4
126	P35/D13	P35	FO5
127	P36/D14	P36	FO6
128	P37/D15	P37	FO7
129	P40/$\overline{\text{IRQ0}}$/RS0/HC0	P40/$\overline{\text{IRQ0}}$/RS0/HC0	NC
130	P41/$\overline{\text{IRQ1}}$/RS1/HC1	P41/$\overline{\text{IRQ1}}$/RS1/HC1	NC
131	P42/$\overline{\text{IRQ2}}$/RS2/HC2	P42/$\overline{\text{IRQ2}}$/RS2/HC2	NC
132	P43/$\overline{\text{IRQ3}}$/RS3/HC3	P43/$\overline{\text{IRQ3}}$/RS3/HC3	NC
133	P52/$\overline{\text{IRQ10}}$/TxD1	P52/$\overline{\text{IRQ10}}$/TxD1	FA18
134	P53/$\overline{\text{IRQ11}}$/RxD1	P53/$\overline{\text{IRQ11}}$/RxD1	FA19
135	FWE	FWE	FWE
136	P54/$\overline{\text{IRQ12}}$/TxD3	P54/$\overline{\text{IRQ12}}$/TxD3	NC
137	P55/$\overline{\text{IRQ13}}$/RxD3	P55/$\overline{\text{IRQ13}}$/RxD3	NC
138	P44/$\overline{\text{IRQ4}}$/RS4/HC4	P44/$\overline{\text{IRQ4}}$/RS4/HC4	NC
139	VSS	VSS	VSS
140	NC	NC	NC
141	PF3/RS11	PF3/RS11	NC
142	$\overline{\text{RESO}}$	$\overline{\text{RESO}}$	NC

				located near this pin) to stabilize in step-down power.
	VSS	7, 42, 95, 111, 139	Input	Ground pins. Connect all these pin system power supply (0V).
Clock	XTAL	143	Input	For connection to a crystal resona external clock can be supplied fro EXTAL pin. For an example of cry resonator connection, see section Clock Pulse Generator.
	EXTAL	144	Input	
	ϕ	18	Output	Supplies the system clock to exter devices.
	EXCL	18	Input	32.768-kHz external clock for sub should be supplied.
Operating mode control	$\overline{\text{MD2}}$ MD1 MD0	14 9 10	Input	These pins set the operating mode at these pins should not be chang operation.
System control	$\overline{\text{RES}}$	8	Input	Reset pin. When this pin is low, th reset.
	$\overline{\text{RESO}}$	142	Output	Outputs a reset signal to an extern device.
	$\overline{\text{STBY}}$	12	Input	When this pin is low, a transition is hardware standby mode.
	FWE	135	Input	Pin for use by flash memory.

Interrupts	NMI	11	Input	Nonmaskable interrupt request in
	$\overline{\text{IRQ15}}$, $\overline{\text{IRQ14}}$, $\overline{\text{IRQ13}}$, $\overline{\text{IRQ12}}$, $\overline{\text{IRQ11}}$, $\overline{\text{IRQ10}}$, $\overline{\text{IRQ9}}$, $\overline{\text{IRQ8}}$, $\overline{\text{IRQ7}}$ to $\overline{\text{IRQ5}}$, $\overline{\text{IRQ4}}$, $\overline{\text{IRQ3}}$ to $\overline{\text{IRQ0}}$	6, 5, 137, 136, 134, 133, 15, 16 4 to 2 138 132 to 129	Input	These pins are used to request n interrupts. Either $\overline{\text{IRQn}}$ or $\overline{\text{ExIRQn}}$ can be se the IRQn interrupt signal input pi
	$\overline{\text{ExIRQ15}}$ to $\overline{\text{ExIRQ12}}$, $\overline{\text{ExIRQ11}}$ to $\overline{\text{ExIRQ8}}$, $\overline{\text{ExIRQ7}}$ to $\overline{\text{ExIRQ5}}$, $\overline{\text{ExIRQ4}}$ to $\overline{\text{ExIRQ0}}$	43 to 50, 75 to 68, 6, 5		

	$\overline{LWR}$	24	Output	Low level on this pin indicates that MCU is writing to an external addr space. The lower byte of the data bus is v
	$\overline{AS}/\overline{IOS}$	19	Output	Low level on this pin indicates that address output on the address bus
	$\overline{CS256}$	17	Output	Indicates access to the 256-Kbyte H'F80000 to H'FBFFFF.
	$\overline{WR}$	20	Output	Low level on this pin indicates that MCU is writing to an external addr space.
	$\overline{HBE}$	22	Output	Low level on this pin indicates that MCU is accessing an external add space. The upper byte of the data bus is v
	$\overline{LBE}$	24	Output	Low level on this pin indicates that MCU is accessing an external add space. The lower byte of the data bus is v
	$\overline{AH}$	23	Output	Address latch signal for the addres multiplex bus
Boundary scan	$\overline{ETRST}$	91	Input	Boundary scan interface pins
	ETMS	87	Input	
	ETDO	88	Output	
	ETDI	89	Input	
	ETCK	90	Input	

cation Interface with FIFO (SCIF)	RxDF	15	Input	Receive data input pin
	$\overline{CTS}$	82	Input	Transmit grant input pin
	$\overline{RTS}$	83	Output	Transmit request output pin
I^2C bus interface (IIC)	SCL0, SCL1, SCL2, SCL3, SCL4, SCL5	50, 48 32, 30 28, 60	Input/ Output	IIC clock input/output pins. These drive a bus directly with the NMC drain output.
	SDA0, SDA1, SDA2, SDA3, SDA4, SDA5	49, 47 31, 29 27, 59	Input/ Output	IIC data input/output pins. These drive a bus directly with the NMC drain output.
A/D converter	AN7 to AN0	75 to 68	Input	Analog input pins
	AVCC	76	Input	Analog power supply pins. When converter is not used, these pins connected to the system power s (+3.3 V).
	AVref	77	Input	Analog reference voltage input pi the A/D converter is not used, thi should be connected to the syste supply (+3.3 V).
	AVSS	67	Input	Analog ground pin. This pin shou connected to the system power s V).
	$\overline{ADTRG}$	43	Input	External trigger input pin to start conversion

	$\overline{\text{LSCI}}$, $\overline{\text{LSMI}}$, $\overline{\text{PME}}$	63, 65, 64	Input/ Output	LPC auxiliary output. Their functio general I/O port.
	GA20	63	Input/ Output	GATE A20 control signal output pi used as the input pin for monitorin output state.
	$\overline{\text{CLKRUN}}$	62	Input/ Output	Input/output pin used to request st LCLK operation while LCLK is sto
	$\overline{\text{LPCPD}}$	61	Input	Input pin used to control shutdown LCP module
Event Counter	EVENT15 to EVENT8, EVENT7 to EVENT5, EVENT4 to EVENT0	113 to 120, 33 to 35, 37 to 41	Input	Event counter input pins
Retain state output pins	RS11, RS10 to RS8, RS7 to RS5, RS4, RS3 to RS0	141, 92 to 94, 4 to 2, 138, 132 to 129	Output	The outputs on these pins are only initialized by a system reset.
Debounced input pins	DB15 to DB8	85 to 78	Input	Pins with noise eliminating functio
Large current output pins	HC7 to HC5, HC4, HC3 to HC0	4 to 2, 138, 132 to 129	Output	These pins can be used to drive L for other purposes where large cu required.

P55, P54 P53, P52 P51, P50	137, 136, 134, 133, 15, 16	Output	
P67 to P60	85 to 78	Input/ Output	8-bit input/output pins
P77 to P70	75 to 68	Input	8-bit input pins
P87 to P80	43 to 50	Input/ Output	8-bit input/output pins
P97 to P90	17 to 24	Input/ Output	8-bit input/output pins
PA7 to PA5, PA4 to PA0	33 to 35 37 to 41	Input/ Output	8-bit input/output pins
PB7 to PB0	113 to 120	Input/ Output	8-bit input/output pins
PC7 to PC0	25 to 32	Input/ Output	8-bit input/output pins
PD7 to PD0	59 to 66	Input/ Output	8-bit input/output pins
PE7 to PE0	51 to 58	Input/ Output	8-bit input/output pins
PF3, PF2 to PF0	141, 92 to 94	Input/ Output	4-bit input/output pins

- Upward-compatible with H8/300 and H8/300H CPUs
  - Can execute H8/300 and H8/300H CPUs object programs
- General-register architecture
  - Sixteen 16-bit general registers also usable as sixteen 8-bit registers or eight 32-b
- Sixty-nine basic instructions
  - 8/16/32-bit arithmetic and logic instructions
  - Multiply and divide instructions
  - Powerful bit-manipulation instructions
  - Multiply-and-accumulate instruction
- Eight addressing modes
  - Register direct [Rn]
  - Register indirect [@ERn]
  - Register indirect with displacement [@(d:16,ERn) or @(d:32,ERn)]
  - Register indirect with post-increment or pre-decrement [@ERn+ or @–ERn]
  - Absolute address [@aa:8, @aa:16, @aa:24, or @aa:32]
  - Immediate [#xx:8, #xx:16, or #xx:32]
  - Program-counter relative [@(d:8,PC) or @(d:16,PC)]
  - Memory indirect [@@aa:8]
- 16-Mbyte address space
  - Program: 16 Mbytes
  - Data: 16 Mbytes
- High-speed operation
  - All frequently-used instructions execute in one or two states
  - 8/16/32-bit register-register add/subtract: 1 state
  - 8 × 8-bit register-register multiply: 2 states

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

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

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

- Register configuration

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

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

		Execution States	
Instruction	**Mnemonic**	**H8S/2600**	**H8S/2000**
MULXU	MULXU.B Rs, Rd	2*	12
	MULXU.W Rs, ERd	2*	20
MULXS	MULXS.B Rs, Rd	3*	13
	MULXS.W Rs, ERd	3*	21
CLRMAC	CLRMAC	1*	Not supported
LDMAC	LDMAC ERs,MACH	1*	
	LDMAC ERs,MACL	1*	
STMAC	STMAC MACH,ERd	1*	
	STMAC MACl,ERd	1*	

Note: * This becomes one state greater immediately after a MAC instruction.
In addition, there are differences in address space, CCR and EXR register fun and power-down modes, etc., depending on the model.

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

2.1.3 Differences from H8/300H CPU

In comparison to the H8/300H CPU, the H8S/2600 CPU has the following enhancement

- More control registers
  - One 8-bit and two 32-bit control registers have been added.
- Enhanced instructions
  - Addressing modes of bit-manipulation instructions have been enhanced.
  - A multiply-and-accumulate instruction has been added.
  - Two-bit shift instructions have been added.
  - Instructions for saving and restoring multiple registers have been added.
  - A test and set instruction has been added.
- Higher speed
  - Basic instructions execute twice as fast.

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

- Extended Registers (En)

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

- Instruction Set

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

- Exception Vector Table and Memory Indirect Branch Addresses

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

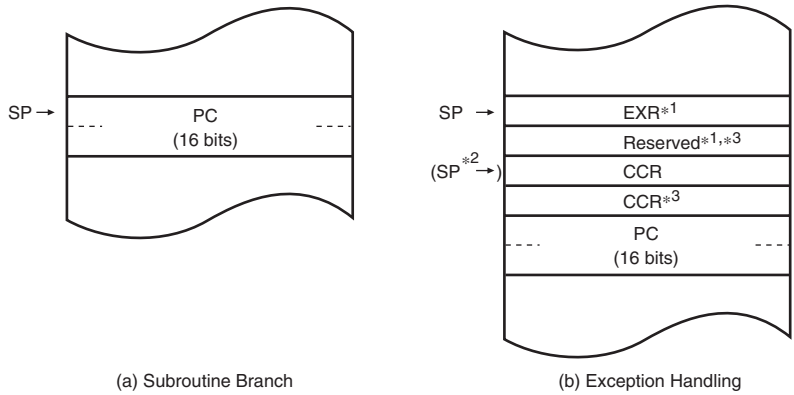
  The memory indirect addressing mode (@@aa:8) employed in the JMP and JSR instr
  uses an 8-bit absolute address included in the instruction code to specify a memory op
  that contains a branch address. In normal mode the operand is a 16-bit word operand,
  providing a 16-bit branch address. Branch addresses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

- Stack Structure

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

Note: Normal mode is not available in this LSI.

Figure 2.1 Exception Vector Table (Normal Mode)



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

Figure 2.2 Stack Structure in Normal Mode

Exception Vector Table and Memory Indirect Branch Addresses

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

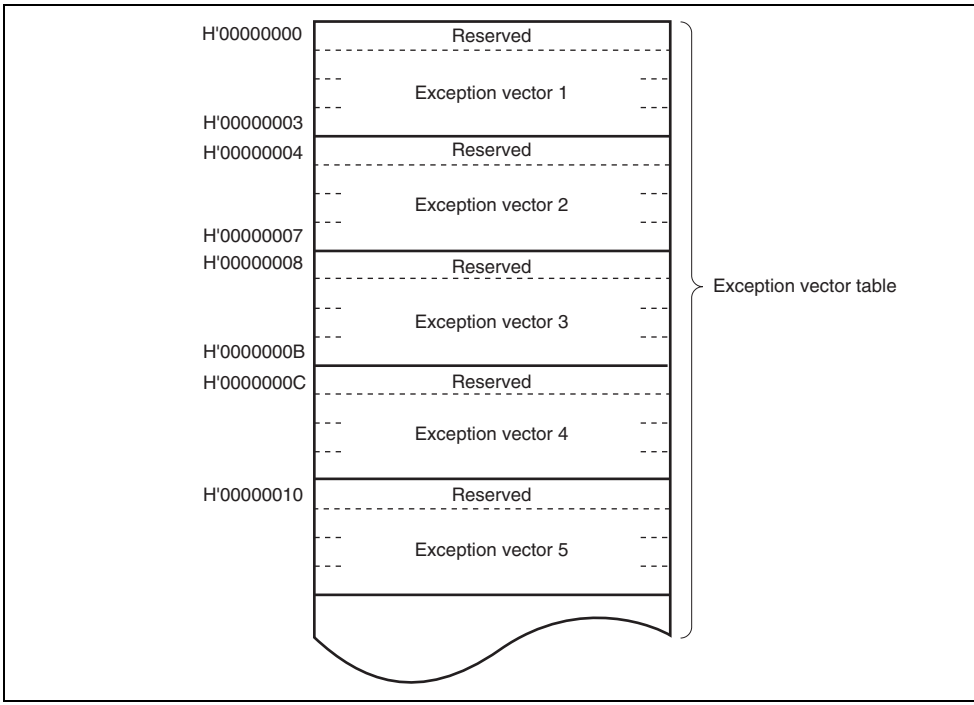


Figure 2.3 Exception Vector Table (Advanced Mode)

Exception Handling.

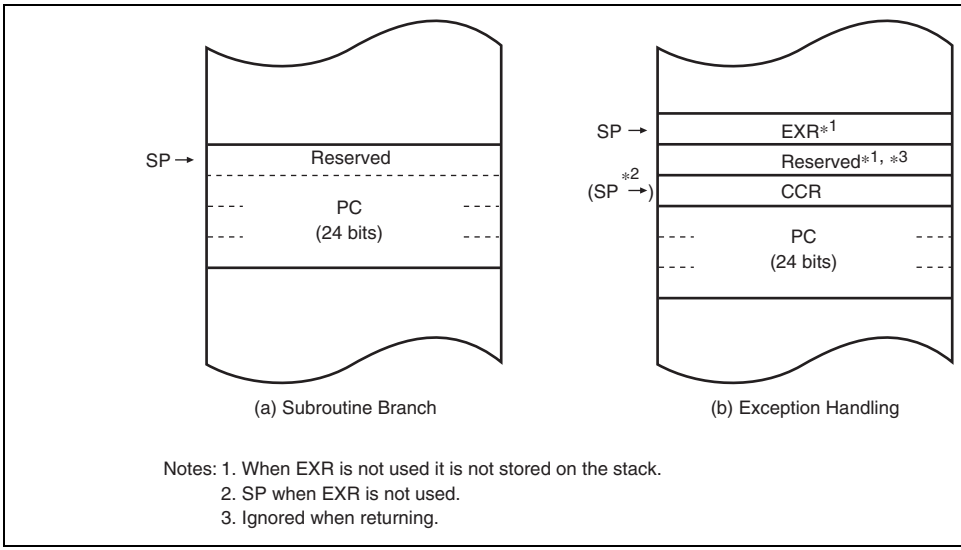


Figure 2.4 Stack Structure in Advanced Mode

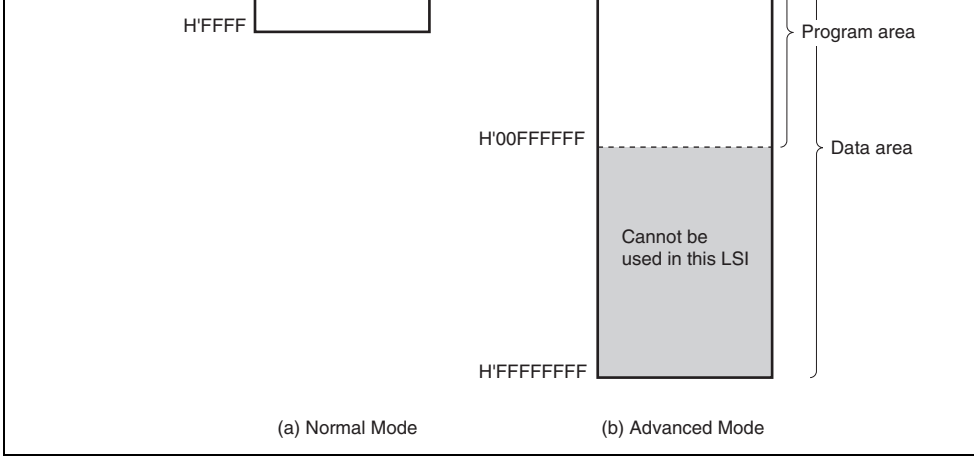


Figure 2.5 Memory Map

ER2	E2	R2H	R2L
ER3	E3	R3H	R3L
ER4	E4	R4H	R4L
ER5	E5	R5H	R5L
ER6	E6	R6H	R6L
ER7 (SP)	E7	R7H	R7L

Control Registers (CR)

[Legend]

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

Figure 2.6 CPU Registers

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

The usage of each register can be selected independently.

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

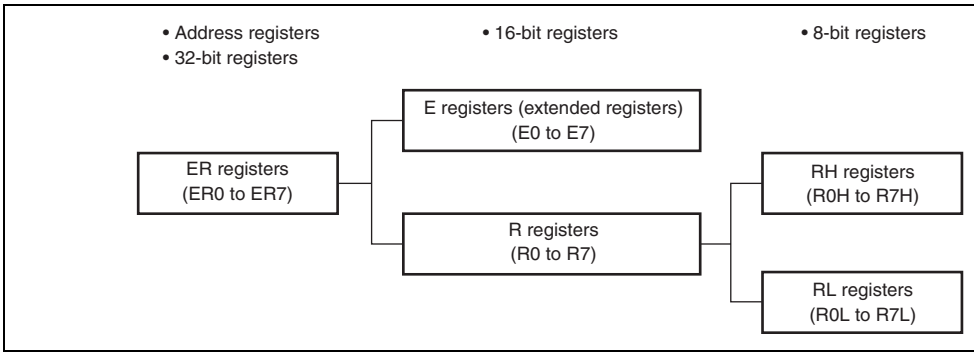


Figure 2.7 Usage of General Registers

Figure 2.8 Stack

2.4.2 Program Counter (PC)

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

2.4.3 Extended Control Register (EXR)

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

Bit	Bit Name	Initial Value	R/W	Description
7	T	0	R/W	Trace Bit When this bit is set to 1, a trace exception is g each time an instruction is executed. When th cleared to 0, instructions are executed in sequ
6 to 3	—	All 1	—	Reserved These bits are always read as 1.
2	I2	1	R/W	These bits designate the interrupt mask level For details, refer to section 5, Interrupt Contro
1	I1	1	R/W	
0	I0	1	R/W	

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

- Shift and rotate instructions, to indicate a

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

2.4.5 Multiply-Accumulate Register (MAC)

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

2.4.6 Initial Values of CPU Registers

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

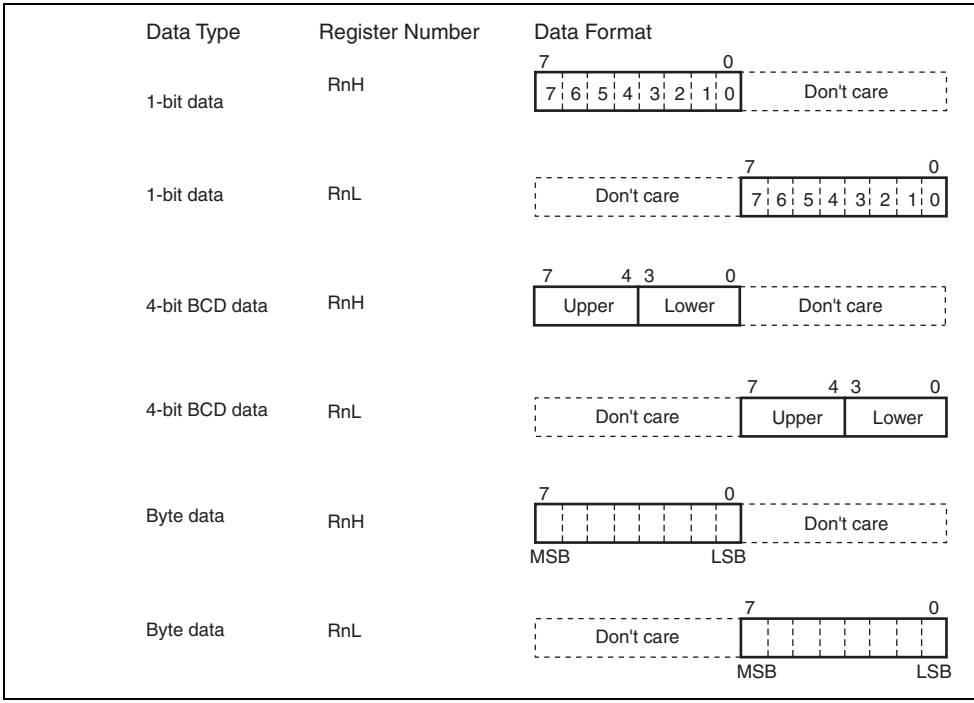
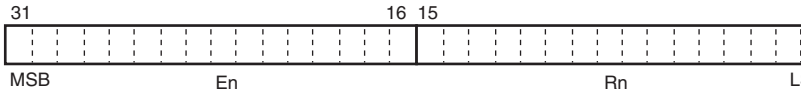


Figure 2.9 General Register Data Formats (1)

Longword data ERn



[Legend]

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

Figure 2.9 General Register Data Formats (2)

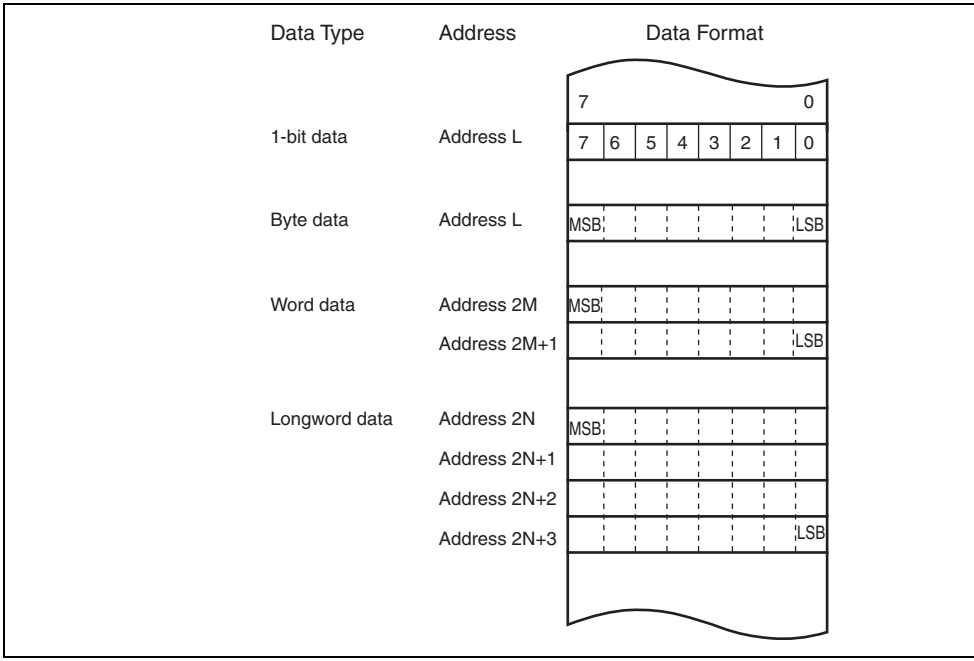


Figure 2.10 Memory Data Formats

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

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

1. POP.W Rn and PUSH.W Rn are identical to MOV.W @SP+,Rn and MOV.W POP.L ERn and PUSH.L ERn are identical to MOV.L @SP+,ERn and MOV.L ERn,@-SP.
2. Bcc is the general name for conditional branch instructions.
3. Cannot be used in this LSI.
4. Only register ER0, ER1, ER4, or ER5 should be used when using the TAS ins

ERn	General register (32-bit register)
MAC	Multiply-accumulate register (32-bit register)
(EAd)	Destination operand
(EAs)	Source operand
EXR	Extended control register
CCR	Condition-code register
N	N (negative) flag in CCR
Z	Z (zero) flag in CCR
V	V (overflow) flag in CCR
C	C (carry) flag in CCR
PC	Program counter
SP	Stack pointer
#IMM	Immediate data
disp	Displacement
+	Addition
−	Subtraction
×	Multiplication
÷	Division
∧	Logical AND
∨	Logical OR
⊕	Logical XOR
→	Move
~	NOT (logical complement)

MOVFPE	B	Cannot be used in this LSI.
MOVTPE	B	Cannot be used in this LSI.
POP	W/L	@SP+ → Rn Pops a general register from the stack. POP.W Rn is identical MOV.W @SP+, Rn. POP.L ERn is identical to MOV.L @SP+,
PUSH	W/L	Rn → @–SP Pushes a general register onto the stack. PUSH.W Rn is ident MOV.W Rn, @–SP. PUSH.L ERn is identical to MOV.L ERn,
LDM	L	@SP+ → Rn (register list) Pops two or more general registers from the stack.
STM	L	Rn (register list) → @–SP Pushes two or more general registers onto the stack.

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

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

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

		Takes the two's complement (arithmetic complement) of data i general register.
EXTU	W/L	Rd (zero extension) → Rd Extends the lower 8 bits of a 16-bit register to word size, or the bits of a 32-bit register to longword size, by padding with zeros left.
EXTS	W/L	Rd (sign extension) → Rd Extends the lower 8 bits of a 16-bit register to word size, or the bits of a 32-bit register to longword size, by extending the sign
TAS*[2]	B	@ERd – 0, 1 → (<bit 7> of @ERd) Tests memory contents, and sets the most significant bit (bit 7
MAC	—	(EAs) × (EAd) + MAC → MAC Performs signed multiplication on memory contents and adds to the multiply-accumulate register. The following operations c performed: 16 bits × 16 bits + 32 bits → 32 bits, saturating 16 bits × 16 bits + 42 bits → 42 bits, non-saturating
CLRMAC	—	0 → MAC Clears the multiply-accumulate register to zero.
LDMAC STMAC	L	Rs → MAC, MAC → Rd Transfers data between a general register and a multiply-accu register.

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

NOT	B/W/L	~(Rd) → (Rd) Takes the one's complement (logical complement) of general re contents.

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

Table 2.6 Shift Instructions

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

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

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

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

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

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

		BCC(BHS)	Carry clear (high or same)	C = 0
		BCS(BLO)	Carry set (low)	C = 1
		BNE	Not equal	Z = 0
		BEQ	Equal	Z = 1
		BVC	Overflow clear	V = 0
		BVS	Overflow set	V = 1
		BPL	Plus	N = 0
		BMI	Minus	N = 1
		BGE	Greater or equal	N ⊕ V = 0
		BLT	Less than	N ⊕ V = 1
		BGT	Greater than	Z∨(N ⊕ V) = 0
		BLE	Less or equal	Z∨(N ⊕ V) = 1
JMP	—	Branches unconditionally to a specified address.		
BSR	—	Branches to a subroutine at a specified address.		
JSR	—	Branches to a subroutine at a specified address.		
RTS	—	Returns from a subroutine		

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

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

```
else next;
```

Transfers a data block. Starting from the address set in ER5, t
data for the number of bytes set in R4L or R4 to the address lo
in ER6.

Execution of the next instruction begins as soon as the transfe
completed.

Some instructions have two operation fields.

- Register Field

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

- Effective Address Extension

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

- Condition Field

  Specifies the branching condition of Bcc instructions.

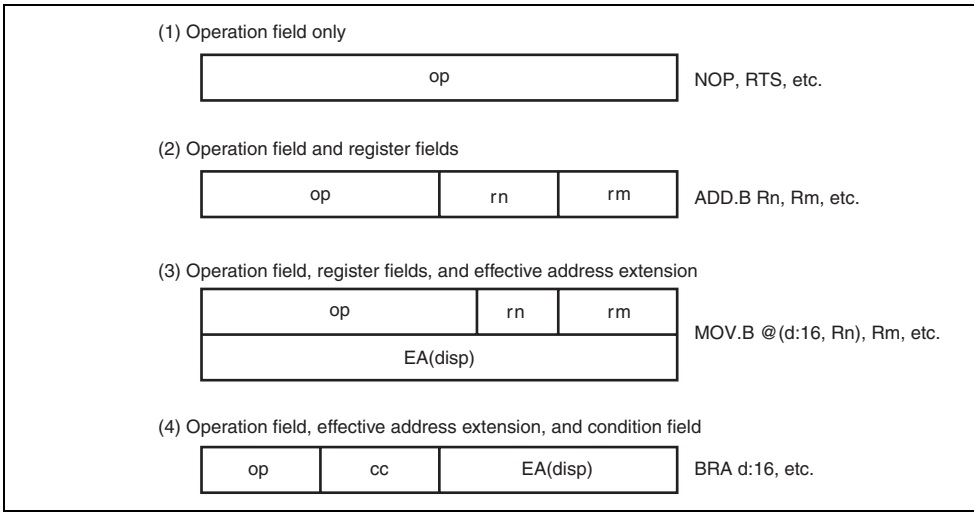


Figure 2.11 Instruction Formats (Examples)

No.	Addressing Mode	Symbol
1	Register direct	Rn
2	Register indirect	@ERn
3	Register indirect with displacement	@(d:16,ERn)/@(d:32,ERn)
4	Register indirect with post-increment Register indirect with pre-decrement	@ERn+ @–ERn
5	Absolute address	@aa:8/@aa:16/@aa:24/@a
6	Immediate	#xx:8/#xx:16/#xx:32
7	Program-counter relative	@(d:8,PC)/@(d:16,PC)
8	Memory indirect	@@aa:8

2.7.1 Register Direct—Rn

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

2.7.2 Register Indirect—@ERn

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

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

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

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

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

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

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

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

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

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

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

This mode is used in the Bcc and BSR instructions. An 8-bit or 16-bit displacement cont
the instruction is sign-extended and added to the 24-bit PC contents to generate a branch
Only the lower 24 bits of this branch address are valid; the upper 8 bits are all assumed t
(H′00). The PC value to which the displacement is added is the address of the first byte
instruction, so the possible branching range is –126 to +128 bytes (–63 to +64 words) or
+32768 bytes (–16383 to +16384 words) from the branch instruction. The resulting valu
be an even number.

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

Note: Normal mode is not available in this LSI.

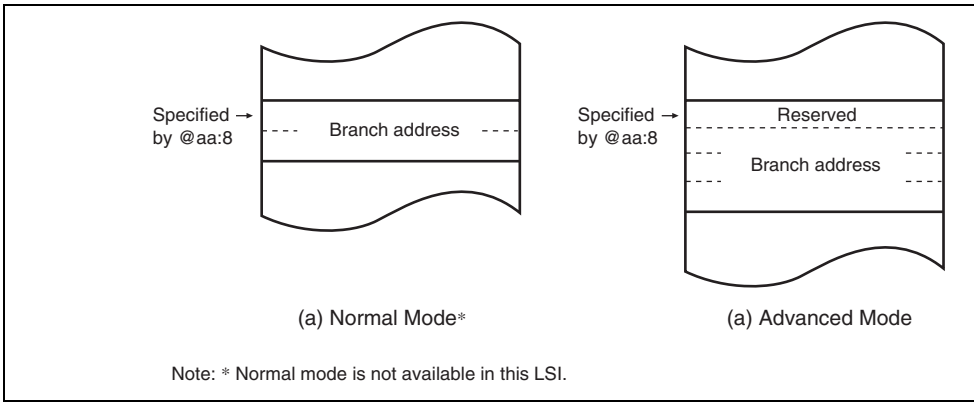
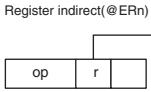
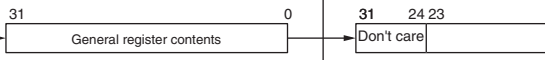
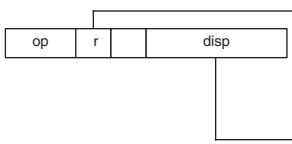
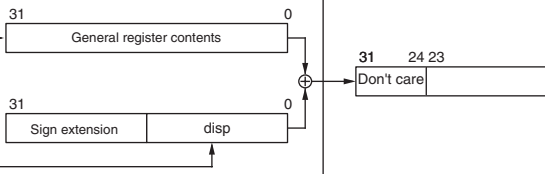
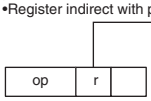
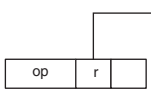
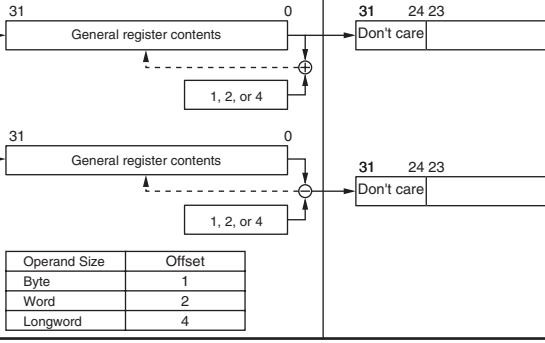
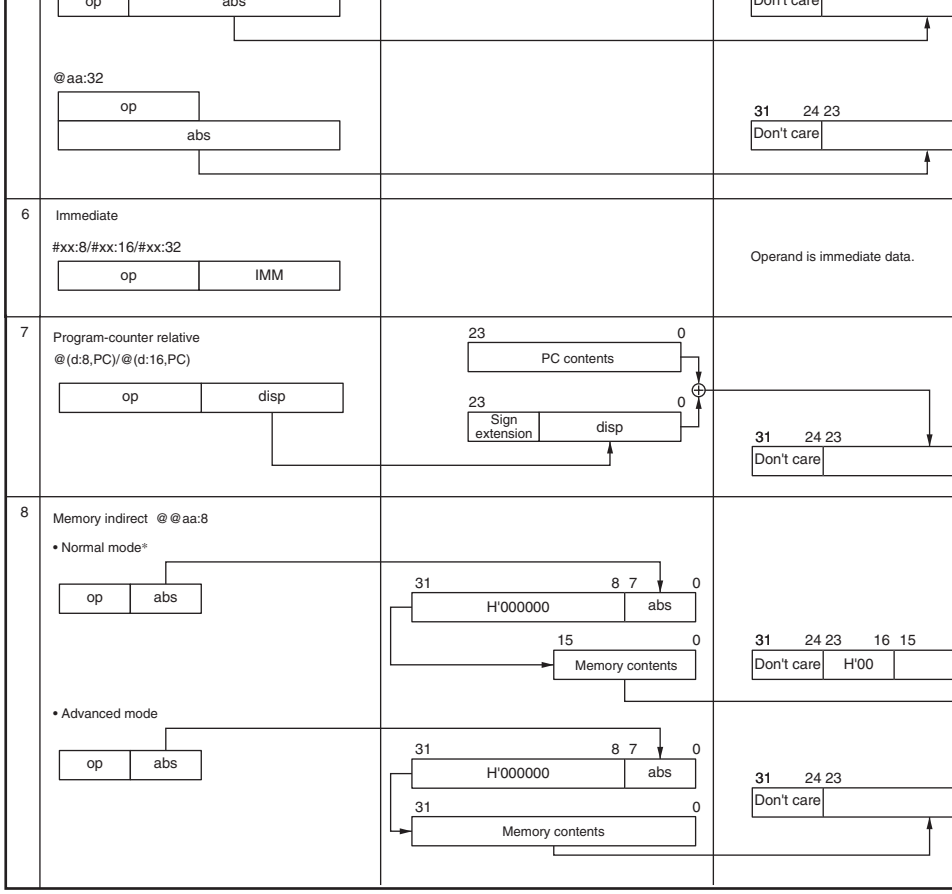


Figure 2.12 Branch Address Specification in Memory Indirect Mode

2	Register indirect(@ERn) 	
3	Register indirect with displacement @(d:16,ERn) or @(d:32,ERn) 	
4	Register indirect with post-increment or pre-decrement •Register indirect with post-increment @ERn+  •Register indirect with pre-decrement @-ERn 	 Operand Size / Offset Byte: 1 Word: 2 Longword: 4



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

- Exception-Handling State

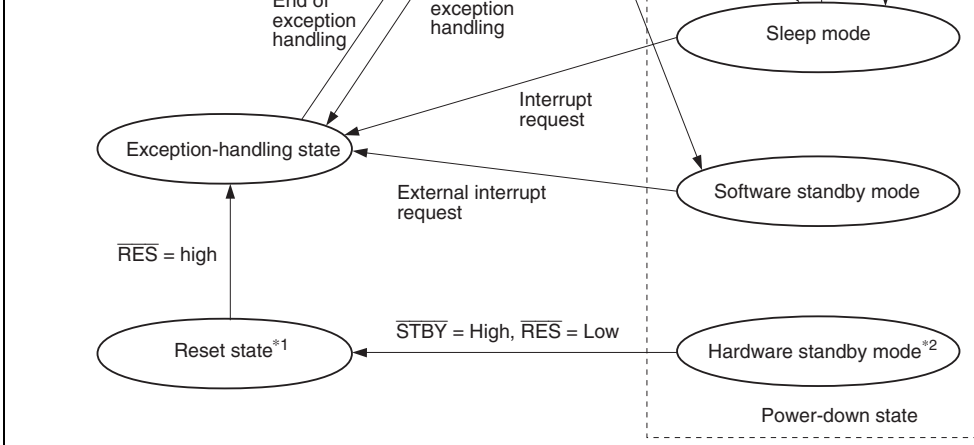
  The exception-handling state is a transient state that occurs when the CPU alters the processing flow due to an exception source, such as a reset, trace, interrupt, or trap i The CPU fetches a start address (vector) from the exception vector table and branche address. For further details, refer to section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

- Program Execution State

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

- Program Stop State

  This is a power-down state in which the CPU stops operating. The program stop stat when a SLEEP instruction is executed or the CPU enters software standby mode. Fo details, refer to section 24, 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 $\overline{STBY}$ goes low.

Figure 2.13 State Transitions

beforehand.

MCU Operating Mode	$\overline{MD2}$	MD1	MD0	CPU Operating Mode	Description
2	1	1	0	Advanced	Extended mode with on-chi Single-chip mode
4	0	0	0	—	Flash programming/erasing
6	0	1	0	Emulation	On-chip emulation mode

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

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

3.2.1 Mode Control Register (MDCR)

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

Bit	Bit Name	Initial Value	R/W	Description
7	EXPE	0	R/W	Extended Mode Enable Specifies extended mode. 0: Single-chip mode 1: Extended mode
6 to 3	—	All 0	R	Reserved
2	MDS2	—*	R	Mode Select 2, 1, and 0
1	MDS1	—*	R	These bits indicate the input levels at mode pins
0	MDS0	—*	R	MD1, and MD0) (the current operating mode). Bi MDS1, and MDS0 correspond to $\overline{MD2}$, MD1, and respectively. MDS2 and MDS1 are read-only bits cannot be written to. The mode pin ($\overline{MD2}$, MD1, a input levels are latched into these bits when MDC read. These latches are canceled by a reset.

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

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

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

3.2.3 Serial Timer Control Register (STCR)

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

Bit	Bit Name	Initial Value	R/W	Description
7	IICX2	0	R/W	IIC Transfer Rate Select 2, 1, and 0
6	IICX1	0	R/W	These bits control the IIC operation. These bits s transfer rate in master mode together with bits CI CKS0 in the I^2C bus mode register (ICMR). For d the transfer rate, see table 17.3. The IICXn bit co IIC_n. (n = 0 to 2)
5	IICX0	0	R/W	

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

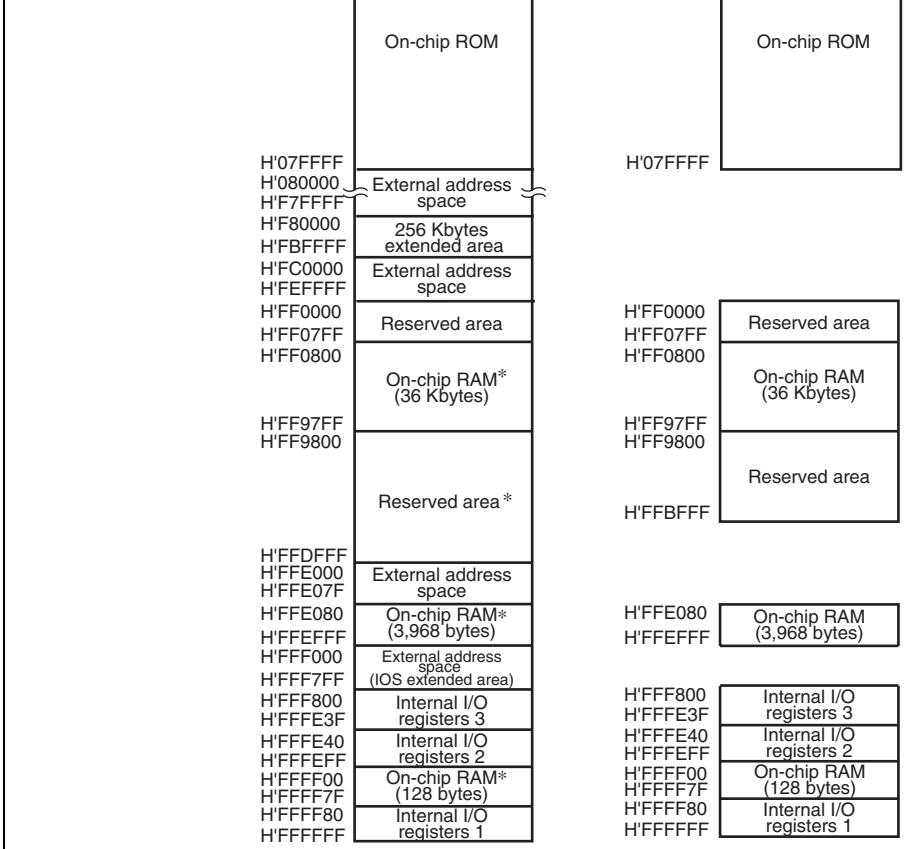
In extended modes, ports 1 and 2 function as input ports after a reset.

Ports 1 and 2 function as an address bus by setting 1 to the corresponding port data di
register (DDR). Port 3 functions as a data bus port, and parts of port 9 carry bus contr
signals. 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/output regardless of the setting of the data direction register (DDR). Port 1 can b a general port.

  When 16-bit bus is specified, ports 1 and 2 function as the port for address output and input/output regardless of the setting of the data direction register (DDR).



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

Figure 3.1 Address Map

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

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

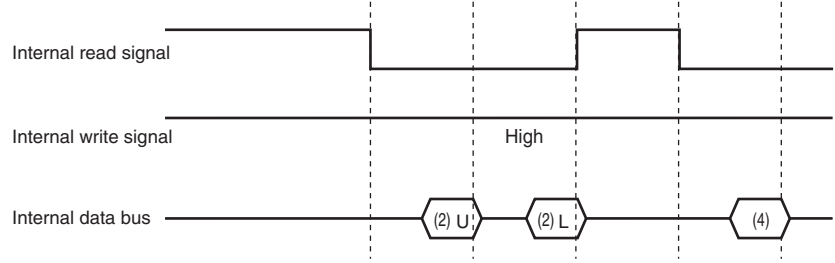
	IRQ10	58	H'0000E8 to H'0000EB
	IRQ11	59	H'0000EC to H'0000EF
	IRQ12	60	H'0000F0 to H'0000F3
	IRQ13	61	H'0000F4 to H'0000F7
	IRQ14	62	H'0000F8 to H'0000FB
	IRQ15	63	H'0000FC to H'0000FF
Internal interrupt*		64 ⎮ 107	H'000100 to H'000103 ⎮ H'0001AC to H'0001AF

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

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

1. The internal state of the CPU and the registers of the on-chip peripheral modules are i
   and the I bit in CCR is set to 1.
2. The reset exception handling vector address is read and transferred to the PC, and pro
   execution starts from the address indicated by the PC.

Figure 4.1 shows an example of the reset sequence.



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

Figure 4.1 Reset Sequence

4.3.2 Interrupts after Reset

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

4.3.3 On-Chip Peripheral Modules after Reset is Cancelled

After a reset is cancelled, the module stop control registers (MSTPCR, MSTPCRA, 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.

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

4.5 Trap Instruction Exception Handling

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

Trap instruction exception handling is conducted as follows:

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

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

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

Table 4.3 Status of CCR after Trap Instruction Exception Handling

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

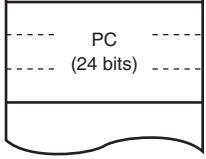


Figure 4.2 Stack Status after Exception Handling

Use the following instructions to restore registers:

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

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

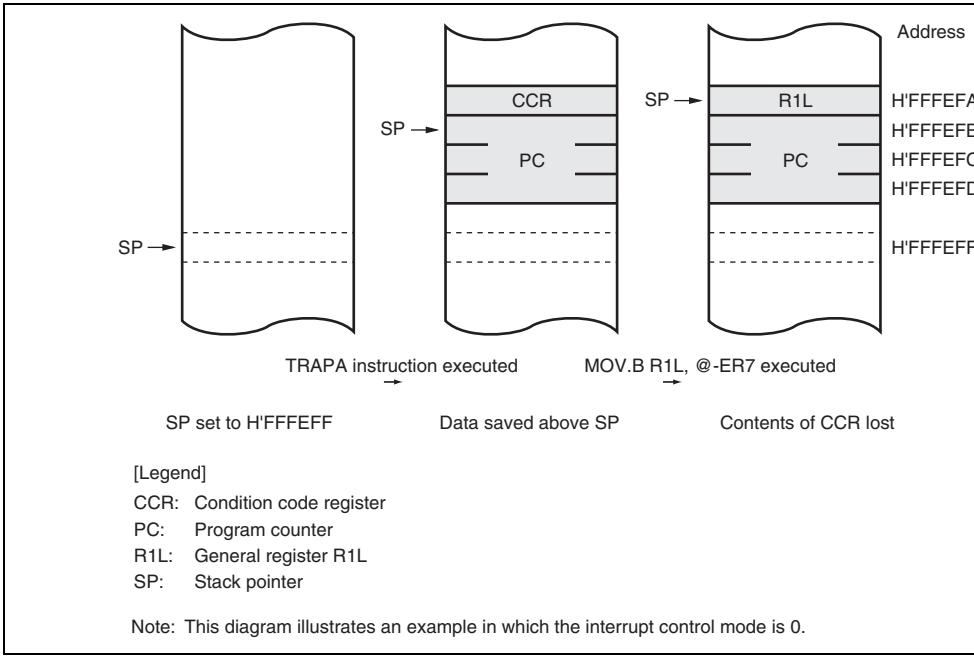


Figure 4.3 Operation When SP Value is Odd

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

- Three-level interrupt mask control

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

- Independent vector addresses

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

- Thirty-three external interrupts

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

- DTC control

  The DTC can be activated by an interrupt request.

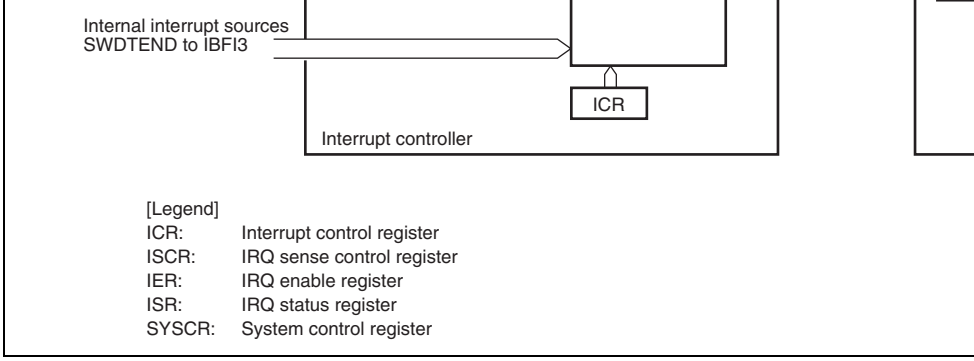


Figure 5.1 Block Diagram of Interrupt Controller

5.2 Input/Output Pins

Table 5.1 summarizes the pins of the interrupt controller.

Table 5.1 Pin Configuration

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

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

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

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

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

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

[Legend]
n: A to D

1	ICRn1	WDT_0	TMR_Y	LPC	SCIF
0	ICRn0	WDT_1	IIC_4, IIC_5	—	—

[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 1 address break is requested.

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

- BARB

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

- BARC

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

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

Note: * IRQn stands for IRQ15 to IRQ12.

- ISCR16L

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

Note: * IRQn stands for IRQ11 to IRQ8.

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

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

- ISCRL

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

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

- IER

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

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

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

- ISR

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

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

edge on the NMI pin.

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

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

The detection of IRQ15 to IRQ0 interrupts does not depend on whether the relevant pin h
set for input or output. However, when a pin is used as an external interrupt input pin, cle
corresponding port DDR to 0 so that it is not used as an I/O pin for another function.

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

Figure 5.2 Block Diagram of Interrupts IRQ15 to IRQ0

5.4.2 Internal Interrupts

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

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

Table 5.3 Interrupt Sources, Vector Addresses, and Interrupt Priorities

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

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

Note: Vector numbers not listed above are reserved by the system.

0	0	0	ICR	I	Interrupt mask control is perfor 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 5.3 shows a block diagram of the priority decision circuit.

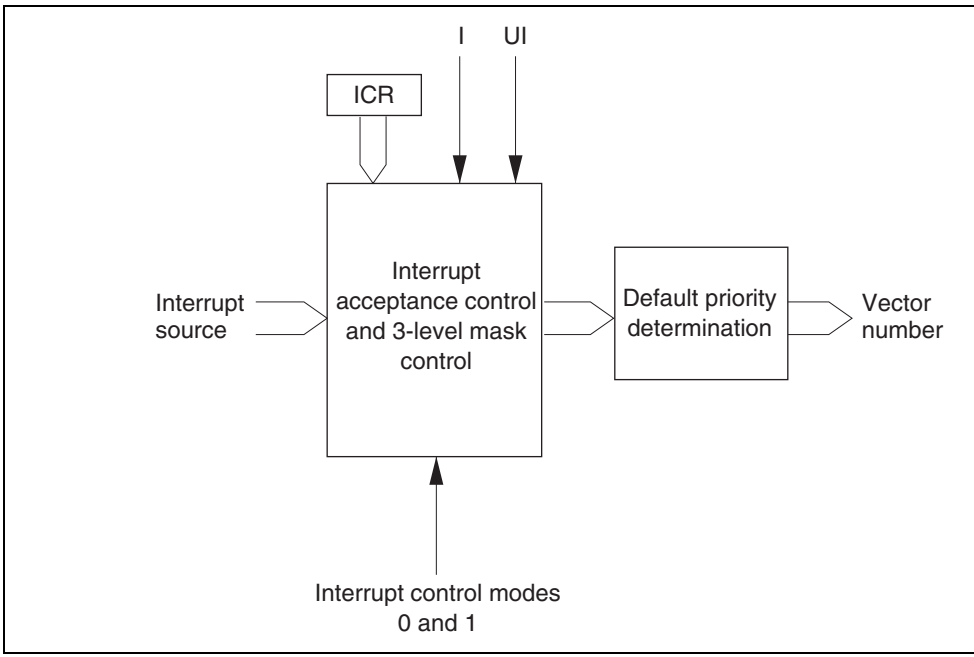


Figure 5.3 Block Diagram of Interrupt Control Operation

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

[Legend]

X: 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.

PR: Sets priority
—: Not used

5.6.1 Interrupt Control Mode 0

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

1. If an interrupt source occurs when the corresponding interrupt enable bit is set to 1, 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.

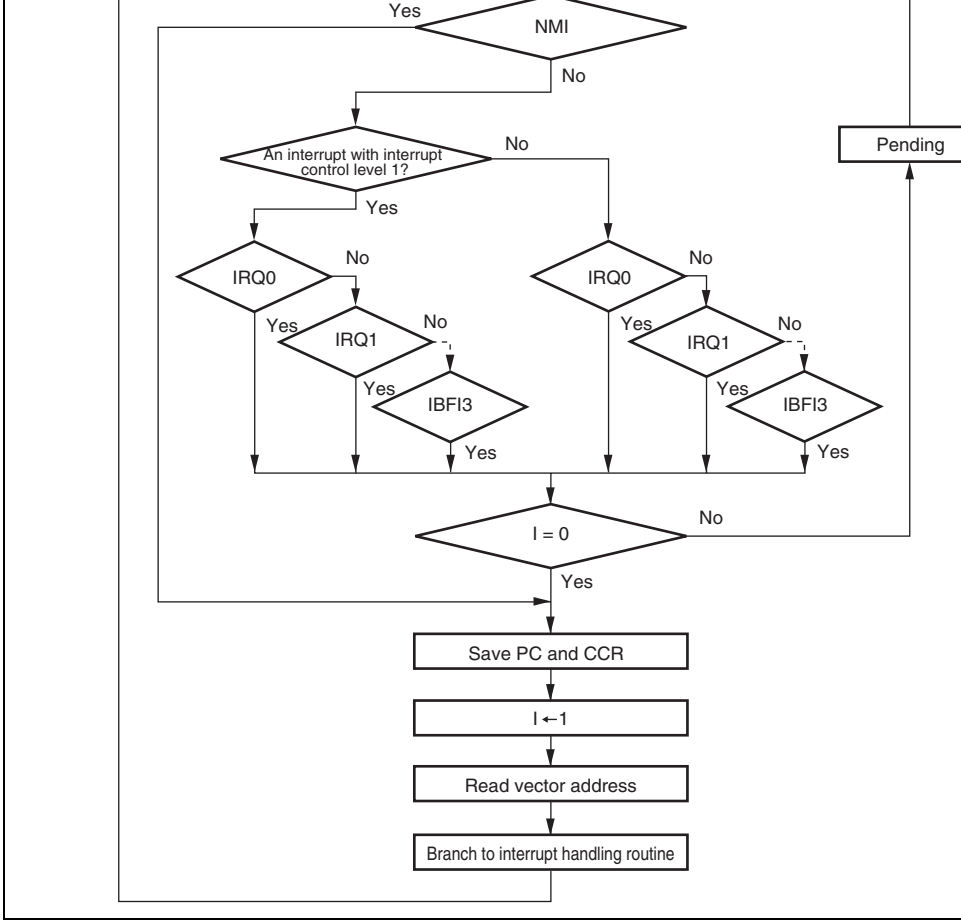


Figure 5.4 Flowchart of Procedure up to Interrupt Acceptance in Interrupt Contr

For instance, the state when the interrupt enable bit corresponding to each interrupt is set ICRA to ICRD are set to H'20, H'00, H'00, and H'00, respectively (IRQ2 and IRQ3 interr 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.

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

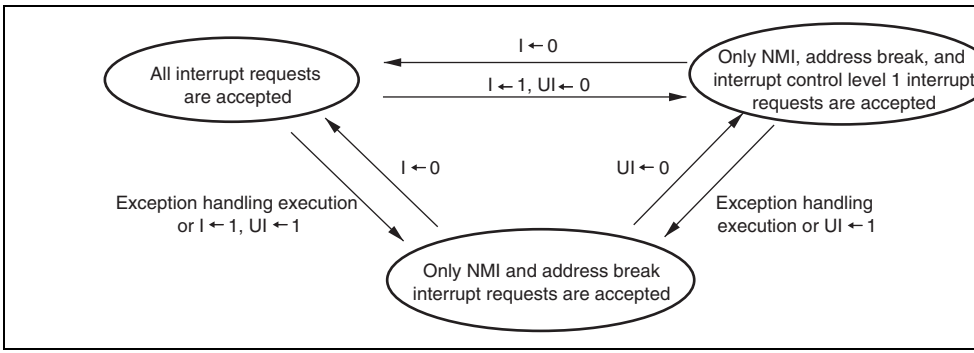


Figure 5.5 State Transition in Interrupt Control Mode 1

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

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

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

4. When the CPU accepts an interrupt request, it starts interrupt exception handling 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.

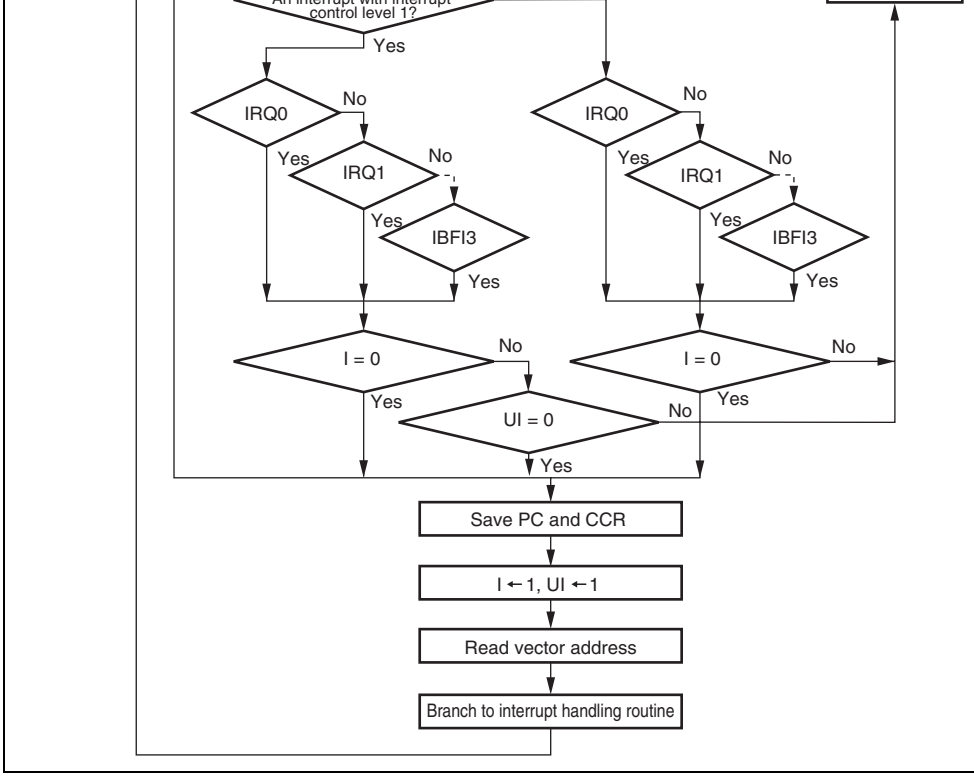


Figure 5.6 Flowchart of Procedure Up to Interrupt Acceptance in Interrup Control Mode 1

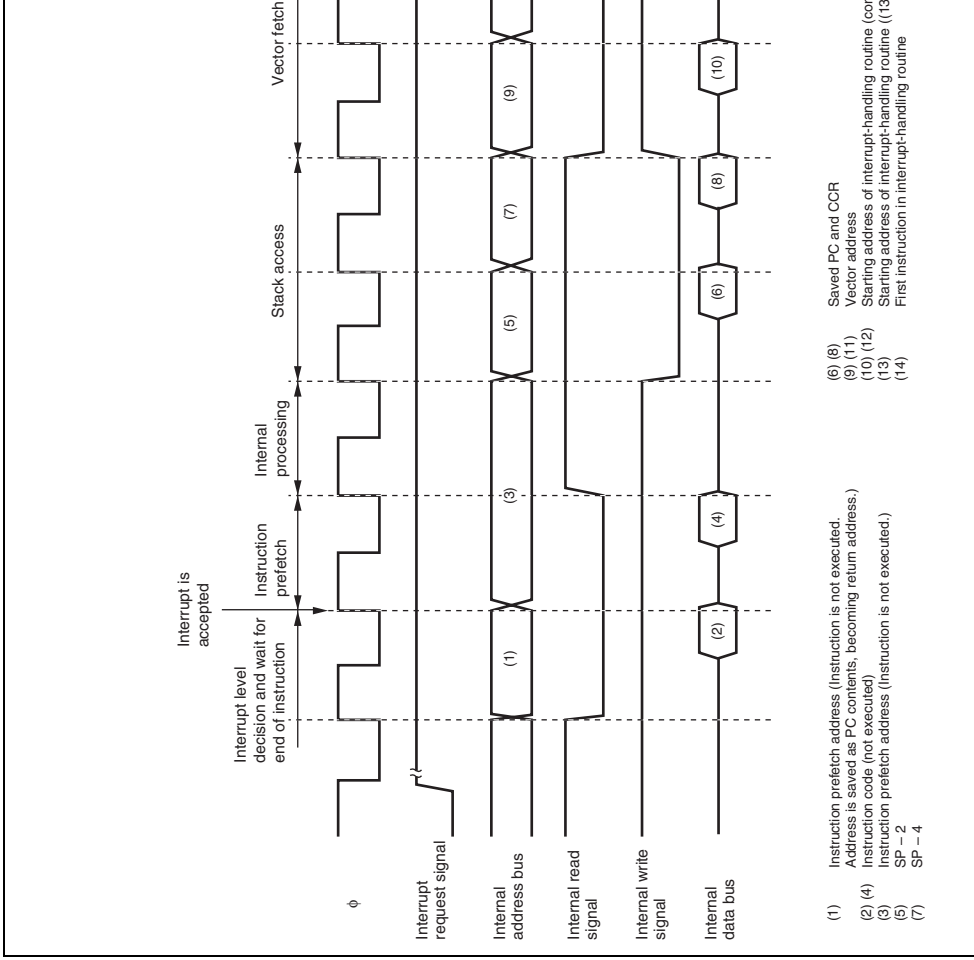


Figure 5.7 Interrupt Exception Handling

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

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

Table 5.8 Number of States in Interrupt Handling Routine Execution Status

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

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

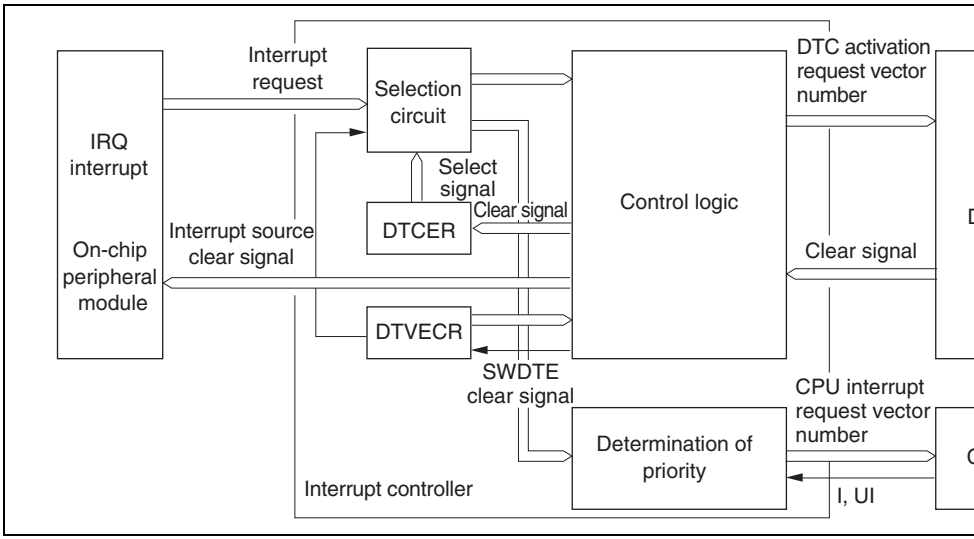


Figure 5.8 Interrupt Control for DTC

The interrupt controller has three main functions in DTC control.

(1) Selection of Interrupt Source

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

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

Table 5.9 Interrupt Source Selection and Clearing Control

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

[Legend]

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

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

×: The relevant interrupt cannot be used.

X: Don't care

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

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

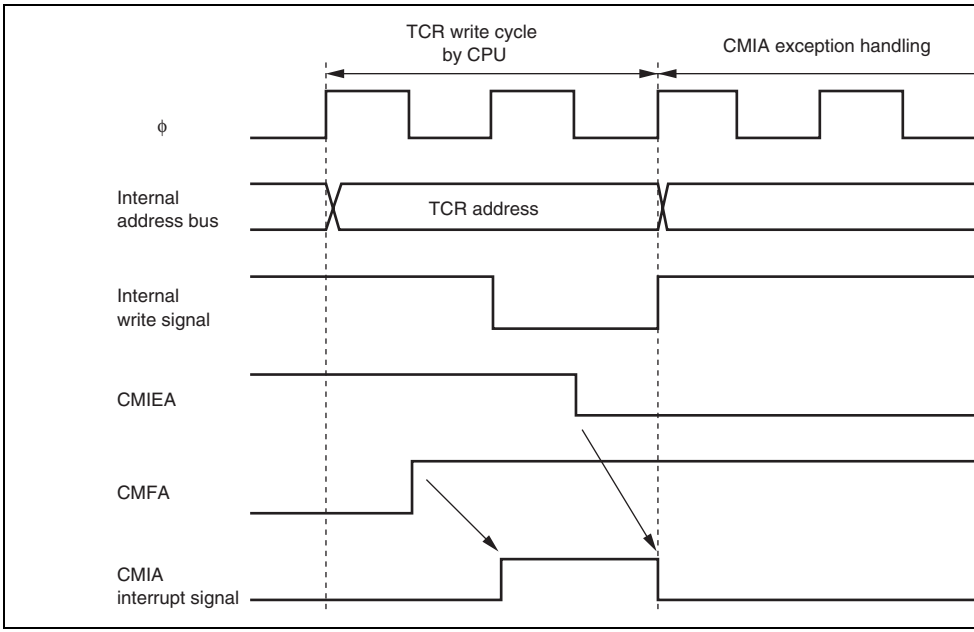


Figure 5.9 Conflict between Interrupt Generation and Disabling

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

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

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

5.7.4 IRQ Status Registers (ISR16, ISR)

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

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

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

- Extended area division

  Possible in normal extended mode

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

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

- Address pin reduction

  In normal extended mode:

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

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

  In address-data multiplex extended mode:

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

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

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

- Control address hold signal and area select signal polarity

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

Program wait states can be inserted for each area.

- Burst ROM interface

  In normal extended mode

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

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

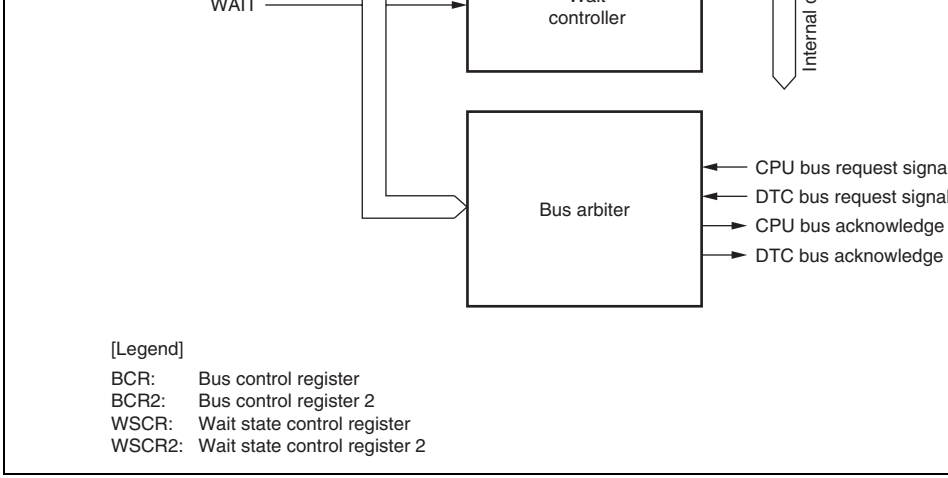
- Idle cycle insertion

  In normal extended mode

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

- Bus arbitration function

  Includes a bus arbiter that arbitrates bus mastership between the CPU 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

		being accessed (when the IOSE bit in SYSCR is 1).
$\overline{CS256}$	Output	Chip select signal indicating that the 256-Kbyte exte area is being accessed (when the CS256E bit in SY 1).
$\overline{RD}$	Output	Strobe signal indicating that the external address sp being read.
$\overline{HWR}$	Output	Strobe signal indicating that the external address sp being written to, and the upper half (D15 to D8, AD1 AD8) of the data bus is valid.
$\overline{LWR}$	Output	Strobe signal indicating that the external address sp being written to, and the lower half (D7 to D0, AD7 t the data bus is valid.
$\overline{WAIT}$	Input	Wait request signal when accessing the external spa
$\overline{WR}$	Output	Strobe signal indicating that the external address sp being written to.
$\overline{HBE}$	Output	Strobe signal indicating that the external address sp being accessed, and the upper half (D15 to D8) of th bus is valid.
$\overline{LBE}$	Output	Strobe signal indicating that the external address sp being accessed, and the lower half (D7 to D0) of the bus is valid.
$\overline{AH}$	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.

6.3.1 Bus Control Register (BCR)

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

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

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

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

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

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

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

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

6.3.5 System Control Register 2 (SYSCR2)

SYSCR2 controls the address-data multiplex operation.

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

(a) Bus Width

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

(b) Number of Access States

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

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

(c) Wait Mode and Number of Program Wait States

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

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

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

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

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

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

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

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

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

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

[Legend]
X: Don't care

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

[Legend]

X: Don't care

- IOS Extended Area

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

- 256-Kbyte Extended Area

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

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

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

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

[Legend]

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

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

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

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

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

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

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

6.4.2 Advanced Mode

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

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

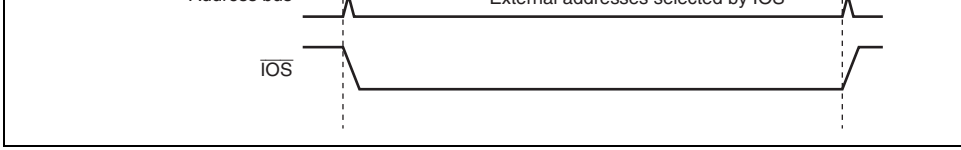


Figure 6.2 $\overline{\text{IOS}}$ Signal Output Timing

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

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

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

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

6.3.1 Data Size and Data Alignment

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

(1) 8-Bit Access Space

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

The upper data bus (AD15 to AD8) is used in address-data multiplex extended mode.

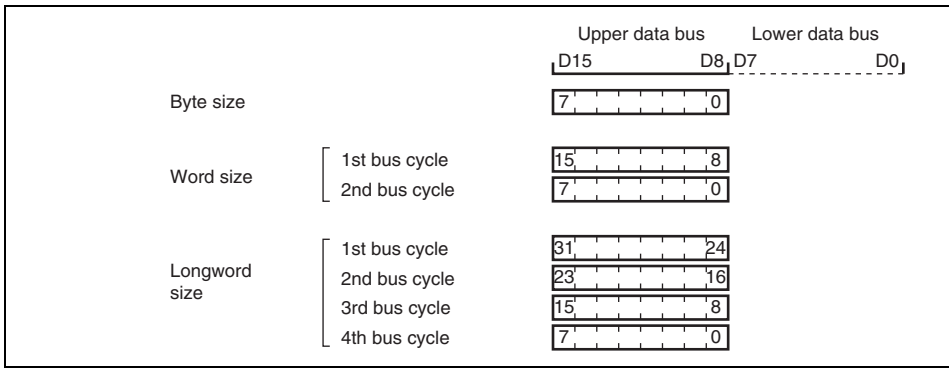


Figure 6.3 Access Sizes and Data Alignment Control (8-bit Access Space

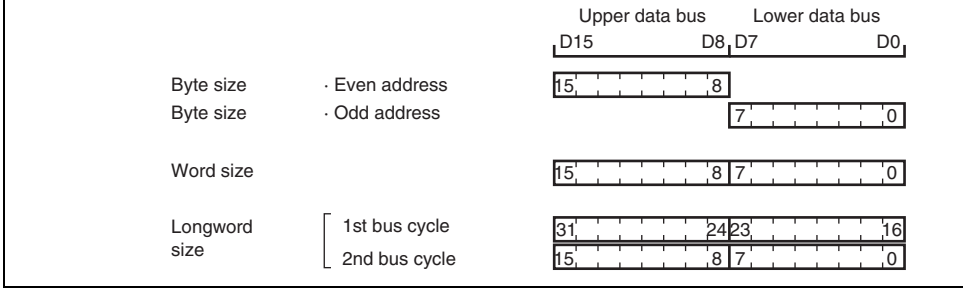


Figure 6.4 Access Sizes and Data Alignment Control (16-bit Access Space)

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

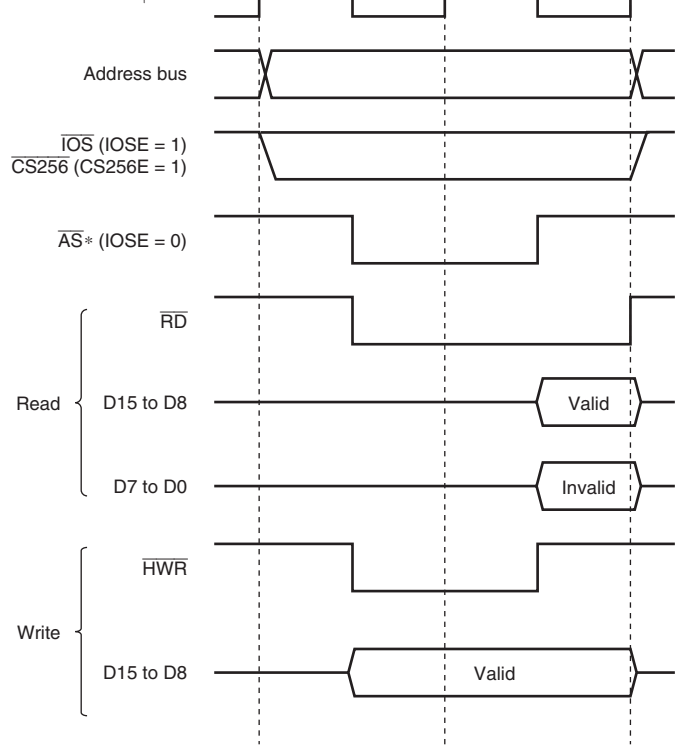
8-bit access space	Byte	Read	—	RD	Valid	Ports o
		Write	—	WR		
16-bit access space	Byte	Read	Even	RD, HBE	Valid	Invalid
			Odd	RD, LBE	Invalid	Valid
		Write	Even	WR, HBE	Valid	Undefi
			Odd	WR, LBE	Undefined	Valid
	Word	Read	—	RD, HBE, LBE	Valid	Valid
		Write	—	WR, HBE, LBE		

[Legend]

Undefined: Undefined data is output.

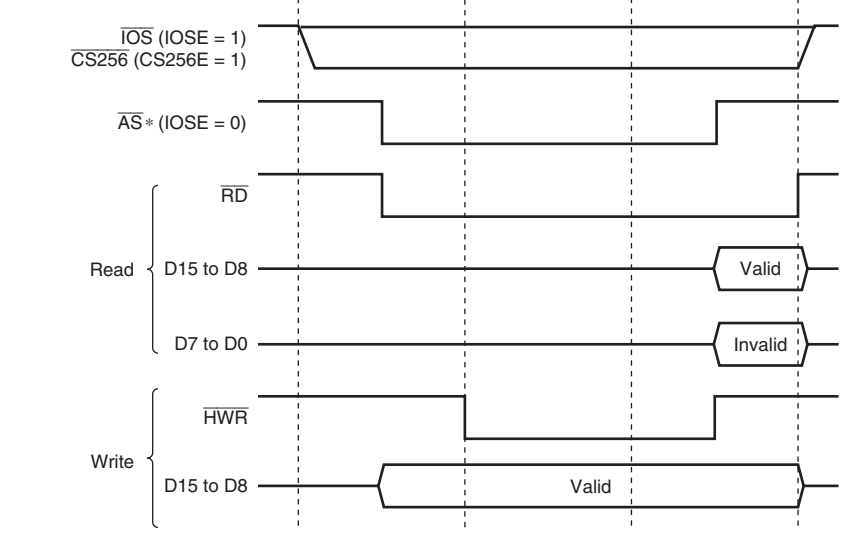
Invalid: Input state with the input value ignored.

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



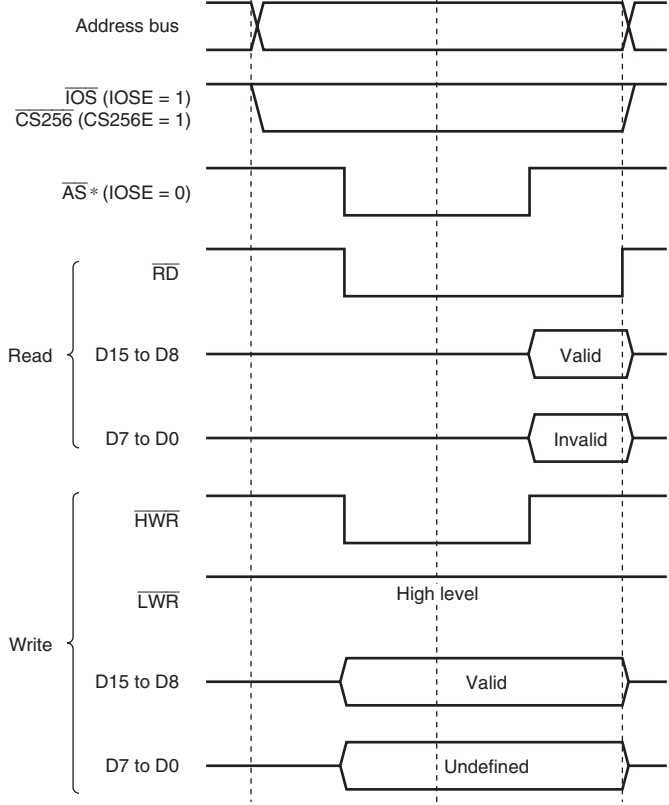
Note: * For external address space access, this signal is not output when the 256-Kbyte extended are
is accessed with CS256E = 1.

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



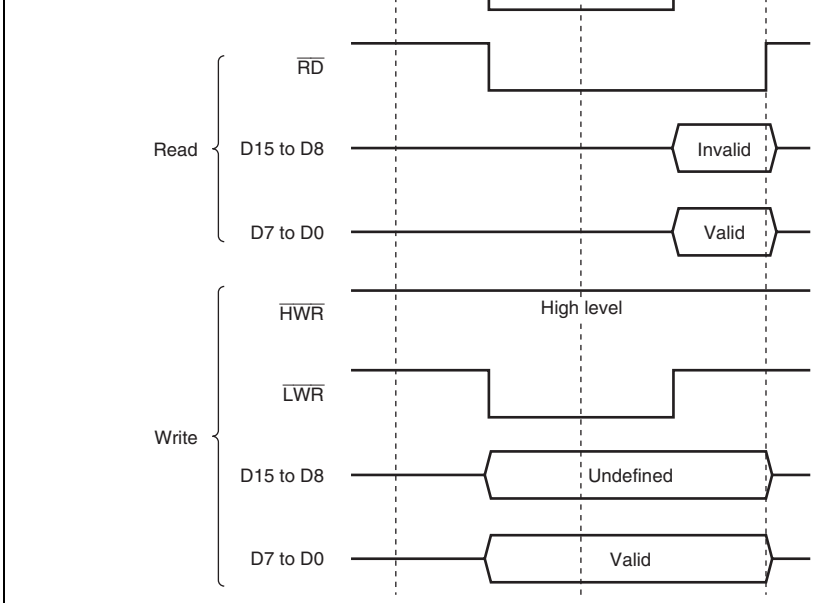
Note: * For external address space access, this signal is not output when the 256-Kbyte extended area is accessed with CS256E = 1.

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



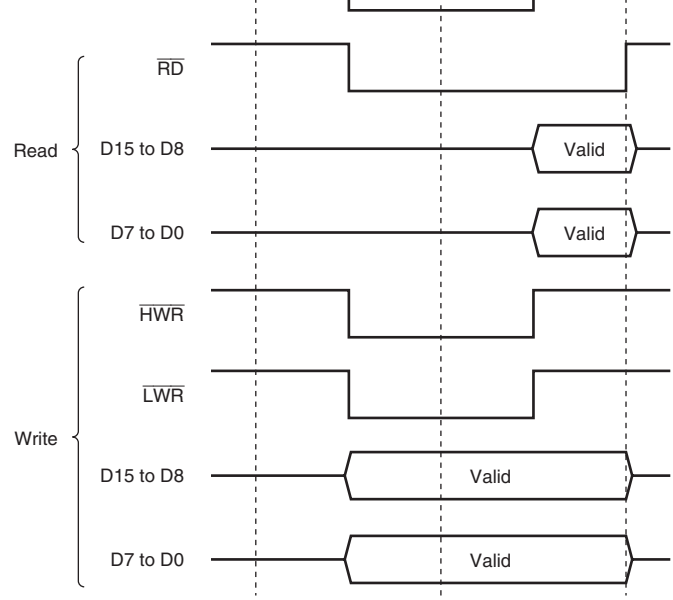
Note: * For external address space access, this signal is not output when the 256-Kbyte extended are
is accessed with CS256E = 1.

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



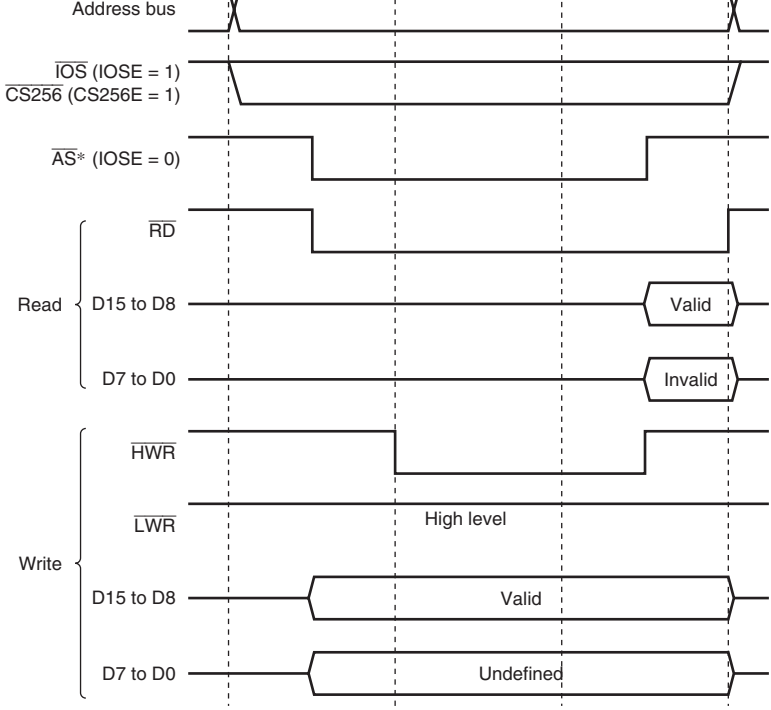
Note: * For external address space access, this signal is not output when the 256-Kbyte extended area is accessed with CS256E = 1.

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



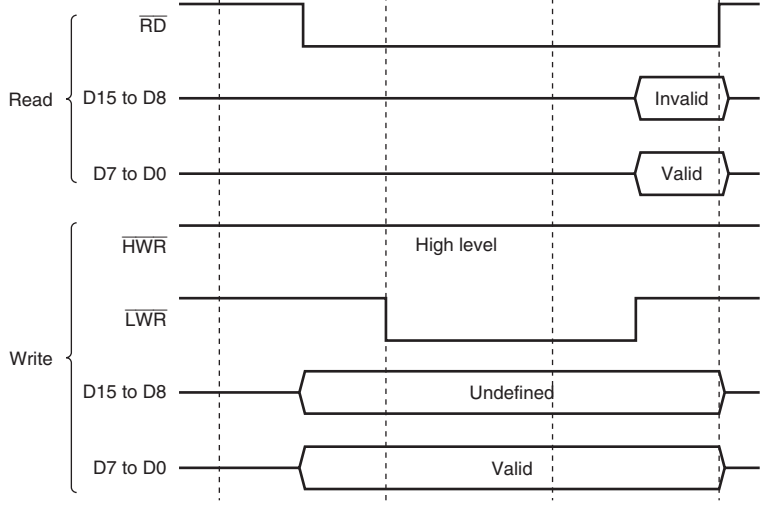
Note: * For external address space access, this signal is not output when the 256-Kbyte extended area is accessed with CS256E = 1.

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



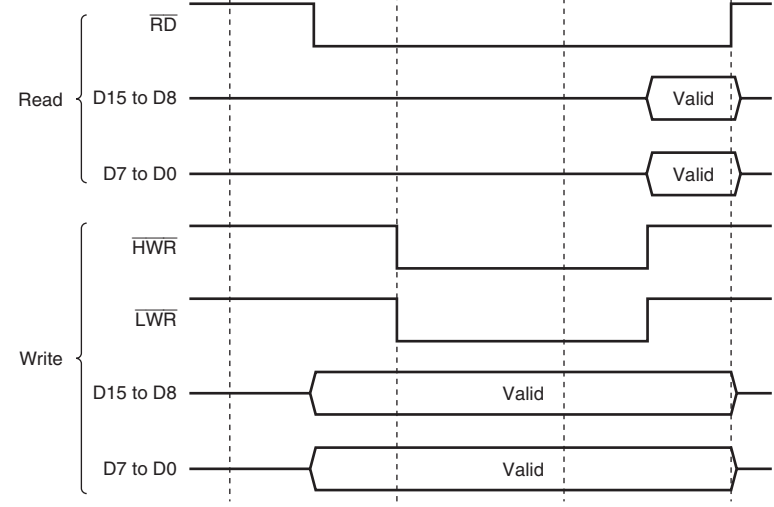
Note: * For external address space access, this signal is not output when the 256-Kbyte extended area is accessed with CS256E = 1.

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



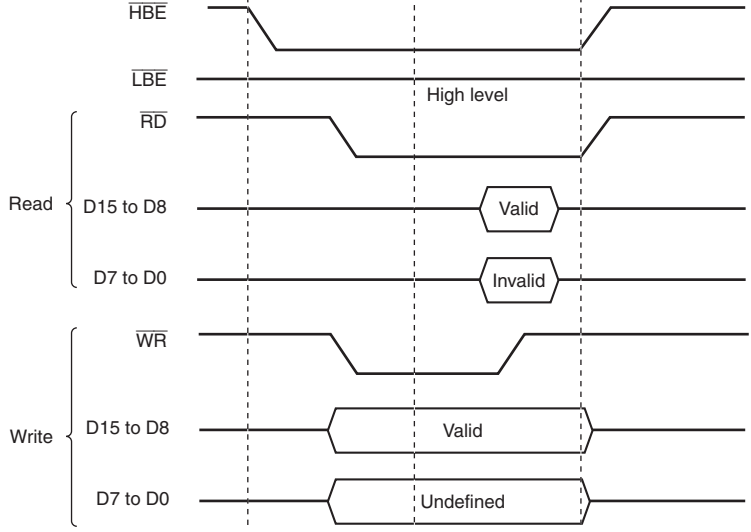
Note: * For external address space access, this signal is not output when the 256-Kbyte extended area is accessed with CS256E = 1.

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



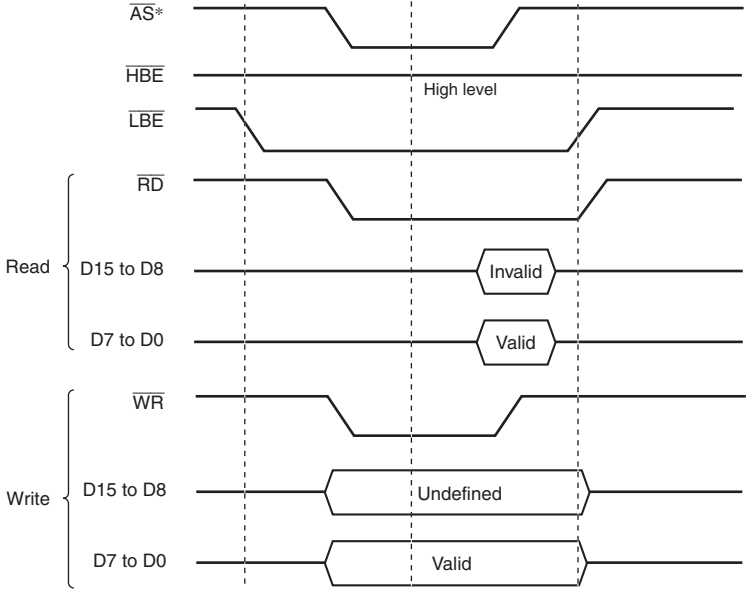
Note: * For external address space access, this signal is not output when the 256-Kbyte extended area is accessed with CS256E = 1.

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



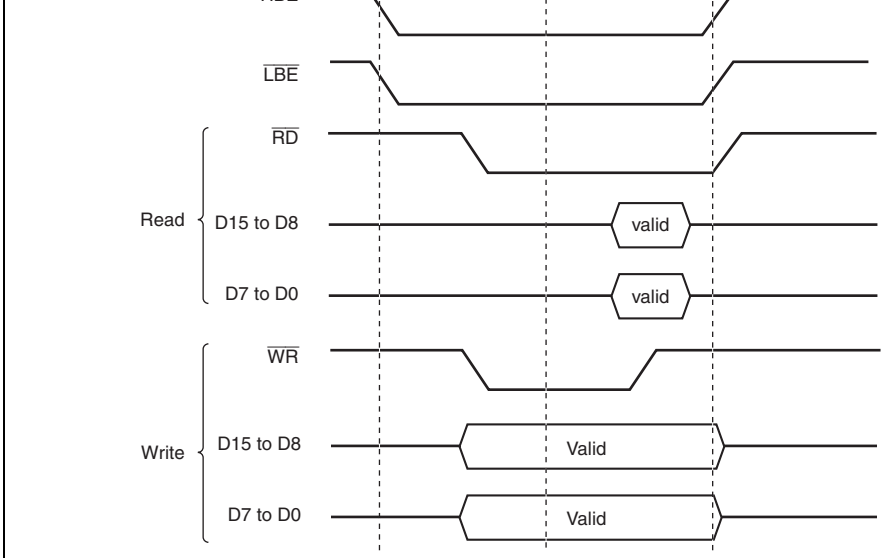
Note: * For external address space access, this signal is not output when the 256-Kbyte extended are is accessed with CS256E = 1.

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



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

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



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

Figure 6.15 Glueless Extension Word Access (ADMXE = 0)

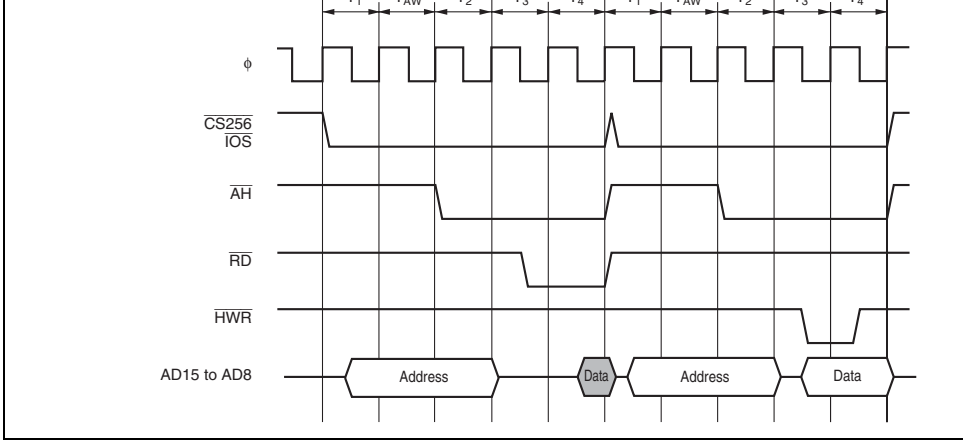


Figure 6.16 Bus Timing for 8-Bit, 2-State Access Space

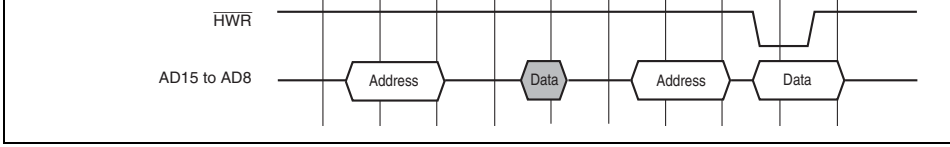


Figure 6.17 Bus Timing for 8-Bit, 2-State Access Space

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

Figure 6.18 shows the bus timing for an 8-bit, 3-state access space. When an 8-bit acces
accessed, the upper half (AD15 to AD8) of the data bus is used. Wait states can be inser

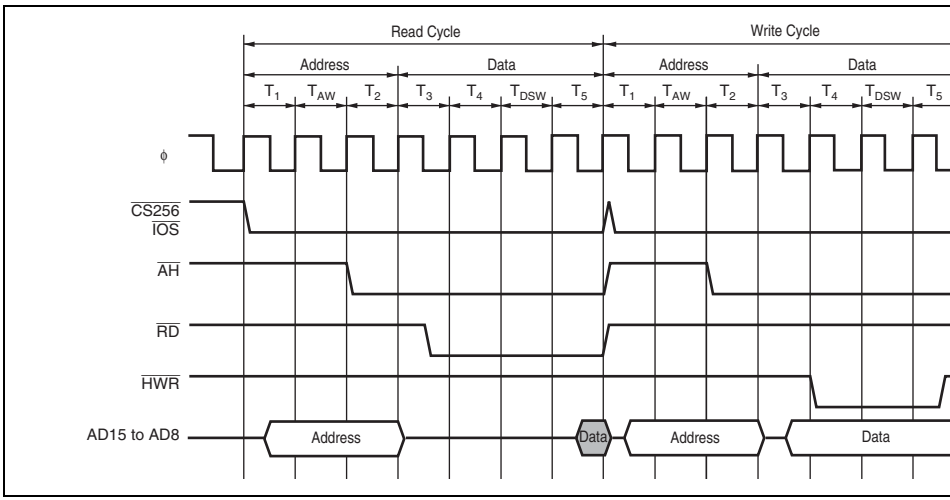


Figure 6.18 Bus Timing for 8-Bit, 3-State Access Space

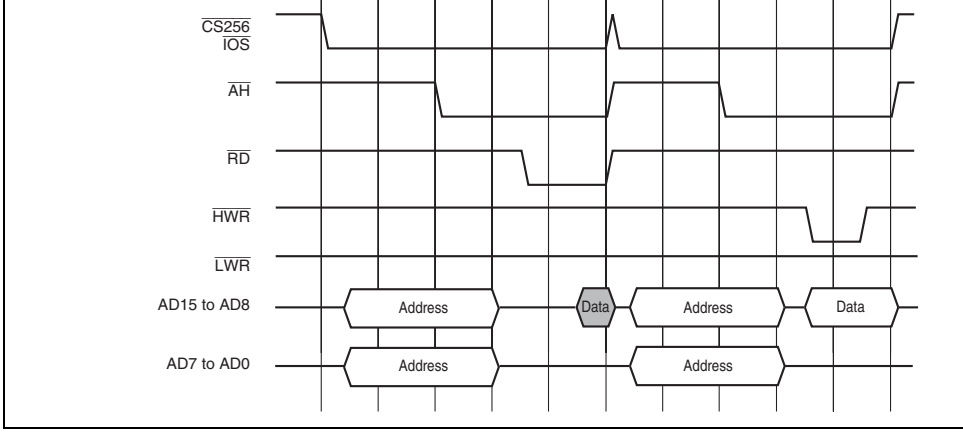


Figure 6.19 Bus Timing for 16-Bit, 2-State Access Space (1) (Even Byte Acce

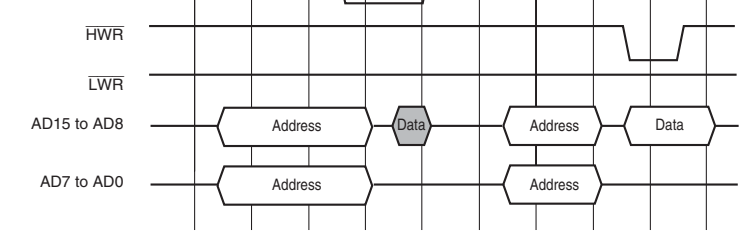


Figure 6.20 Bus Timing for 16-Bit, 2-State Access Space (2) (Even Byte Acc

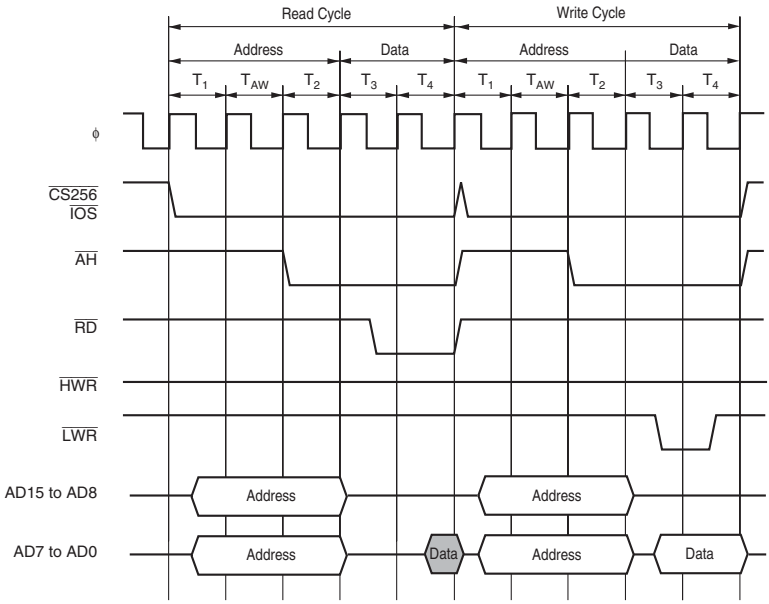


Figure 6.21 Bus Timing for 16-Bit, 2-State Access Space (3) (Odd Byte Acc

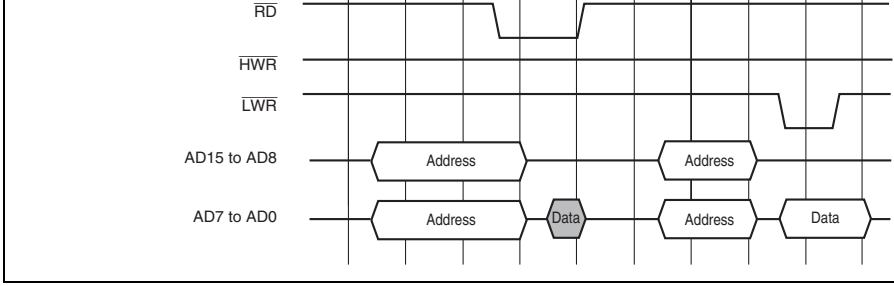


Figure 6.22 Bus Timing for 16-Bit, 2-State Access Space (4) (Odd Byte Acces

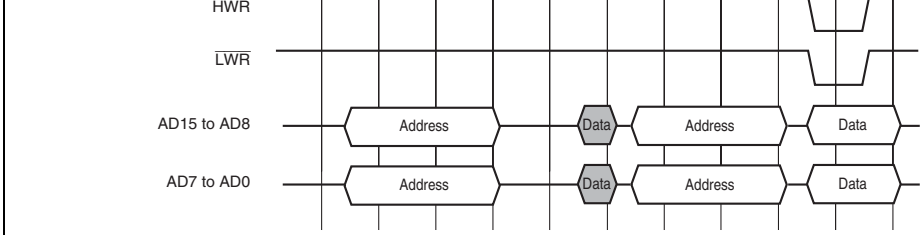


Figure 6.23 Bus Timing for 16-Bit, 2-State Access Space (5) (Word Acces

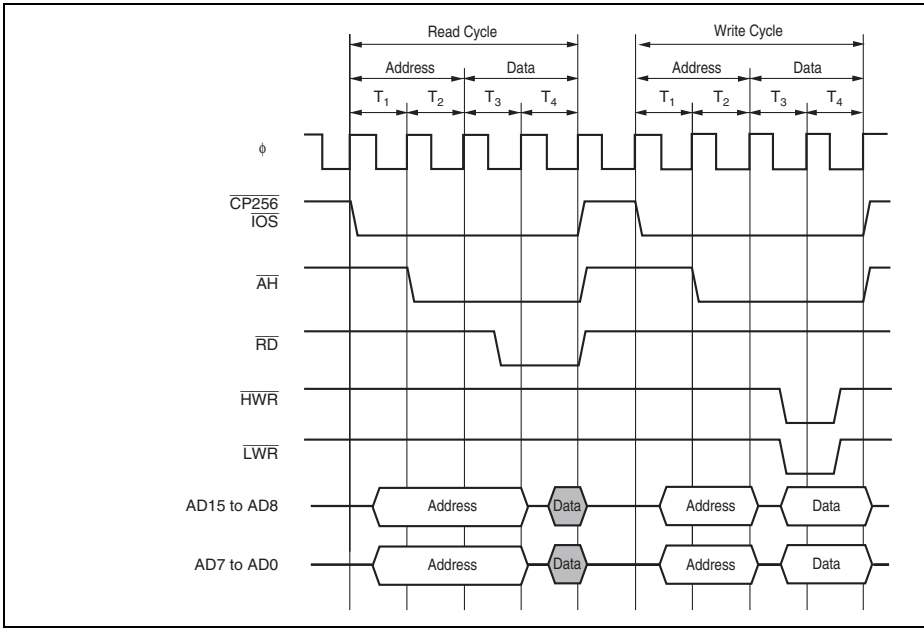


Figure 6.24 Bus Timing for 16-Bit, 2-State Access Space (6) (Word Acces

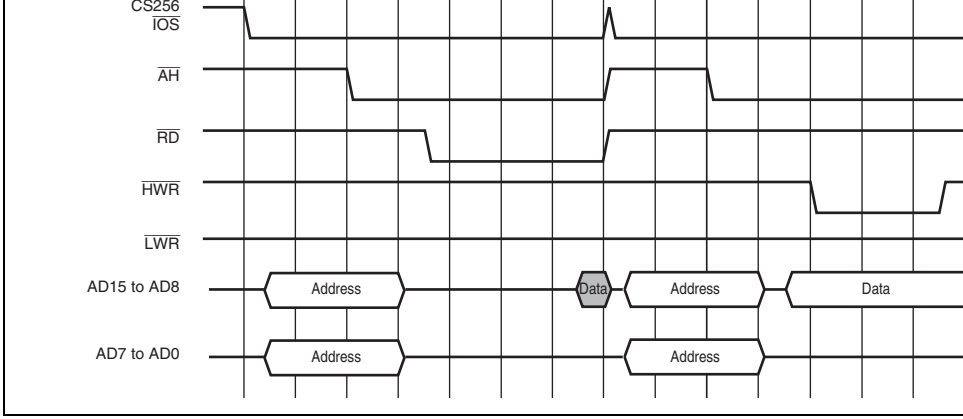


Figure 6.25 Bus Timing for 16-Bit, 3-State Access Space (1) (Even Byte Acce

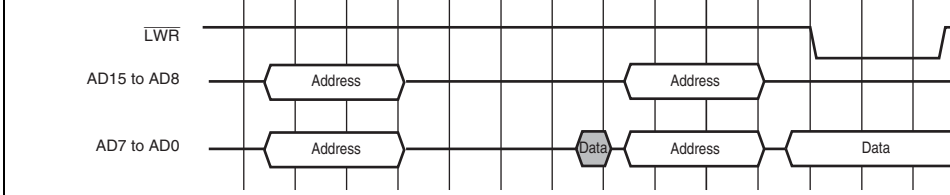


Figure 6.26 Bus Timing for 16-Bit, 3-State Access Space (2) (Odd Byte Acc

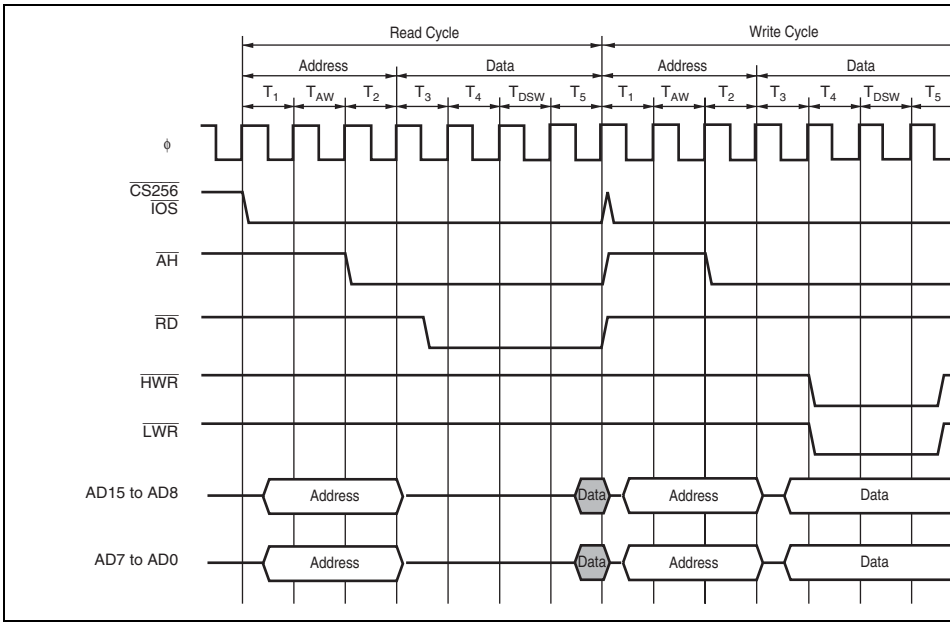


Figure 6.27 Bus Timing for 16-Bit, 3-State Access Space (3) (Word Acces

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

(b) Pin Wait Mode

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

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

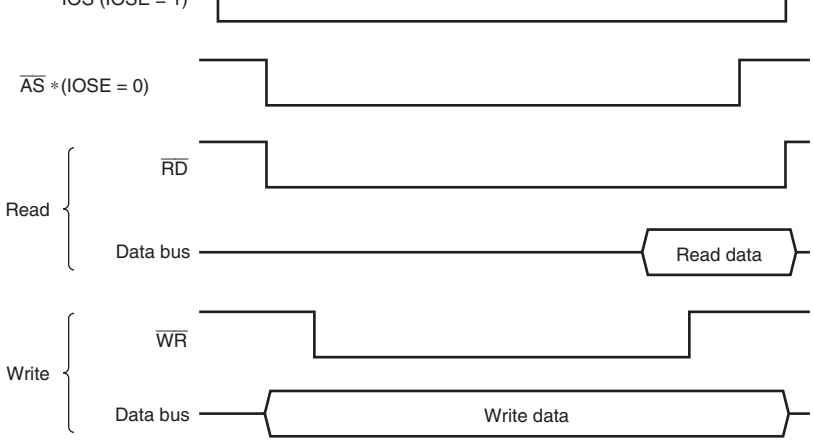
(c) Pin Auto-Wait Mode

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

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

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

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



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

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

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

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

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

(c) Pin Auto-Wait Mode

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

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

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

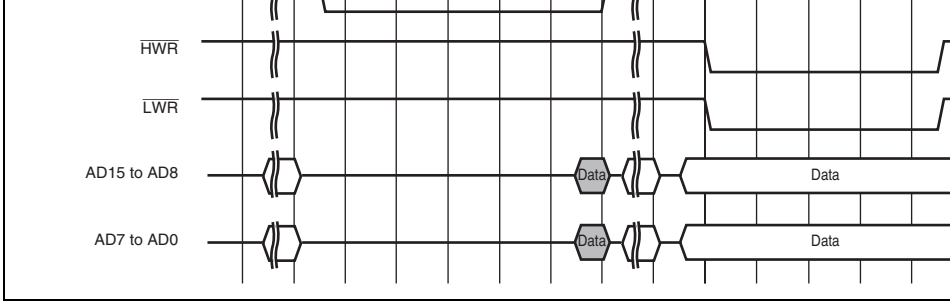


Figure 6.29 Example of Wait State Insertion Timing

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

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

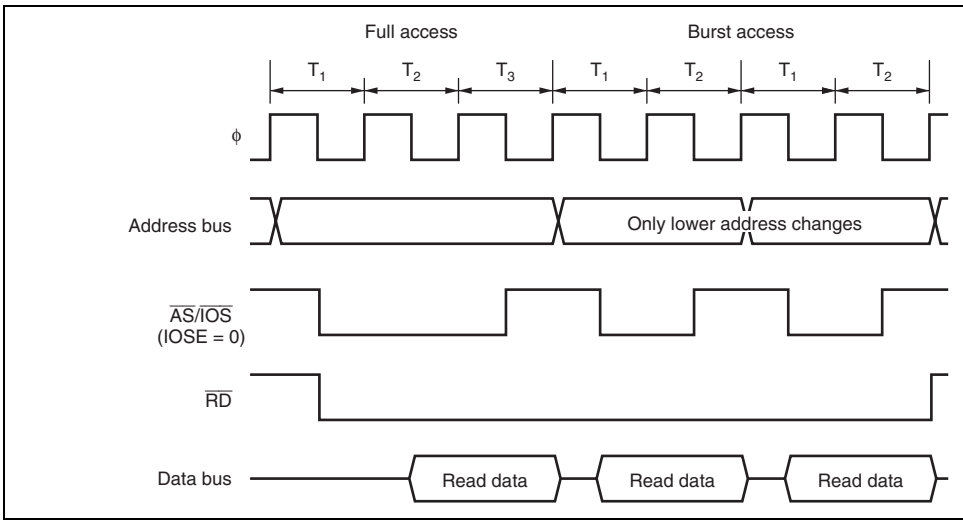


Figure 6.30 Access Timing Example in Burst ROM Space (AST = BRSTS1 =

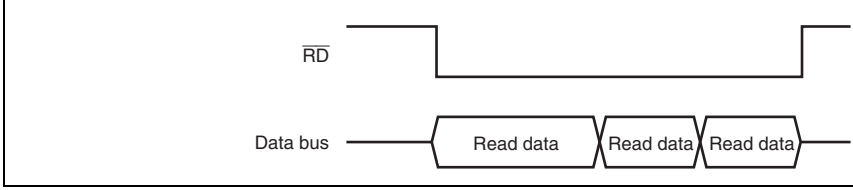


Figure 6.31 Access Timing Example in Burst ROM Space (AST = BRSTS1

6.6.2 Wait Control

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

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

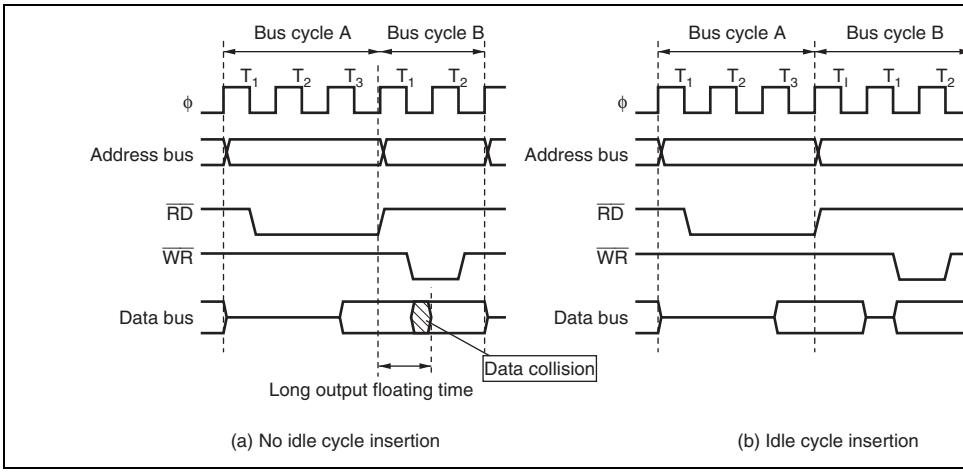


Figure 6.32 Examples of Idle Cycle Operation

6.8 Bus Arbitration

6.8.1 Overview

The BSC has a bus arbiter that arbitrates bus master operations. There are two bus maste
CPU and DTC – that perform read/write operations while they have bus mastership.

6.8.2 Operation

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

(High) DTC > CPU (Low)

mastership is as follows:

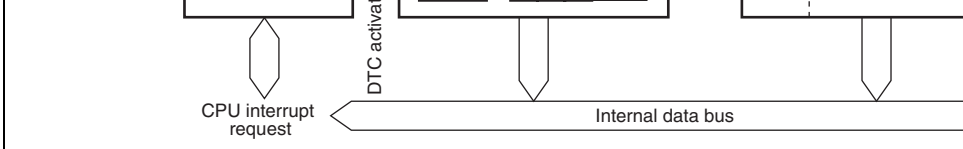
- Bus mastership is transferred at a break between bus cycles. However, if bus cycle is
  in discrete operations, as in the case of a long-word size access, the bus is not transfer
  break between the operations. For details see section 2.7, Bus States During Instructio
  Execution in the H8S/2600 Series,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

(2) DTC

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

7.1 Features

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



[Legend]

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

Figure 7.1 Block Diagram of DTC

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

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

				(by +1 when Sz = 0, by +2 when Sz = 1)
				11: SAR is decremented after a transfer (by –1 when Sz = 0, by –2 when Sz = 1)
5	DM1	Undefined	—	Destination Address Mode 1 and 0
4	DM0			These bits specify a DAR operation after a dat transfer.
				0*: DAR is fixed
				10: DAR is incremented after a transfer (by +1 when Sz = 0, by +2 when Sz = 1)
				11: DAR is decremented after a transfer (by –1 when Sz = 0, by –2 when Sz = 1)
3	MD1	Undefined	—	DTC Mode
2	MD0			These bits specify the DTC transfer mode.
				00: Normal transfer mode
				01: Repeat transfer mode
				10: Block transfer mode
				11: Setting prohibited
1	DTS	Undefined	—	DTC Transfer Mode Select
				Specifies whether the source side or the destin side is set to be a repeat area or block area in transfer mode or block transfer mode.
				0: Destination side is repeat area or block area
				1: Source side is repeat area or block area
0	Sz	Undefined	—	DTC Data Transfer Size
				Specifies the size of data to be transferred.
				0: Byte-size transfer
				1: Word-size transfer

Note: * Don't care

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

7.2.3 DTC Source Address Register (SAR)

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

7.2.4 DTC Destination Address Register (DAR)

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

and the lowest eight bits as CRAL. CRAH holds the value for the number of data transfer
CRAL functions as an 8-bit transfer counter (1 to 256). CRAL is decremented by 1 every
data is transferred, and the contents of CRAH are transferred when the counter value reac
The number of times data is transferred is one when CRAH = CRAL = H'01, 255 when C
CRAL = H'FF, and 256 when CRAH = CRAL = H'00.

In block transfer mode CRA is divided in two, with the highest eight bits designated as C
the lowest eight bits as CRAL. CRAH holds the value for the block size, and CRAL funct
an 8-bit block size counter (1 to 256). CRAL is decremented by 1 every time data is trans
and the contents of CRAH are transferred when the counter value reaches H'00. The bloc
one byte (or one word) when CRAH = CRAL = H'01, 255 bytes (or 255 words) when CR
CRAL = H'FF, and 256 bytes (or 256 words) when CRAH = CRAL = H'00.

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.

	DTCE0	Setting this bit to 1 specifies a relevant interru as a DTC activation source. [Clearing conditions] • When data transfer has ended with the DI MRB set to 1 • When the specified number of transfers ha These bits are not cleared when the DISEL bi the specified number of transfers have not be completed

Table 7.1 Correspondence between Interrupt Sources and DTCER

		Register				
Bit	**Bit Name**	**DTCERA**	**DTCERB**	**DTCERC**	**DTCERD**	**DTC**
7	DTCEn7	(16)IRQ0	—	—	(86)TXI1	—
6	DTCEn6	(17)IRQ1	(76)IICI2	—	—	—
5	DTCEn5	(18)IRQ2	(94)IICI0	—	—	—
4	DTCEn4	(19)IRQ3	—	(29)EVENTI	(78)IICI3	—
3	DTCEn3	(28)ADI	—	—	(98)IICI1	(104
2	DTCEn2	—	—	(81)RXI3	—	(105
1	DTCEn1	—	—	(82)TXI3	—	(106
0	DTCEn0	—	—	(85)RXI1	—	(107

[Legend]

n: A to E

( ): Vector number

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

				• When the DISEL bit is 0 and the specified nu transfers have not ended • When 0 is written to the DISEL bit after a soft activated data transfer end interrupt (SWDTE request has been sent to the CPU. This bit will not be cleared when the DISEL bit is data transfer has ended or when the specified nu transfers has ended.
6 to 0	DTVEC6 to DTVEC0	All 0	R/W	DTC Software Activation Vectors 6 to 0 These bits specify a vector number for DTC softw activation. The vector address is expressed as H'0400 + (ve number × 2). For example, when DTVEC6 to DTV H'10, the vector address is H'0420. When the SW is 0, these bits can be written to.

				These bits are always read as 0 and cannot be r
4 to 0	—	All 0	R/W	Reserved The initial value should not be changed.

7.2.10 Event Counter Control Register (ECCR)

ECCR selects the event counter channels for use and the detection edge.

Bit	Bit Name	Initial Value	R/W	Description
7	EDSB	0	R/W	Event Counter Edge Select Selects the detection edge for the event counter 0: Counts the rising edges 1: Counts the falling edges
6 to 4	—	All 0	R	Reserved These bits are always read as 0 and cannot be

				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

7.2.11 Event Counter Status Register (ECS)

ECS is a 16-bit register that holds events temporarily. The DTC decides the counter to be incremented according to the state of this register. Reading this register allows the monitor events that are not yet counted by the event counter. Access in 8-bit unit is not allowed.

Bit	Bit Name	Initial Value	R/W	Description
15 to 0	E15 to E0	All 0	R	Event Monitor 15 to 0 These bits indicate processed/unprocessed state events that are input to EVENT15 to EVENT0. 0: The corresponding event is already processed 1: The corresponding event is not yet processed

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

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

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

													1	0	
													1	0	0
												1	0	0	0
											1	0	0	0	0
										1	0	0	0	0	0
									1	0	0	0	0	0	0
								1	0	0	0	0	0	0	0
							1	0	0	0	0	0	0	0	0
						1	0	0	0	0	0	0	0	0	0
					1	0	0	0	0	0	0	0	0	0	0
				1	0	0	0	0	0	0	0	0	0	0	0
			1	0	0	0	0	0	0	0	0	0	0	0	0
		1	0	0	0	0	0	0	0	0	0	0	0	0	0
	1	0	0	0	0	0	0	0	0	0	0	0	0	0	0
1	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.

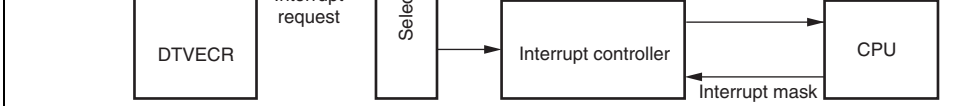


Figure 7.2 Block Diagram of DTC Activation Source Control

then reads the register information from that start address.

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

The configuration of the vector address is a 2-byte unit. Specify the lower two bytes of t information start address.

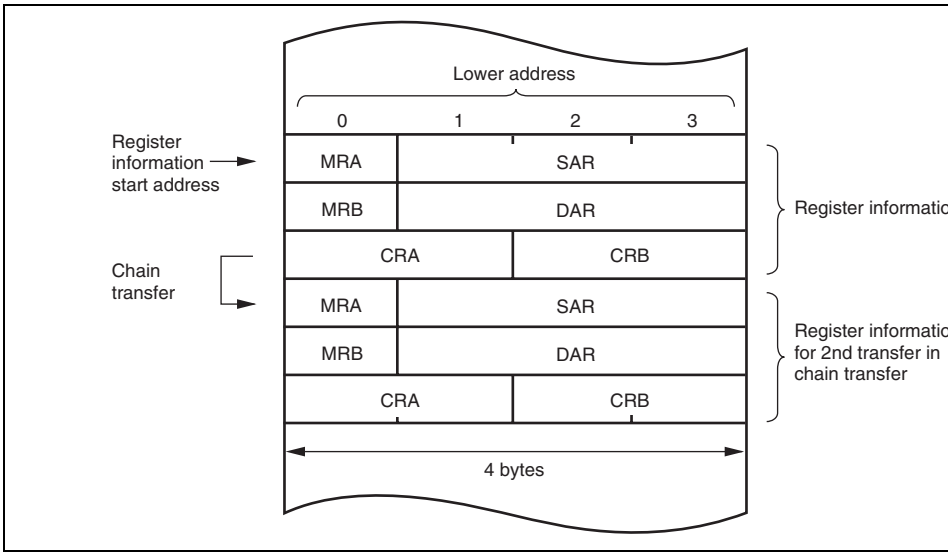


Figure 7.3 DTC Register Information Location in Address Space

Table 7.4 Interrupt Sources, DTC Vector Addresses, and Corresponding DTCEs

Activation Source Origin	Activation Source	Vector Number	DTC Vector Address	DTCE*
Software	Write to DTVECR	DTVECR	H'0400 + (vector number x 2)	—
External pins	IRQ0	16	H'0420	DTCEA7
	IRQ1	17	H'0422	DTCEA6
	IRQ2	18	H'0424	DTCEA5
	IRQ3	19	H'0426	DTCEA4
A/D converter	ADI	28	H'0438	DTCEA3
EVC	EVENTI	29	H'043A	DTCEC4
IIC_2	IICI2	76	H'0498	DTCEB6
IIC_3	IICI3	78	H'049C	DTCED4
SCI_3	RXI3	81	H'04A2	DTCEC2
	TXI3	82	H'04A4	DTCEC1
SCI_1	RXI1	85	H'04AA	DTCEC0
	TXI1	86	H'04AC	DTCED7
IIC_0	IICI0	94	H'04BC	DTCEB5
IIC_1	IICI1	98	H'04C4	DTCED3
LPC	ERRI	104	H'04D0	DTCEE3
	IBFI1	105	H'04D2	DTCEE2
	IBFI2	106	H'04D4	DTCEE1
	IBFI3	107	H'04D6	DTCEE0

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

transfer destination address. After each transfer, SAR and DAR are independently incre
decremented, or left fixed depending on its register information.

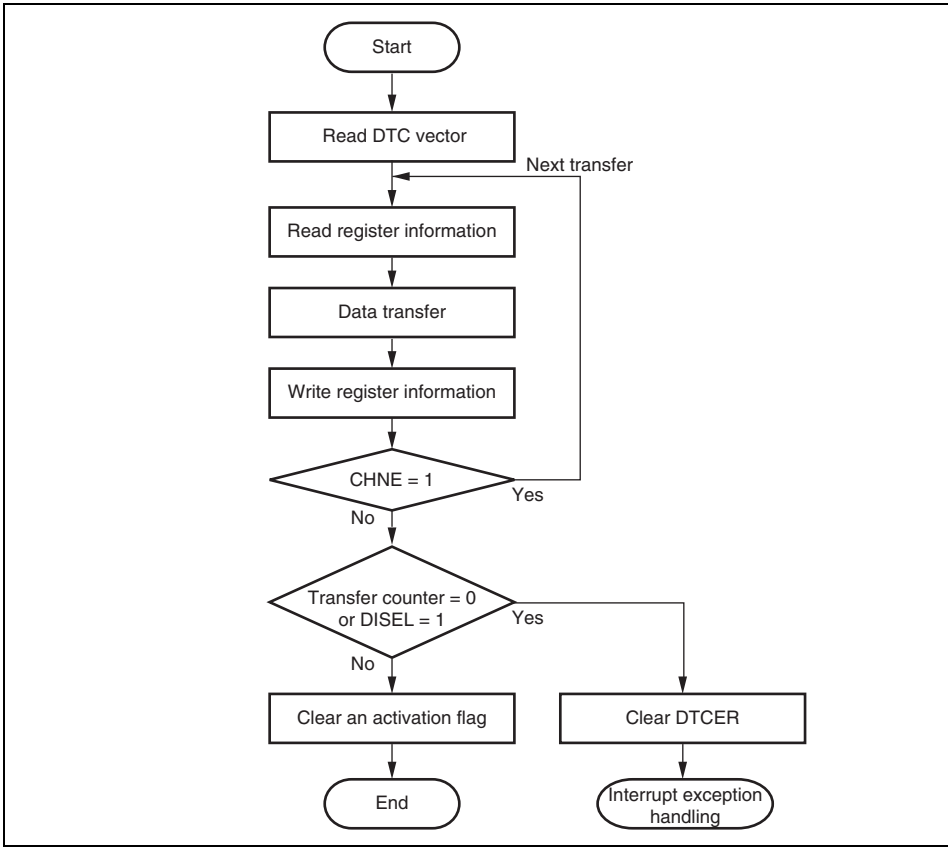


Figure 7.5 DTC Operation Flowchart

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

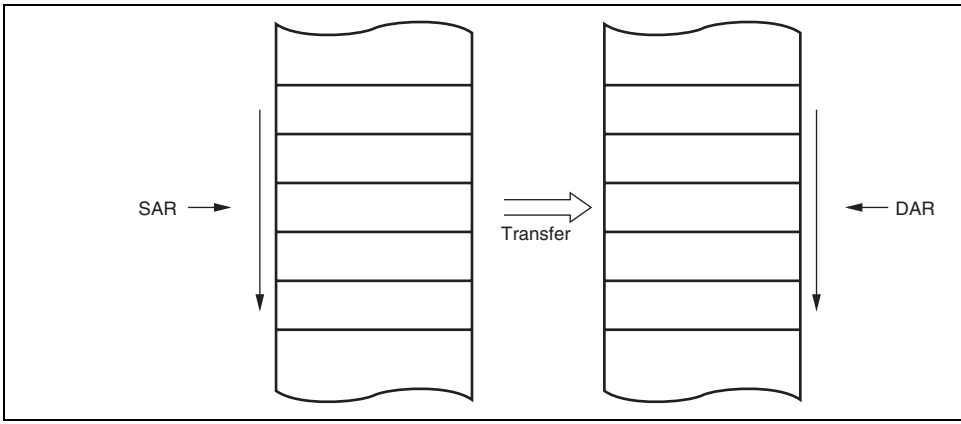


Figure 7.6 Memory Mapping in Normal Transfer Mode

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

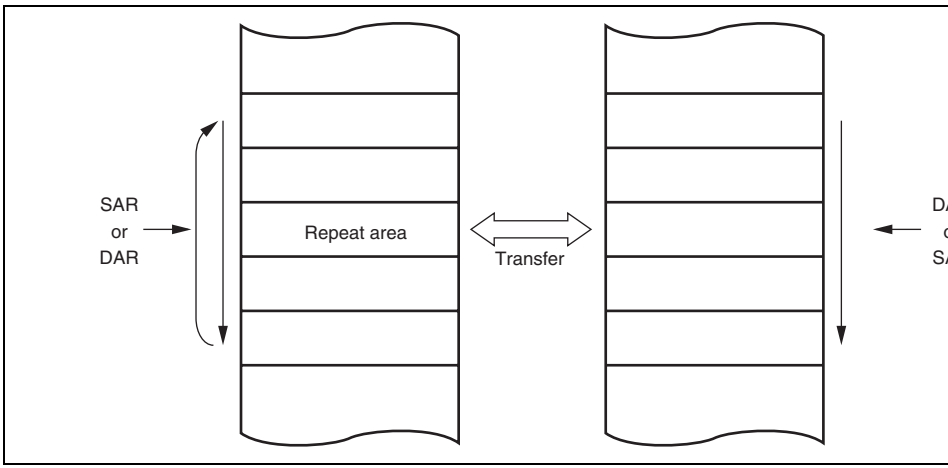


Figure 7.7 Memory Mapping in Repeat Transfer Mode

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

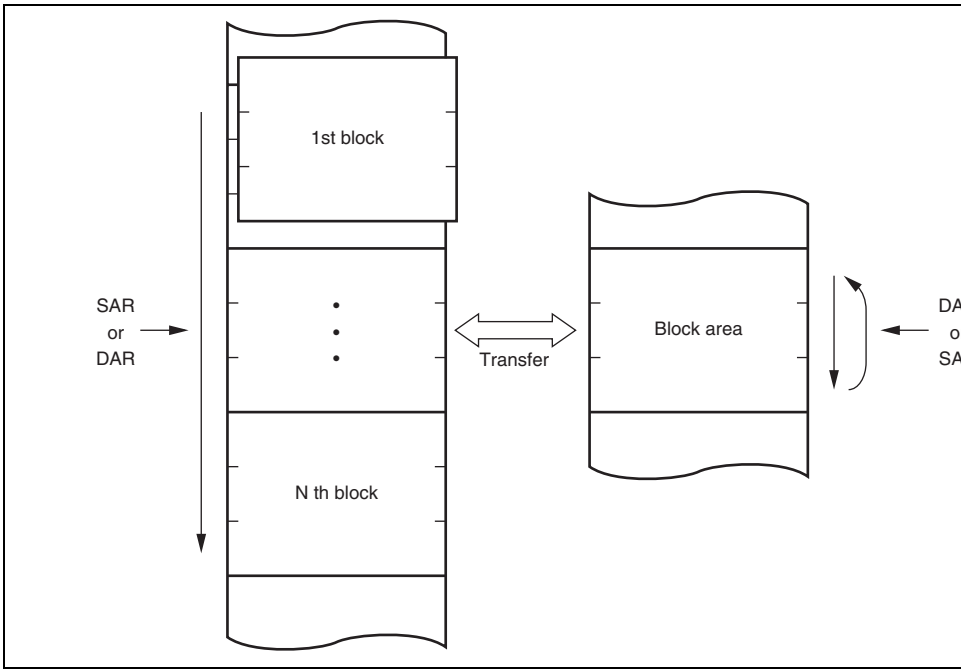


Figure 7.8 Memory Mapping in Block Transfer Mode

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

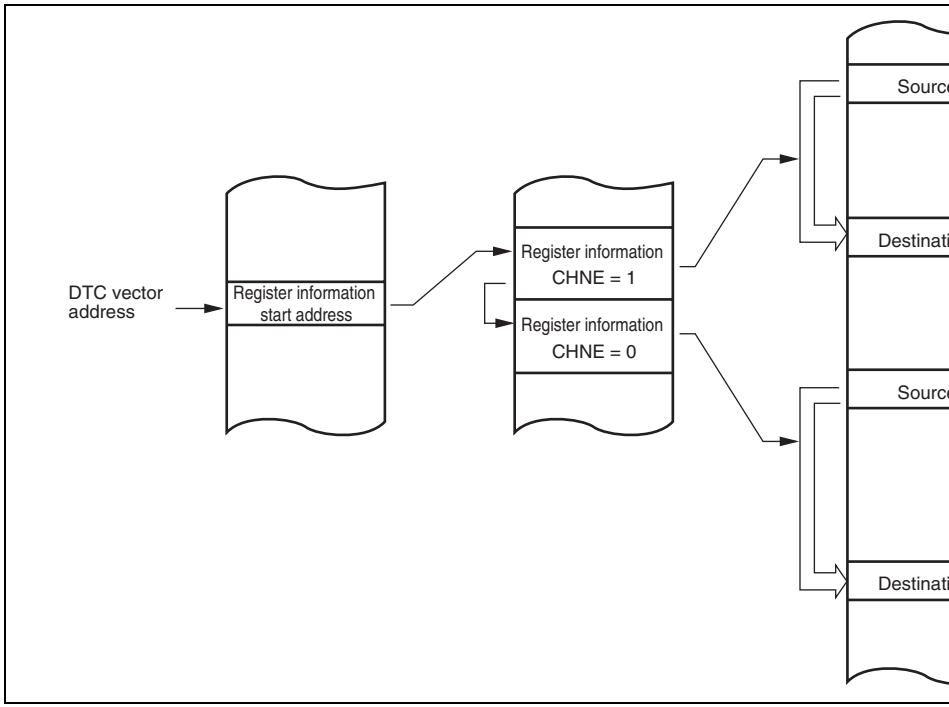


Figure 7.9 Chain Transfer Operation

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

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

7.6.6 Operation Timing

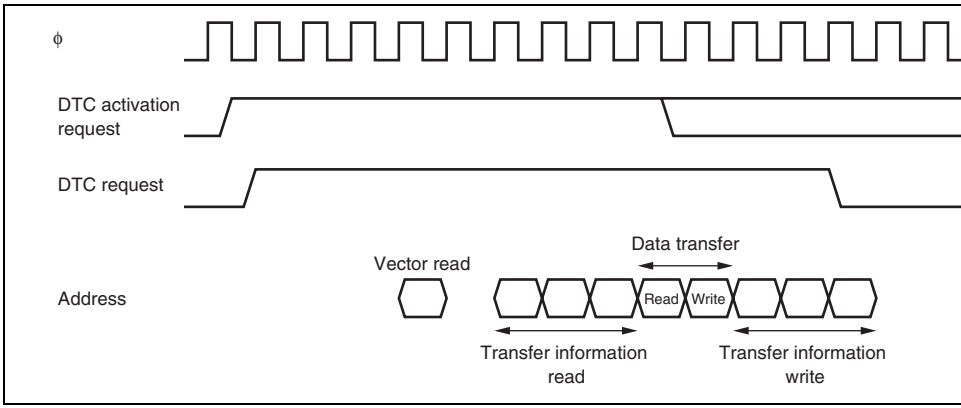


**Figure 7.10 DTC Operation Timing (Example in Normal Transfer Mode or Re
Transfer Mode)**

Figure 7.11 DTC Operation Timing (Example of Block Transfer Mode, with Block Size of 2)

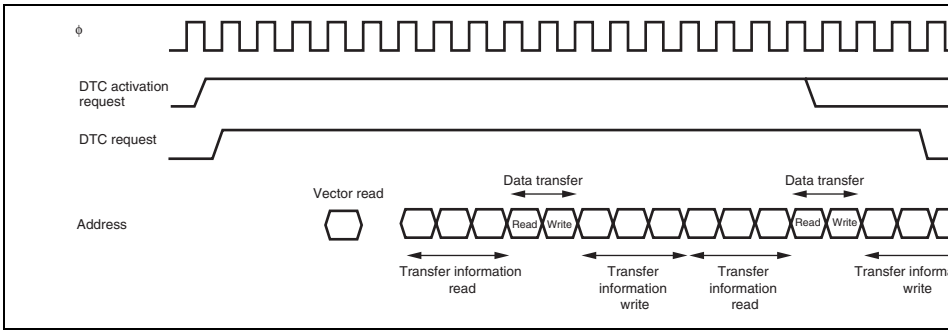


Figure 7.12 DTC Operation Timing (Example of Chain Transfer)

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 7.9 Number of States Required for Each Execution Status

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

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

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

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

7.7.2 Activation by Software

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

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

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

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

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

7.9.3 DTCE Bit Setting

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

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

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

control register (PCR), in addition to DDR and DR, to control the on/off status of the inp
MOSs.

Port 6 has built-in de-bouncers (DBn) that eliminate noises in the input signals.

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

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

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

	control I/O and bidirectional data bus I/O	P65/DB5/$\overline{RTS}$ P64/DB4/$\overline{CTS}$ P63/DB3 P62/DB2 P61/DB1 P60/DB0	D5* D4* D3* D2* D1* D0*	P65/DB5/$\overline{RTS}$ P64/DB4/$\overline{CTS}$ P63/DB3 P62/DB2 P61/DB1 P60/DB0	
Port 7	General input port multiplexed with A/D converter analog input and interrupt input	P77/$\overline{ExIRQ7}$/AN7 P76/$\overline{ExIRQ6}$/AN6 P75/$\overline{ExIRQ5}$/AN5 P74/$\overline{ExIRQ4}$/AN4 P73/$\overline{ExIRQ3}$/AN3 P72/$\overline{ExIRQ2}$/AN2 P71/$\overline{ExIRQ1}$/AN1 P70/$\overline{ExIRQ0}$/AN0		Same as left	
Port 8	General I/O port multiplexed with A/D converter external trigger input, interrupt input, SCI_1 and SCI_3 clock I/O, and IIC_0 and IIC_1 I/O	P87/$\overline{ExIRQ15}$/$\overline{ADTRG}$ P86/$\overline{ExIRQ14}$ P85/$\overline{ExIRQ13}$/SCK1 P84/$\overline{ExIRQ12}$/SCK3		Same as left	
		P83/$\overline{ExIRQ11}$/SDA1 P82/$\overline{ExIRQ10}$/SCL1 P81/$\overline{ExIRQ9}$/SDA0 P80/$\overline{ExIRQ8}$/SCL0		Same as left	NM pu ou

	event counter input and address output	PA5/EVENT5/A21 PA4/EVENT4/A20 PA3/EVENT3/A19 PA2/EVENT2/A18 PA1/EVENT1/A17 PA0/EVENT0/A16	PA5/EVENT5 PA4/EVENT4 PA3/EVENT3 PA2/EVENT2 PA1/EVENT1 PA0/EVENT0	
Port B	General I/O port multiplexed with DTC event counter input	PB7/EVENT15 PB6/EVENT14 PB5/EVENT13 PB4/EVENT12 PB3/EVENT11 PB2/EVENT10 PB1/EVENT9 PB0/EVENT8	Same as left	
Port C	General I/O port multiplexed with PWMX output and IIC_2, IIC_3, and IIC_4 I/O	PC7/PWX3 PC6/PWX2	Same as left	
		PC5/SDA4 PC4/SCL4 PC3/SDA3 PC2/SCL3 PC1/SDA2 PC0/SCL2	Same as left	NM pus outp

Port E	General I/O port multiplexed with LPC I/O	PE7/SERIRQ PE6/LCLK PE5/$\overline{\text{LRESET}}$ PE4/$\overline{\text{LFRAME}}$ PE3/LAD3 PE2/LAD2 PE1/LAD1 PE0/LAD0	Same as left
Port F	General I/O port	PF3/RS11 PF2/RS10 PF1/RS9 PF0/RS8	Same as left

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

8.1.1 Port 1 Data Direction Register (P1DDR)

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

Bit	Bit Name	Initial Value	R/W	Description
7	P17DDR	0	W	• Normal extended mode (ADMXE = 0)
6	P16DDR	0	W	When set to 1, the corresponding pins fur
5	P15DDR	0	W	address output pins; when cleared to 0, fu
4	P14DDR	0	W	as input port pins.
3	P13DDR	0	W	• Address-data multiplex extended mode (A
2	P12DDR	0	W	1)
1	P11DDR	0	W	These bits correspond to the AD7 to AD0
0	P10DDR	0	W	the address-data multiplex bus.
				• Single-chip mode
				When set to 1, the corresponding pins fur
				output port pins; when cleared to 0, functi
				input port pins.

2	P12DR	0	R/W
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 the input state, the corresponding input pull-up MOS is turned a P1PCR bit is set to 1. Do not change the initial value when using address-data multiplex extended bus mod
6	P16PCR	0	R/W	
5	P15PCR	0	R/W	
4	P14PCR	0	R/W	
3	P13PCR	0	R/W	
2	P12PCR	0	R/W	
1	P11PCR	0	R/W	
0	P10PCR	0	R/W	

ABW256		(8/16-bit bus)	are 1 (8-bit bus)		(8/16-bit bus)	ar bu
Pin function	P1n input pin	ADn input/output pin	P1n input pin	An output pin	Setting prohibited	P1 pi

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

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

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

P1nDDR	0	1
Pin function	P1n input pin	P1n output pin

[Legend] n = 7 to 0

8.1.5 Port 1 Input Pull-Up MOS

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

Table 8.2 Port 1 Input Pull-Up MOS States

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

[Legend]

Off: Always off.

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

8.2.1 Port 2 Data Direction Register (P2DDR)

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	• Normal extended mode (ADMXE = 0) When set to 1, the corresponding pin as address output pins; when cleared function as input port pins. The address output pins used are in with the settings of the IOSE and CS of SYSCR. • Address-data multiplex extended mo (ADMXE = 1) These bits correspond to the AD15 to of the address-data multiplex bus. • Single-chip mode When set to 1, the corresponding pin as output port pins; when cleared to as input port pins.
6	P26DDR	0	W	
5	P25DDR	0	W	
4	P24DDR	0	W	
3	P23DDR	0	W	
2	P22DDR	0	W	
1	P21DDR	0	W	
0	P20DDR	0	W	

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

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 the input state, the corresponding input pull-up MOS is turned when a P2PCR bit is set to 1. Do not change the initial value when using address-data multiplex extended mos mod
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	

P2nDDR	0		1		
ADMXE	0	1	0		
Address 11	X	X	0	1	
Pin function	P2n input pin	ADm input/output pin	Am output pin	P2n output pin	A input/o

[Legend] m = 15 to 11, n = 7 to 3, X: Don't care.

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

[Legend] m = 10 to 8, n = 2 to 0

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

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

P2nDDR	0	1
Pin function	P2n input pin	P2n output pin

[Legend] n = 7 to 0

Off: Always off.

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

8.3.1 Port 3 Data Direction Register (P3DDR)

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

Bit	Bit Name	Initial Value	R/W	Description
7	P37DDR	0	W	• Normal extended mode (ADMXE = 0)
6	P36DDR	0	W	The pins function as bidirectional data bus
5	P35DDR	0	W	• Other modes
4	P34DDR	0	W	When set to 1, the corresponding pins fun
3	P33DDR	0	W	output port pins; when cleared to 0, functi
2	P32DDR	0	W	port pins.
1	P31DDR	0	W	
0	P30DDR	0	W	

2	P32DR	0	R/W	For the bits with the corresponding P3DDR cleared to 0, 1 is read.
1	P31DR	0	R/W	• Other modes
0	P30DR	0	R/W	P3DR stores output data for the port 3 pins used as the general output port. If this register is read, the P3DR values are the bits with the corresponding P3DDR bits For the bits with the corresponding P3DDR cleared to 0, the pin states are read.

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

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

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

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

P3nDDR	0	1
Pin function	P3n input pin	P3n output pin

[Legend] n = 7 to 0

8.3.5 Port 3 Input Pull-Up MOS

Port 3 has built-in input pull-up MOSs that can be controlled by software. The input pul
can be used in single-chip mode and address-data multiplex extended mode. Table 8.4 s
the input pull-up MOS states.

Table 8.4 Port 3 Input Pull-Up MOS States

Mode	Reset	Hardware Standby Mode	Software Standby Mode	In O Ope
Normal extended mode (EXPE = 1, ADMXE = 0)	Off	Off	Off	Off
Single-chip mode (EXPE = 0)	Off	Off	On/Off	On/
Address-data multiplex extended mode (EXPE = 1, ADMXE = 1)				

[Legend]

Off: Always off.

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

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

Bit	Bit Name	Initial Value	R/W	Description
7	P47DDR	0	W	The corresponding pins function as output port
6	P46DDR	0	W	the P4DDR bits are set to 1, and as input port
5	P45DDR	0	W	cleared to 0.
4	P44DDR	0	W	
3	P43DDR	0	W	
2	P42DDR	0	W	
1	P41DDR	0	W	
0	P40DDR	0	W	

3	P43DR	0	R/W	the pin states are read.
2	P42DR	0	R/W	
1	P41DR	0	R/W	
0	P40DR	0	R/W	

8.4.3 Port 4 Pull-Up MOS Control Register (P4PCR)

P4PCR controls the port 4 built-in input pull-up MOSs.

Bit	Bit Name	Initial Value	R/W	Description
7	P47PCR	0	R/W	When the pins are in the input state, the corre
6	P46PCR	0	R/W	input pull-up MOS is turned on when a P4PC
5	P45PCR	0	R/W	to 1.
4	P44PCR	0	R/W	
3	P43PCR	0	R/W	
2	P42PCR	0	R/W	
1	P41PCR	0	R/W	
0	P40PCR	0	R/W	

	$\overline{\text{IRQn}}$ input pin	

[Legend] n = 7 to 0

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

Bit	Bit Name	Initial Value	R/W	Description
7	P57DDR	0	W	If port 5 pins are specified for use as the ge port, the corresponding pins function as out when the P5DDR bits are set to 1, and as in when cleared to 0.
6	P56DDR	0	W	
5	P55DDR	0	W	
4	P54DDR	0	W	
3	P53DDR	0	W	
2	P52DDR	0	W	
1	P51DDR	0	W	
0	P50DDR	0	W	

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

8.5.3 Pin Functions

Port 5 pins can operate as the PWMX output, SCI_1, SCI_3, and SCIF input/output, or ge
port pins. The relationship between register setting values and pin functions are as follow

- P57/$\overline{\text{IRQ15}}$/PWX1

  The pin function is switched as shown below according to the combination of the OE
  DACR of PWMX and the P57DDR bit.

  When the ISS15 bit in ISSR16 is cleared to 0 and the IRQ15E bit in IER16 of the inte
  controller is set to 1, this pin can be used as the $\overline{\text{IRQ15}}$ input pin. To use this pin as th
  input pin, clear the P57DDR bit to 0.

OEB	0		1
P57DDR	0	1	X
Pin function	P57 input pin	P57 output pin	PWX1 outp
	$\overline{\text{IRQ15}}$ input pin		

[Legend] X: Don't care.

	IRQ14 input pin		

[Legend] X: Don't care.

- P55/$\overline{\text{IRQ13}}$/RxD3

  The pin function is switched as shown below according to the combination of the RE SCR and the SMIF bit in SCMR of SCI_3, 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	
SMIF	0			
P55DDR	0	1	X	
Pin function	P55 input pin	P55 output pin	RxD3 input pin	RxD3 inp
	$\overline{\text{IRQ13}}$ input pin			p

[Legend] X: Don't care.

Pin function	P54 input pin	P54 output pin	TxD3 output
	$\overline{\text{IRQ12}}$ input pin		

[Legend] X: Don't care.

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

  The pin function is switched as shown below according to the combination of the RE SCR and the SMIF bit in SCMR 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 as the $\overline{\text{IRQ11}}$ pin, clear the P53DDR bit to 0.

RE	0		1	
SMIF	0			1
P53DDR	0	1	X	
Pin function	P53 input pin	P53 output pin	RxD1 input pin	RxD3 inpu pi
	$\overline{\text{IRQ11}}$ input pin			

[Legend] X: Don't care.

Pin function	P52 input pin	P52 output pin	TxD1 output
	$\overline{\text{IRQ10}}$ input pin		

[Legend] X: Don't care.

- P51/$\overline{\text{IRQ9}}$/RxDF

  The pin function is switched as shown below according to the combination of the enable/disable setting of the SCIF and the P51DDR bit.

  When the ISS9 bit in ISSR16 is cleared to 0 and the IRQ9E bit in IER16 of the inter controller is set to 1, this pin can be used as the $\overline{\text{IRQ9}}$ input pin. To use as the $\overline{\text{IRQ9}}$ i clear the P51DDR bit to 0.

SCIF	Disabled		Enabled
P51DDR	0	1	X
Pin function	P51 input pin	P51 output pin	RxDF input
	$\overline{\text{IRQ9}}$ input pin		

[Legend] X: Don't care.

	IRQ8 input pin		

[Legend] X: Don't care.

- Noise canceler enable register (P6NCE)
- Noise canceler mode control register (P6NCMC)
- Noise cancel cycle setting register (NCCS)

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 bus) These bits have no effect on operation. • Other modes If port 6 pins are specified for use as the I/O port, the corresponding pins function port when the P6DDR bits are set to 1, a input port when cleared to 0.
6	P66DDR	0	W	
5	P65DDR	0	W	
4	P64DDR	0	W	
3	P63DDR	0	W	
2	P62DDR	0	W	
1	P61DDR	0	W	
0	P60DDR	0	W	

2	P62DR	0	R/W	to 1. For the bits with the corresponding P bits cleared to 0, 1 is read. • Other modes These bits store output data for the port 6 that are used as the general output port. If this register is read, the P6DR values a for the bits with the corresponding P6DDI to 1. For the bits with the corresponding P bits cleared to 0, the pin states are read.
1	P61DR	0	R/W	
0	P60DR	0	R/W	

8.6.3 Port 6 Pull-Up MOS Control Register (P6PCR)

P6PCR controls the port 6 built-in input pull-up MOSs.

Bit	Bit Name	Initial Value	R/W	Description
7	P67PCR	0	R/W	• Normal extended mode (16-bit bus) This register has no effect on operatio • Other modes When the pins are in the input state, th corresponding input pull-up MOS is tu when a P6PCR bit is set to 1.
6	P66PCR	0	R/W	
5	P65PCR	0	R/W	
4	P64PCR	0	R/W	
3	P23PCR	0	R/W	
2	P62PCR	0	R/W	
1	P61PCR	0	R/W	
0	P60PCR	0	R/W	

2	P62NCE	0	R/W	
1	P61NCE	0	R/W	
0	P60NCE	0	R/W	

8.6.5 Noise Canceler Mode Control Register (P6NCMC)

P6NCMC controls whether 1 or 0 is expected for the input signal to port 6 in bit units.

Bit	Bit Name	Initial Value	R/W	Description
7	P67NCMC	0	R/W	1 expected: 1 is stored in the port data re while 1 is input stably.
6	P66NCMC	0	R/W	
5	P65NCMC	0	R/W	0 expected: 0 is stored in the port data re while 0 is input stably.
4	P64NCMC	0	R/W	
3	P63NCMC	0	R/W	
2	P62NCMC	0	R/W	
1	P61NCMC	0	R/W	
0	P60NCMC	0	R/W	

000: 0.06 µs	φ/2	100: 963.8 µs	φ/32
001: 0.94 µs	φ/32	101: 1.9 ms	φ/65
010: 15.1 µs	φ/512	110: 3.9 ms	φ/13
011: 240.9 µs	φ/8192	111: 7.7 ms	φ/26

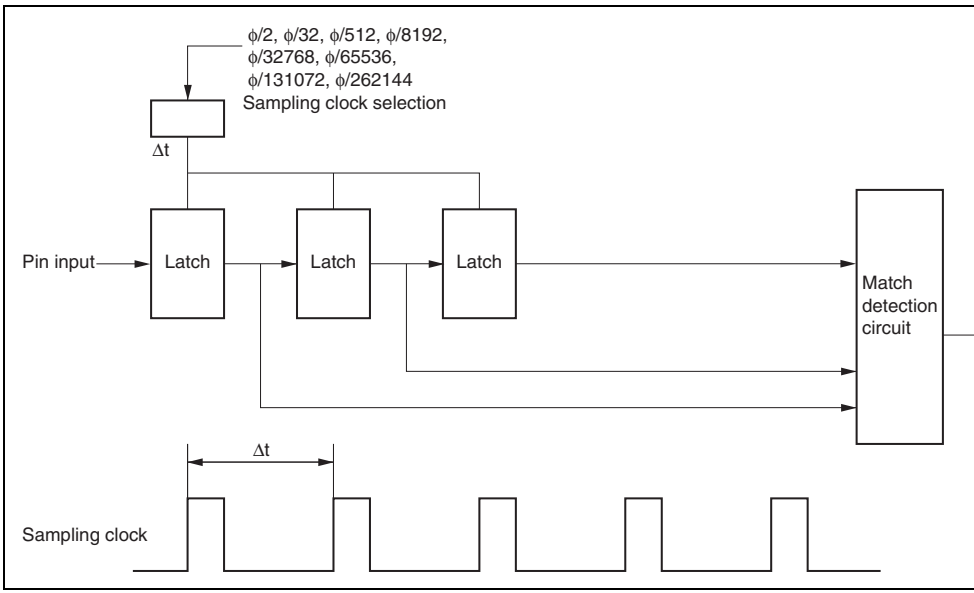


Figure 8.1 Noise Canceler Circuit

(n = 7 to 0)

Figure 8.2 Noise Canceler Operation

8.6.7 Pin Functions

(1) Normal Extended Mode

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

(2) Address-Data Multiplex Extended Mode

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

(3) Single-Chip Mode

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

P66DDR	0		1
P66NCE	0	1	X
Pin function	P66 input pin	DB14 input pin	P66 output pin

[Legend] X: Don't care.

- P65/DB13/$\overline{\text{RTS}}$

  The pin function is switched as shown below according to the combination of the enable/disable setting of the SCIF and the P65DDR bit and P65NCE bit.

SCIF	Disabled			Enab
P65DDR	0		1	X
P65NCE	0	1	X	X
Pin function	P65 input pin	DB13 input pin	P65 output pin	$\overline{\text{RTS}}$ out

[Legend] X: Don't care.

- P64/DB12/$\overline{\text{CTS}}$

  The pin function is switched as shown below according to the combination of the enable/disable setting of the SCIF and the P64DDR bit and P64NCE bit.

SCIF	Disabled			Enab
P64DDR	0		1	X
P64NCE	0	1	X	X
Pin function	P64 input pin	DB12 input pin	P64 output pin	$\overline{\text{CTS}}$ inp

[Legend] X: Don't care.

P62DDR	0		1
P62NCE	0	1	X
Pin function	P62 input pin	DB10 input pin	P62 output pin

[Legend] X: Don't care.

- P61/DB9

  The pin function is switched as shown below according to the P61DDR bit and P61N

P61DDR	0		1
P61NCE	0	1	X
Pin function	P61 input pin	DB9 input pin	P61 output pin

[Legend] X: Don't care.

- P60/DB8

  The pin function is switched as shown below according to the P60DDR bit and P60N

P60DDR	0		1
P60NCE	0	1	X
Pin function	P60 input pin	DB8 input pin	P60 output pin

[Legend] X: Don't care.

On/Off: On when P6DDR = 0 and P6PCR = 1; otherwise off.

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

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

Pin function	AN7 input pin	P77 input pin	$\overline{\text{ExIRQ7}}$ inpu

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

  The pin function is switched as shown below according to the combination of the SCA in ADCR and the CH2 to CH0 bits in ADCSR of the A/D converter, and the ISS6 bit of the interrupt controller. Do not set these bits to other values than those shown in the following table.

SCANE	0			1		
CH2 to CH0	B'110	Other than B'110		B'110 to B'111	B'000 to B'	
ISS6	0	0	1	0	0	
Pin function	AN6 input pin	P76 input pin	$\overline{\text{ExIRQ6}}$ input pin	AN6 input pin	P76 input pin	$\overline{\text{E}}$ in

Pin function	AN5 input pin	P75 input pin	ExIRQ5 input pin	AN5 input pin	P75 input pin	

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

  The pin function is switched as shown below according to the combination of the SC in ADCR and the CH2 to CH0 bits in ADCSR of the A/D converter, and the ISS4 bi of the interrupt controller. Do not set these bits to other values than those shown in th following table.

SCANE	0			1		
CH2 to CH0	B'100	Other than B'100		B'100 to B'111	B'000 to B	
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	

Pin function	AN3 input pin	P73 input pin	$\overline{\text{ExIRQ3}}$ input pin	AN3 input pin	P73 input pin	$\overline{\text{ExIRQ3}}$ input pin	AN3 input pin	P73 input pin

[Legend] X: Don't care.

- P72/$\overline{\text{ExIRQ2}}$/AN2

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

SCANE	0			1				
SCANS	X			0			1	
CH2 to CH0	B'010	Other than B'010		B'010 to B'011	Other than B'010 to B'011		B'010 to B'111	B'000
ISS2	0	0	1	0	0	1	0	0
Pin function	AN2 input pin	P72 input pin	$\overline{\text{ExIRQ2}}$ input pin	AN2 input pin	P72 input pin	$\overline{\text{ExIRQ2}}$ input pin	AN2 input pin	P72 input pin

[Legend] X: Don't care.

Pin function	AN1 input pin	P71 input pin	$\overline{\text{ExIRQ1}}$ input pin	AN1 input pin	P71 input pin	$\overline{\text{ExIRQ1}}$ input pin	AN1 input pin	P71 inp pin

[Legend] X: Don't care.

- P70/$\overline{\text{ExIRQ0}}$/AN0

  The pin function is switched as shown below according to the combination of the SC SCANS bits in ADCR and the CH2 to CH0 bits in ADCSR of the A/D converter, an bit in ISSR of the interrupt controller. Do not set these bits to other values than those the following table.

SCANE	0			1		
SCANS	X			0		
CH2 to CH0	B'000	Other than B'000		B'000 to B'011	Other than B'000 to B'011	
ISS0	0	0	1	0	0	1
Pin function	AN0 input pin	P70 input pin	$\overline{\text{ExIRQ0}}$ input pin	AN0 input pin	P70 input pin	$\overline{\text{ExIRQ0}}$ input pin

[Legend] X: Don't care.

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

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

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

8.8.3 Pin Functions

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

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

  The pin function is switched as shown below according to the P87DDR bit.

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

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

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

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

When the ISS13 bit in ISSR16 is set to 1, this pin can be used as the $\overline{\text{ExIRQ13}}$ input p
use this pin as the $\overline{\text{ExIRQ13}}$ input pin, clear the P85DDR bit to 0.

CKE1	0				
C/$\overline{A}$	0			1	
CKE0	0		1	X	
P85DDR	0	1	X	X	
Pin function	P85 input pin	P85 output pin	SCK1 output pin	SCK1 output pin	SCI
	$\overline{\text{ExIRQ13}}$ input pin				

[Legend] X: Don't care.

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

[Legend] X: Don't care.

- P83/SDA1

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

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

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

[Legend] X: Don't care.

[Legend] X: Don't care.

- P81/SDA0

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

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

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

[Legend] X: Don't care.

- P80/SCL0

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

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

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

[Legend] X: Don't care.

The individual bits of P9DDR specify input or output for the port 9 pins.

Bit	Bit Name	Initial Value	R/W	Description
7	P97DDR	0	W	[P97DDR, P95DDR to P90DDR]
6	P96DDR	0	W	If port 9 pins are specified for use as the ge
5	P95DDR	0	W	port, the corresponding pins function as out
4	P94DDR	0	W	when the P97DDR, and P95DDR to P90DD set to 1, and as input port when cleared to 0
3	P93DDR	0	W	[P96DDR]
2	P92DDR	0	W	The corresponding port 9 pin functions as th
1	P91DDR	0	W	clock output pin (ϕ) when this bit is set to 1, the general I/O port when cleared to 0.
0	P90DDR	0	W	

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

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

8.9.3 Pin Functions

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

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

  The pin function is switched as shown below according to the operating mode and the combination of the CS256E bit in SYSCR, the WMS1 bit in WSCR, the WMS21 bit i WSCR2, and the P97DDR bit.

Operating mode	Extended mode				Single-chip
WMS1, WMS21	All 0			Either bit is 1	X
CS256E	0		1	X	X
P97DDR	0	1	X	X	0
Pin function	P97 input pin	P97 output pin	$\overline{\text{CS256}}$ output pin	$\overline{\text{WAIT}}$ input pin	P97 input pin

[Legend] X: Don't care.

The pin function is switched as shown below according to the operating mode and th
combination of the IOSE bit in SYSCR and the P95DDR bit.

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

[Legend] X: Don't care.

- P94/$\overline{HWR}$

  The pin function is switched as shown below according to the operating mode and th
  P94DDR bit.

Operating mode	Extended mode	Single-chip mode	
P94DDR	X	0	
Pin function	$\overline{HWR}$ output pin	P94 input pin	P94 c

[Legend] X: Don't care.

- P92/$\overline{HBE}$

  The pin function is switched as shown below according to the operating mode, the OB
  PTCNT0, and the P92DDR bit.

Operating mode	Extended mode			Single-chip mo	
OBE	0		1	X	
P92DDR	0	1	X	0	
Pin function	P92 input pin	P92 output pin	$\overline{HBE}$ output pin	P92 input pin	P92 o

[Legend] X: Don't care.

- P91/$\overline{AH}$

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

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

[Legend] X: Don't care.

	pin	pin	pin	pin		

[Legend] X: Don't care.

8.10.1 Port A Data Direction Register (PADDR)

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

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

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

8.10.3 Port A Input Data Register (PAPIN)

PAPIN indicates the states of the port A pins.

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

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

expressions according to the control bits of the bus controller or other module.

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

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

- PA7/EVENT7/A23, PA6/EVENT6/A22, PA5/EVENT5/A21, PA4/EVENT4/A20, PA3/EVENT3/A19, PA2/EVENT2/A18

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

  When using the pin as an EVENT input pin, clear the PAnDDR bit to 0. Though the s for the EVENT input pin have been made, set the PAnDDR bit to 1 when using the pi PAn or Am output pin.

PAnDDR	0	1	1
Address 18	1		
Pin function	PAn input pin	PAn output pin	Am outp
	EVENTn input pin		

[Legend] n = 7 to 2, m = 23 to 18

	EVENTn input pin		

[Legend] n = 1, 0; m = 17, 16

(2) Single-Chip Mode and Address-Data Multiplex Extended Mode

Port A pins can function as the event counter input pins.

- PA7/EVENT7, PA6/EVENT6, PA5/EVENT5, PA4/EVENT4, PA3/EVENT3, PA2/
  PA1/EVENT1, PA0/EVENT0

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

  Though the settings for the EVENT input pin have been made, set the PAnDDR bit t
  the pin as the PAn output pin.

PAnDDR	0	1
Pin function	PAn input pin	PAn output pin
	EVENTn input pin	

[Legend] n = 7 to 0

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 8.6 summarizes the input pull-up MOS states.

Table 8.6 Input Pull-Up MOS States

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

[Legend]

Off: Always off.

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

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

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

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

8.11.3 Port B Input Data Register (PBPIN)

PBPIN indicates the states of the port B pins.

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

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

X: Don't care.

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

8.12.1 Port C Data Direction Register (PCDDR)

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

Bit	Bit Name	Initial Value	R/W	Description
7	PC7DDR	0	W	When set to 1, the corresponding pins functio output port pins; when cleared to 0, function port pins. Since this register is allocated to the same ad PCPIN, states of the port C pins are returned this register is read.
6	PC6DDR	0	W	
5	PC5DDR	0	W	
4	PC4DDR	0	W	
3	PC3DDR	0	W	
2	PC2DDR	0	W	
1	PC1DDR	0	W	
0	PC0DDR	0	W	

2	PC2ODR	0	R/W	
1	PC1ODR	0	R/W	
0	PC0ODR	0	R/W	

8.12.3 Port C Input Data Register (PCPIN)

PCPIN indicates the pin states of port C.

Bit	Bit Name	Initial Value	R/W	Description
7	PC7PIN	Undefined*	R	When this register is read, the pin states are
6	PC6PIN	Undefined*	R	Since this register is allocated to the same a
5	PC5PIN	Undefined*	R	PCDDR, writing to this register writes data t
4	PC4PIN	Undefined*	R	and the port C setting is changed.
3	PC3PIN	Undefined*	R	
2	PC2PIN	Undefined*	R	
1	PC1PIN	Undefined*	R	
0	PC0PIN	Undefined*	R	

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

Pin function	PC7 input pin	PC7 output pin	PWX3 outp

[Legend] X: Don't care.

- PC6/PWX2

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

OEA	0		1
PC6DDR	0	1	X
Pin function	PC6 input pin	PC6 output pin	PWX2 outp

[Legend] X: Don't care.

- PC5/SDA4

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

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

[Legend] X: Don't care.

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

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

[Legend] X: Don't care.

- PC2/SCL3

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

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

[Legend] X: Don't care.

- PC1/SDA2

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

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

[Legend] X: Don't care.

8.13.1 Port D Data Direction Register (PDDDR)

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

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

2	PD2ODR	0	R/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	When this register is read, the pin states are
6	PD6PIN	Undefined*	R	Since this register is allocated to the same ad
5	PD5PIN	Undefined*	R	PDDDR, writing to this register writes data to and the port D setting is changed.
4	PD4PIN	Undefined*	R	
3	PD3PIN	Undefined*	R	
2	PD2PIN	Undefined*	R	
1	PD1PIN	Undefined*	R	
0	PD0PIN	Undefined*	R	

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

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

[Legend] X: Don't care.

- PD6/SCL5

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

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

[Legend] X: Don't care.

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

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

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

in HICR0 of the LPC and the PD3DDR bit.

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

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

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

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

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

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

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

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

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

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

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

Table 8.7 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 Operat
Off	Off	On/Off	On/Off

[Legend]

Off: Always off.

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

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

Bit	Bit Name	Initial Value	R/W	Description
7	PE7DDR	0	W	When set to 1, the corresponding pins functio output port pins; when cleared to 0, function port pins. Since this register is allocated to the same ad PEPIN, states of the port E pins are returned this register is read.
6	PE6DDR	0	W	
5	PE5DDR	0	W	
4	PE4DDR	0	W	
3	PE3DDR	0	W	
2	PE2DDR	0	W	
1	PE1DDR	0	W	
0	PE0DDR	0	W	

2	PE2ODR	0	R/W	
1	PE1ODR	0	R/W	
0	PE0ODR	0	R/W	

8.14.3 Port E Input Data Register (PEPIN)

PEPIN indicates the pin states of port E.

Bit	Bit Name	Initial Value	R/W	Description
7	PE7PIN	Undefined*	R	When this register is read, the pin states are
6	PE6PIN	Undefined*	R	Since this register is allocated to the same a
5	PE5PIN	Undefined*	R	PEDDR, writing to this register writes data t
4	PE4PIN	Undefined*	R	and the port E setting is changed.
3	PE3PIN	Undefined*	R	
2	PE2PIN	Undefined*	R	
1	PE1PIN	Undefined*	R	
0	PE0PIN	Undefined*	R	

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

PE7DDR	0	1	X
Pin function	PE7 input pin	PE7 output pin	SERIRQ input/o

[Legend] X: Don't care.

- PE6/LCLK

  The pin function is switched as shown below according to whether the LPC is enable disabled and the PE6DDR bit.

LPC	Disabled		Enable
PE6DDR	0	1	X
Pin function	PE6 input pin	PE6 output pin	LCLK input

[Legend] X: Don't care.

- PE5/$\overline{\text{LRESET}}$

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

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

[Legend] X: Don't care.

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

LPC	Disabled		Enable
PE3DDR	0	1	X
Pin function	PE3 input pin	PE3 output pin	LAD3 input/ou

[Legend] X: Don't care.

- PE2/LAD2

  The pin function is switched as shown below according to whether the LPC is enable disabled and the PE2DDR bit.

LPC	Disabled		Enable
PE2DDR	0	1	X
Pin function	PE2 input pin	PE2 output pin	LAD2 input/ou

[Legend] X: Don't care.

- PE1/LAD1

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

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

[Legend] X: Don't care.

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

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

8.15.2 Port F Output Data Register (PFODR)

PFODR stores output data for the port F pins. PFODR is initialized only by a system res
retains the value even if an internal reset signal of the WDT is generated.

Bit	Bit Name	Initial Value	R/W	Description
7 to 4	—	—	—	Reserved Undefined value is read from this bit.
3	PF3ODR	0	R/W	Store the output data for the pins that are us
2	PF2ODR	0	R/W	general output port.
1	PF1ODR	0	R/W	
0	PF0ODR	0	R/W	

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

8.15.4 Pin Functions

Port F is a 4-bit I/O port. The relationship between the register settings and the pin functi
shown below.

- PF3/RS11, PF2/RS10, PF1/RS9, PF0/RS8
  The pin function is switched as shown below according to the PFnDDR bit.

PFnDDR	0	1
Pin function	PFn input pin	PFn output pin

[Legend] n = 3 to 0

8.10.1 IRQ Sense Port Select Register 16 (ISSR16), IRQ Sense Port Select Regis

ISSR16 and ISSR select pins for the $\overline{\text{IRQ15}}$ to $\overline{\text{IRQ0}}$ inputs.

- ISSR16

Bit	Bit Name	Initial Value	R/W	Description
15	ISS15	0	R/W	0: 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: P74/$\overline{\text{ExIRQ4}}$ is selected
3	ISS3	0	R/W	0: P43/$\overline{\text{IRQ3}}$ 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	0: P41/$\overline{\text{IRQ1}}$ is selected
				1: P71/$\overline{\text{ExIRQ1}}$ is selected
0	ISS0	0	R/W	0: P40/$\overline{\text{IRQ0}}$ is selected
				1: P70/$\overline{\text{ExIRQ0}}$ is selected

				with the SCI_3 as the smart card interfac 0: TxD3 and RxD3 are not internally con 1: TxD3 and RxD3 are internally connect
5 to 2	—	All 0	R/W	Reserved The initial value should not be changed.
1	OBE	0	R/W	Selects glueless extension. 0: Control by $\overline{RD}$, $\overline{HWR}$, $\overline{LWR}$ 1: Control by $\overline{RD}$, $\overline{WR}$, $\overline{HBE}$, $\overline{LBE}$ (glueless extension)
0	—	0	R/W	Reserved The initial value should not be changed.

- Two base cycle settings
  The base cycle can be set equal to T × 64 or T × 256, where T is the resolution.
- Sixteen operation clocks (by combination of eight resolution settings and two base c settings)

Figure 9.1 shows a block diagram of the PWM (D/A) module.

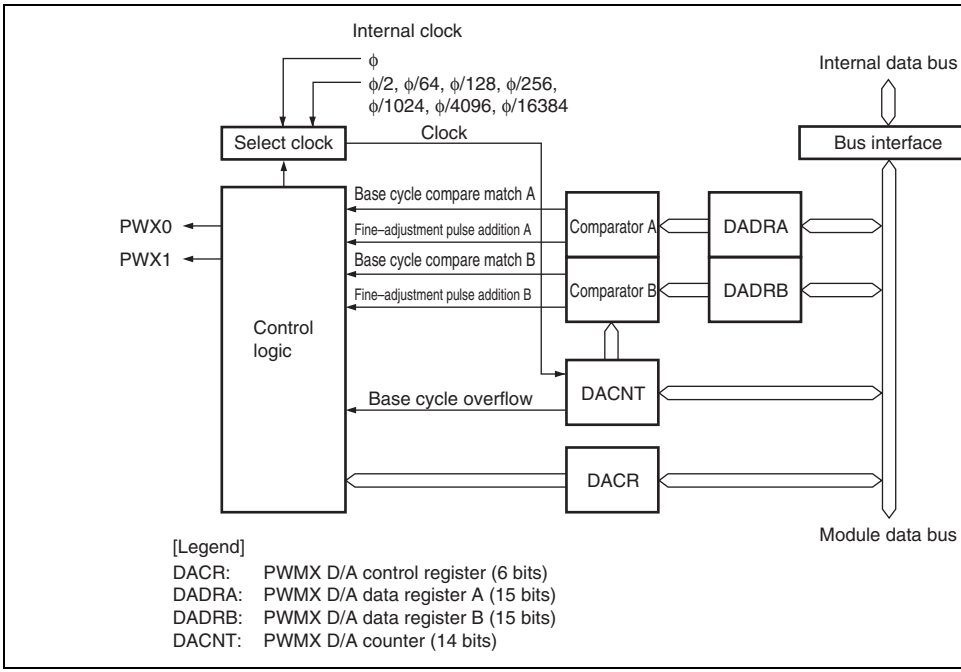


Figure 9.1 PWMX (D/A) Block Diagram

9.3 Register Descriptions

The PWMX (D/A) module has the following registers. For details on the module stop co register, see section 24.1.3, Module Stop Control Registers H, L, and A (MSTPCRH, MS MSTPCRA).

- PWMX (D/A) counter (DACNT)
- PWMX (D/A) data register A (DADRA)
- PWMX (D/A) data register B (DADRB)
- PWMX (D/A) control register (DACR)
- Peripheral clock select register (PCSR)

Note: The same addresses are shared by DADRA and DACR, and by DADRB and DA Switching is performed by the REGS bit in DACNT or DADRB.

Bit	Bit Name	Initial Value	R/W	Description
15 to 8	UC7 to UC0	All 0	R/W	Lower Up-Counter
7 to 2	UC8 to UC13	All 0	R/W	Upper Up-Counter
1	—	1	R	Reserved This bit is always read as 1 and cannot be mo
0	REGS	1	R/W	Register Select DADRA and DACR, and DADRB and DACNT located at the same addresses. The REGS bit which registers can be accessed. When chang register to be accessed, set this bit in advance 0: DADRA and DADRB can be accessed 1: DACR and DACNT can be accessed

				These bits set a digital value to be converted to analog value. In each base cycle, the DACNT value is continu compared with the DADR value to determine th cycle of the output waveform, and to decide wh output a fine-adjustment pulse equal in width to resolution. To enable this operation, this registe set within a range that depends on the CFS bit. DADR value is outside this range, the PWM out held constant. A channel can be operated with 12-bit precision DA0 and DA1 to 0. The two data bits are not co with UC12 and UC13 of DACNT.
1	CFS	1	R/W	Carrier Frequency Select 0: Base cycle = resolution (T) × 64 The range of DA13 to DA0: H'0100 to H'3FFF 1: Base cycle = resolution (T) × 256 The range of DA13 to DA0: H'0040 to H'3FFF
0	—	1	R	Reserved This bit is always read as 1 and cannot be modi

				be set within a range that depends on the CF DADR value is outside this range, the PWM c held constant. A channel can be operated with 12-bit precisi fixing DA0 and DA1 to 0. The two data bits ar compared with UC12 and UC13 of DACNT.
1	CFS	1	R/W	Carrier Frequency Select 0: Base cycle = resolution (T) × 64 DA13 to DA0 range = H'0100 to H'3FFF 1: Base cycle = resolution (T) × 256 DA13 to DA0 range = H'0040 to H'3FFF
0	REGS	1	R/W	Register Select DADRA and DACR, and DADRB and DACNT located at the same addresses. The REGS bi which registers can be accessed. When char register to be accessed, set this bit in advanc 0: DADRA and DADRB can be accessed 1: DACR and DACNT can be accessed

				0: DACNT operates as a 14-bit up-counter 1: DACNT halts at H'0003
5, 4	—	All 1	R	Reserved These bits are always read as 1 and cannot be m
3	OEB	0	R/W	Output Enable B Enables or disables output on PWMX (D/A) chan 0: PWMX (D/A) channel B output (at the PWX1, P pins) is disabled 1: PWMX (D/A) channel B output (at the PWX1, P pins) is enabled
2	OEA	0	R/W	Output Enable A Enables or disables output on PWMX (D/A) chan 0: PWMX (D/A) channel A output (at the PWX0, P pin) is disabled 1: PWMX (D/A) channel A output (at the PWX0, P pins) is enabled
1	OS	0	R/W	Output Select Selects the phase of the PWMX (D/A) output. 0: Direct PWMX (D/A) output 1: Inverted PWMX (D/A) output
0	CKS	0	R/W	Clock Select Selects the PWMX (D/A) resolution. Eight kinds o resolution can be selected. 0: Operates at resolution (T) = system clock cycle (t_{cyc}) 1: Operates at resolution (T) = system clock cycle (t_{cyc}) × 2, × 64, × 128, × 256, × 1024, × 4096, an 16384.

4	PWCKX0A	0	R/W	These bits select a clock cycle with the CKS bit of PWMX_0 being 1. See table 9.2.
3	PWCKX1C	0	R/W	PWMX_1 Clock Select This bit selects a clock cycle with the CKS bit of PWMX_1 being 1. See table 9.2.
2	—	0	R/W	Reserved
1	—	0	R/W	The initial value should not be changed.
0	PWCKX0C	0	R/W	PWMX_0 Clock Select This bit selects a clock cycle with the CKS bit of PWMX_0 being 1. See table 9.2.

Table 9.2 Clock Select of PWMX_1 and PWMX_0

PWCKX0C PWCKX1C	PWCKX0B PWCKX1B	PWCKX0A PWCKX1A	Resolution (T)
0	0	0	Operates on the system clock cycle (t_{cyc})
0	0	1	Operates on the system clock cycle (t_{cyc})
0	1	0	Operates on the system clock cycle (t_{cyc})
0	1	1	Operates on the system clock cycle (t_{cyc})
1	0	0	Operates on the system clock cycle (t_{cyc})
1	0	1	Operates on the system clock cycle (t_{cyc})
1	1	0	Operates on the system clock cycle (t_{cyc})
1	1	1	Setting prohibited

combined 16-bit value is written in the register.

- Read

  When the upper byte is read from, the upper-byte value is transferred to the CPU and lower-byte value is transferred to TEMP. Next, when the lower byte is read from, the byte value in TEMP is transferred to the CPU.

These registers should always be accessed 16 bits at a time with a MOV instruction, and t byte should always be accessed before the lower byte. Correct data will not be transferred the upper byte or only the lower byte is accessed. Also note that a bit manipulation instru cannot be used to access these registers.

Example 1: Write to DACNT

```
MOV.W R0, @DACNT      ; Write R0 contents to DACNT
```

Example 2: Read DADRA

```
MOV.W @DADRA, R0      ; Copy contents of DADRA to R0
```

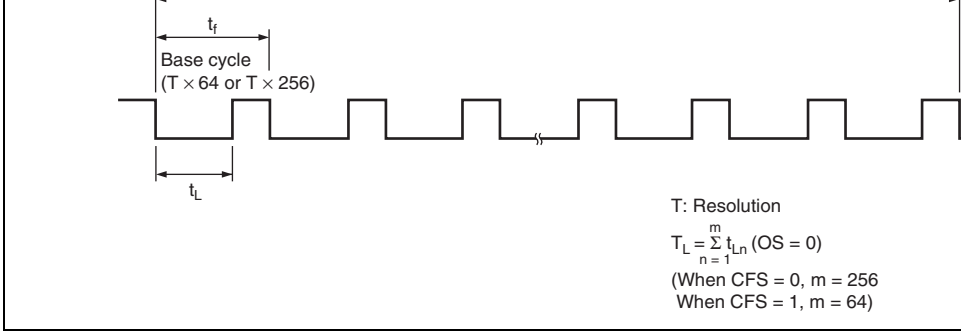


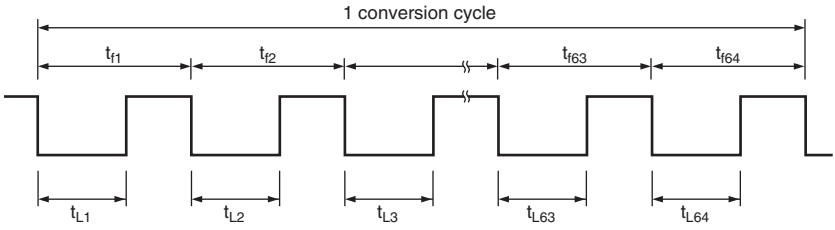
Figure 9.2 PWMX (D/A) Operation

Table 9.3 summarizes the relationships between the CKS and CFS bit settings and the re
base cycle, and conversion cycle. The PWM output remains fixed unless DA13 to DA0
contain at least a certain minimum value. The relationship between the OS bit and the ou
waveform is shown in figures 9.3 and 9.4.

						132.8 kHz		DA13 to 0 = H'0000 to H'003F	12			0	0	
								(Data value) × T						
								DA13 to 0 = H'0040 to H'3FFF	10	0	0	0	0	
0	0	0	1	0.06	0	3.76 µs	0.964 ms	Always low/high output	14					
				(ϕ/2)		265.6 kHz		DA13 to 0 = H'0000 to H'00FF	12			0	0	
								(Data value) × T						
								DA13 to 0 = H'0100 to H'3FFF	10	0	0	0	0	
					1	15.06 µs	0.964 ms	Always low/high output	14					
						66.4 kHz		DA13 to 0 = H'0000 to H'003F	12			0	0	
								(Data value) × T						
								DA13 to 0 = H'0040 to H'3FFF	10	0	0	0	0	
0	0	1	1	1.88	0	120.5 µs	30.840 ms	Always low/high output	14					
				(ϕ/64)		8.3 kHz		DA13 to 0 = H'0000 to H'00FF	12			0	0	
								(Data value) × T						
								DA13 to 0 = H'0100 to H'3FFF	10	0	0	0	0	
					1	481.9 µs	30.840 ms	Always low/high output	14					
						2.1 kHz		DA13 to 0 = H'0000 to H'003F	12			0	0	
								(Data value) × T						
								DA13 to 0 = H'0040 to H'3FFF	10	0	0	0	0	
0	1	0	1	3.76	0	240.9 µs	61.681 ms	Always low/high output	14					
				(ϕ/128)		4.2 kHz		DA13 to 0 = H'0000 to H'00FF	12			0	0	
								(Data value) × T						
								DA13 to 0 = H'0100 to H'3FFF	10	0	0	0	0	
					1	963.8 µs	61.681 ms	Always low/high output	14					
						1.0 kHz		DA13 to 0 = H'0000 to H'003F	12			0	0	
								(Data value) × T						
								DA13 to 0 = H'0040 to H'3FFF	10	0	0	0	0	

								DA13 to 0 = H'0040 to H'3FFF	10	0	0	0	0
1	0	0	1	30.12 (ϕ/1024)	0	1.93 ms 518.8 Hz	493.45 ms	Always low/high output DA13 to 0 = H'0000 to H'00FF (Data value) × T DA13 to 0 = H'0100 to H'3FFF	14				
									12			0	0
									10	0	0	0	0
					1	7.71 ms 129.7 Hz	493.45 ms	Always low/high output DA13 to 0 = H'0000 to H'003F (Data value) × T DA13 to 0 = H'0040 to H'3FFF	14				
									12			0	0
									10	0	0	0	0
1	0	1	1	120.47 (ϕ/4096)	0	7.71 ms 129.7 Hz	1.974 s	Always low/high output DA13 to 0 = H'0000 to H'00FF (Data value) × T DA13 to 0 = H'0100 to H'3FFF	14				
									12			0	0
									10	0	0	0	0
					1	30.84 ms 32.4 Hz	1.974 s	Always low/high output DA13 to 0 = H'0000 to H'003F (Data value) × T DA13 to 0 = H'0040 to H'3FFF	14				
									12			0	0
									10	0	0	0	0
1	1	0	1	481.88 (ϕ/16384)	0	30.84 ms 32.4 Hz	7.895 s	Always low/high output DA13 to 0 = H'0000 to H'00FF (Data value) × T DA13 to 0 = H'0100 to H'3FFF	14				
									12			0	0
									10	0	0	0	0
					1	123.36 ms 8.1 Hz	7.895 s	Always low/high output DA13 to 0 = H'0000 to H'003F (Data value) × T DA13 to 0 = H'0040 to H'3FFF	14				
									12			0	0
									10	0	0	0	0
1	1	1	1	Setting prohibited	—	—	—	—	—	—	—	—	—

Note: * Indicates the conversion cycle when specific DA3 to DA0 bits are fixed.



$t_{f1} = t_{f2} = t_{f3} = \cdots = t_{f63} = t_{f64} = T \times 256$
$t_{L1} + t_{L2} + t_{L3} + \cdots + t_{L63} + t_{L64} = T_L$

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

Figure 9.3 Output Waveform (OS = 0, DADR corresponds to T_L)

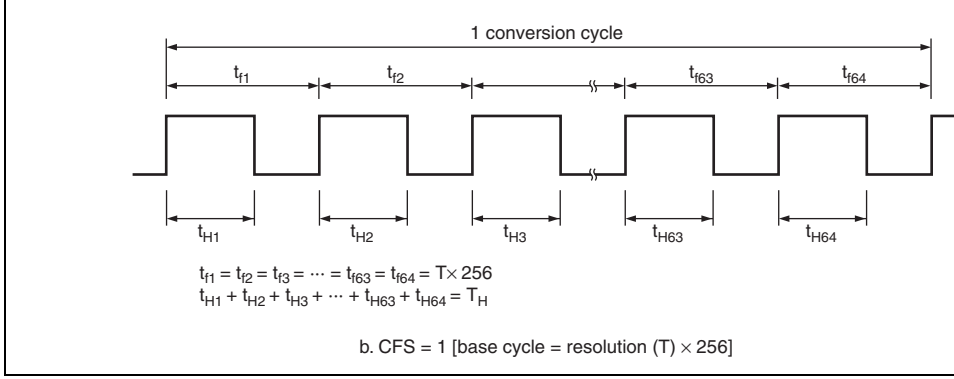


Figure 9.4 Output Waveform (OS = 1, DADR corresponds to T_H)

An example of the additional pulses when CFS = 1 (base cycle = resolution (T) × 256) a
(inverted PWM output) is described below. When CFS = 1, the upper eight bits (DA13 t
DADR determine the duty cycle of the base pulse while the subsequent six bits (DA5 to
determine the locations of the additional pulses as shown in figure 9.5.

Table 9.4 lists the locations of the additional pulses.

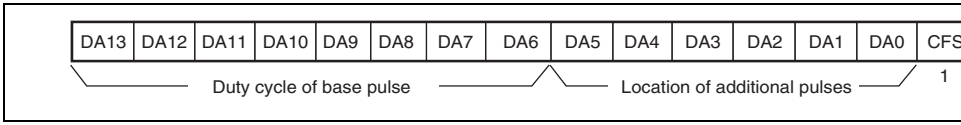


Figure 9.5 D/A Data Register Configuration when CFS = 1

In this example, DADR = H'0207 (B'0000 0010 0000 0111). The output waveform is sh
figure 9.6. Since CFS = 1 and the value of the upper eight bits is B'0000 0010, the high
the base pulse duty cycle is 2/256 × (T).

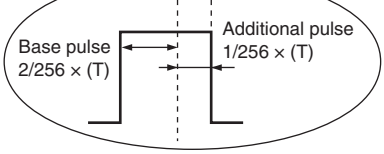


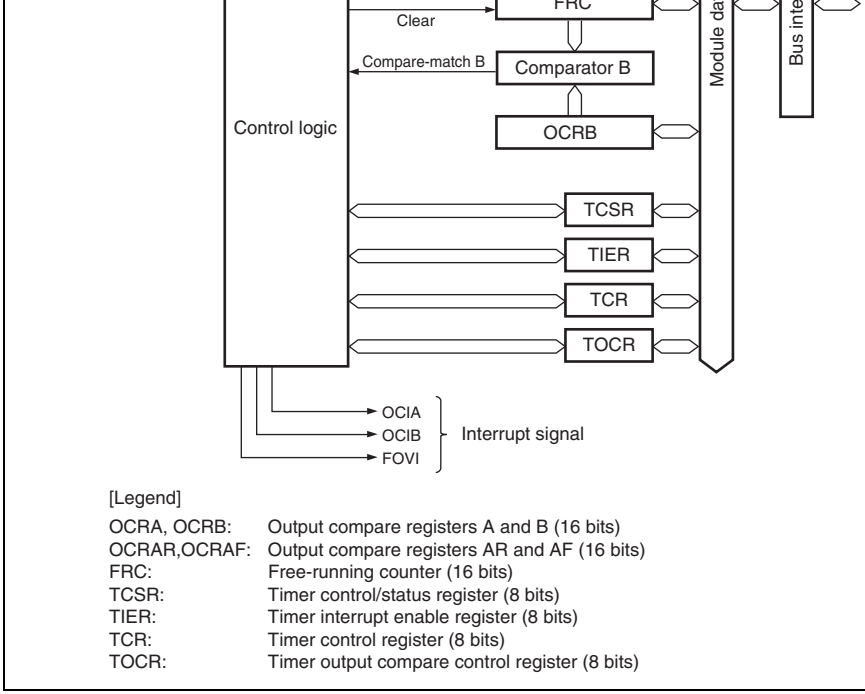
Figure 9.6 Output Waveform when DADR = H'0207 (OS = 1)

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

Lower 6 bits							Base pulse No.																																												
							0	1	2	3	4	5	6	7	8	9	10	11	12	13	14	15	16	17	18	19	20	21	22	23	24	25	26	27	28	29	30	31	32	33	34	35	36	37	38	39	40	41	42	43	44	45	46
0	0	0	0	0	0	0																																															
0	0	0	0	0	1	1																																															
0	0	0	0	1	0	2																																O															
0	0	0	0	1	1	3																																O															
0	0	0	1	0	0	4																O																O															
0	0	0	1	0	1	5																O																O															
0	0	0	1	1	0	6																O								O								O															
0	0	0	1	1	1	7																O								O								O								O							
0	0	1	0	0	0	8								O								O								O								O								O							
0	0	1	0	0	1	9								O								O								O								O								O							
0	0	1	0	1	0	10								O								O								O				O			O								O							
0	0	1	0	1	1	11								O								O								O				O			O								O				O			
0	0	1	1	0	0	12								O				O			O								O				O			O								O				O			
0	0	1	1	0	1	13								O				O			O								O				O			O								O				O			
0	0	1	1	1	0	14								O				O			O				O			O				O			O								O				O			
0	0	1	1	1	1	15								O				O			O				O			O				O			O				O			O				O			
0	1	0	0	0	0	16				O			O				O			O				O			O				O			O				O			O				O			
0	1	0	0	0	1	17				O			O				O			O				O			O				O			O				O			O				O			
0	1	0	0	1	0	18				O			O				O			O				O			O				O		O		O				O			O				O			
0	1	0	0	1	1	19				O			O				O			O				O			O				O		O		O				O			O				O		O	
0	1	0	1	0	0	20				O			O				O		O		O				O			O				O		O		O				O			O				O		O	
0	1	0	1	0	1	21				O			O				O		O		O				O			O				O		O		O				O			O				O		O	
0	1	0	1	1	0	22				O			O				O		O		O				O		O		O				O		O		O				O			O				O		O	
0	1	0	1	1	1	23				O			O				O		O		O				O		O		O				O		O		O				O		O		O				O		O	
0	1	1	0	0	0	24				O		O		O				O		O		O				O		O		O				O		O		O				O		O		O				O		O	
0	1	1	0	0	1	25				O		O		O				O		O		O				O		O		O				O		O		O				O		O		O				O		O	
0	1	1	0	1	0	26				O		O		O				O		O		O				O		O		O		O		O		O		O				O		O		O				O		O	
0	1	1	0	1	1	27				O		O		O				O		O		O				O		O		O		O		O		O		O				O		O		O		O		O		O	
0	1	1	1	0	0	28				O		O		O		O		O		O		O				O		O		O		O		O		O		O				O		O		O		O		O		O	
0	1	1	1	0	1	29				O		O		O		O		O		O		O				O		O		O		O		O		O		O				O		O		O		O		O		O	
0	1	1	1	1	0	30				O		O		O		O		O		O		O		O		O		O		O		O		O		O		O				O		O		O		O		O		O	
0	1	1	1	1	1	31				O		O		O		O		O		O		O		O		O		O		O		O		O		O		O		O		O		O		O		O		O		O	
1	0	0	0	0	0	32		O		O		O		O		O		O		O		O		O		O		O		O		O		O		O		O		O		O		O		O		O		O	
1	0	0	0	0	1	33		O		O		O		O		O		O		O		O		O		O		O		O		O		O		O		O		O		O		O		O		O		O	
1	0	0	0	1	0	34		O		O		O		O		O		O		O		O		O		O		O		O		O		O		O	O	O		O		O		O		O		O		O	
1	0	0	0	1	1	35		O		O		O		O		O		O		O		O		O		O		O		O		O		O		O	O	O		O		O		O		O		O		O	O
1	0	0	1	0	0	36		O		O		O		O		O		O		O	O	O		O		O		O		O		O		O	O	O		O		O		O		O		O		O	O
1	0	0	1	0	1	37		O		O		O		O		O		O		O	O	O		O		O		O		O		O		O	O	O		O		O		O		O		O		O	O
1	0	0	1	1	0	38		O		O		O		O		O		O		O	O	O		O		O		O	O	O		O		O		O	O	O		O		O		O		O		O		O	O
1	0	0	1	1	1	39		O		O		O		O		O		O		O	O	O		O		O		O	O	O		O		O		O	O	O		O		O		O	O	O		O		O	O
1	0	1	0	0	0	40		O		O		O	O	O		O		O		O	O	O		O		O		O	O	O		O		O		O	O	O		O		O		O	O	O		O		O	O
1	0	1	0	0	1	41		O		O		O	O	O		O		O		O	O	O		O		O		O	O	O		O		O		O	O	O		O		O		O	O	O		O		O	O
1	0	1	0	1	0	42		O		O		O	O	O		O		O		O	O	O		O		O		O	O	O		O	O	O		O	O	O		O		O		O	O	O		O		O	O
1	0	1	0	1	1	43		O		O		O	O	O		O		O		O	O	O		O		O		O	O	O		O	O	O		O	O	O		O		O		O	O	O		O	O	O		O	O
1	0	1	1	0	0	44		O		O		O	O	O		O	O	O		O	O	O		O		O		O	O	O		O	O	O		O	O	O		O		O		O	O	O		O	O	O		O	O
1	0	1	1	0	1	45		O		O		O	O	O		O	O	O		O	O	O		O		O		O	O	O		O	O	O		O	O	O		O		O		O	O	O		O	O	O		O	O
1	0	1	1	1	0	46		O		O		O	O	O		O	O	O		O	O	O		O	O	O		O	O	O		O	O	O		O	O	O		O		O		O	O	O		O	O	O		O	O
1	0	1	1	1	1	47		O		O		O	O	O		O	O	O		O	O	O		O	O	O		O	O	O		O	O	O		O	O	O		O	O	O		O	O	O		O	O	O		O	O
1	1	0	0	0	0	48		O	O	O		O	O	O		O	O	O		O	O	O		O	O	O		O	O	O		O	O	O		O	O	O		O	O	O		O	O	O		O	O	O		O	O
1	1	0	0	0	1	49		O	O	O		O	O	O		O	O	O		O	O	O		O	O	O		O	O	O		O	O	O		O	O	O		O	O	O		O	O	O		O	O	O		O	O
1	1	0	0	1	0	50		O	O	O		O	O	O		O	O	O		O	O	O		O	O	O		O	O	O		O	O	O	O	O	O	O		O	O	O		O	O	O		O	O	O		O	O
1	1	0	0	1	1	51		O	O	O		O	O	O		O	O	O		O	O	O		O	O	O		O	O	O		O	O	O	O	O	O	O		O	O	O		O	O	O		O	O	O	O	O	O
1	1	0	1	0	0	52		O	O	O		O	O	O		O	O	O	O	O	O	O		O	O	O		O	O	O		O	O	O	O	O	O	O		O	O	O		O	O	O		O	O	O	O	O	O
1	1	0	1	0	1	53		O	O	O		O	O	O		O	O	O	O	O	O	O		O	O	O		O	O	O		O	O	O	O	O	O	O		O	O	O		O	O	O		O	O	O	O	O	O
1	1	0	1	1	0	54		O	O	O		O	O	O		O	O	O	O	O	O	O		O	O	O	O	O	O	O		O	O	O	O	O	O	O		O	O	O		O	O	O		O	O	O	O	O	O
1	1	0	1	1	1	55		O	O	O		O	O	O		O	O	O	O	O	O	O		O	O	O	O	O	O	O		O	O	O	O	O	O	O		O	O	O	O	O	O	O		O	O	O	O	O	O
1	1	1	0	0	0	56		O	O	O	O	O	O	O		O	O	O	O	O	O	O		O	O	O	O	O	O	O		O	O	O	O	O	O	O		O	O	O	O	O	O	O		O	O	O	O	O	O
1	1	1	0	0	1	57		O	O	O	O	O	O	O		O	O	O	O	O	O	O		O	O	O	O	O	O	O		O	O	O	O	O	O	O		O	O	O	O	O	O	O		O	O	O	O	O	O
1	1	1	0	1	0	58		O	O	O	O	O	O	O		O	O	O	O	O	O	O		O	O	O	O	O	O	O	O	O	O	O	O	O	O	O		O	O	O	O	O	O	O		O	O	O	O	O	O
1	1	1	0	1	1	59		O	O	O	O	O	O	O		O	O	O	O	O	O	O		O	O	O	O	O	O	O	O	O	O	O	O	O	O	O		O	O	O	O	O	O	O	O	O	O	O	O	O	O
1	1	1	1	0	0	60		O	O	O	O	O	O	O	O	O	O	O	O	O	O	O		O	O	O	O	O	O	O	O	O	O	O	O	O	O	O		O	O	O	O	O	O	O	O	O	O	O	O	O	O

— The free-running counters can be cleared on compare-match A.

- Three independent interrupts
  — Two compare-match interrupts and one overflow interrupt can be requested indep



[Legend]

OCRA, OCRB:	Output compare registers A and B (16 bits)
OCRAR,OCRAF:	Output compare registers AR and AF (16 bits)
FRC:	Free-running counter (16 bits)
TCSR:	Timer control/status register (8 bits)
TIER:	Timer interrupt enable register (8 bits)
TCR:	Timer control register (8 bits)
TOCR:	Timer output compare control register (8 bits)

Figure 10.1 Block Diagram of 16-Bit Free-Running Timer

- Timer control/status register (TCSR)
- Timer control register (TCR)
- Timer output compare control register (TOCR)

Note: OCRA and OCRB share the same address. Register selection is controlled by th bit in TOCR.

10.2.1 Free-Running Counter (FRC)

FRC is a 16-bit readable/writable up-counter. The clock source is selected by bits CKS1 CKS0 in TCR. FRC can be cleared by compare-match A. When FRC overflows from H H'0000, the overflow flag bit (OVF) in TCSR is set to 1. FRC should always be accesse units; cannot be accessed in 8-bit units. FRC is initialized to H'0000.

10.2.2 Output Compare Registers A and B (OCRA and OCRB)

The FRT has two output compare registers, OCRA and OCRB, each of which is a 16-bi readable/writable register whose contents are continually compared with the value in FR a match is detected (compare-match), the corresponding output compare flag (OCFA or set to 1 in TCSR. OCR should always be accessed in 16-bit units; cannot be accessed in units. OCR is initialized to H'FFFF.

OCRAR and OCRAF should always be accessed in 16-bit units; cannot be accessed in 8-
OCRAR and OCRAF are initialized to H'FFFF.

				request (OCIA) when output compare flag A (OC TCSR is set to 1. 0: OCIA requested by OCFA is disabled 1: OCIA requested by OCFA is enabled
2	OCIBE	0	R/W	Output Compare Interrupt B Enable Selects whether to enable output compare interr request (OCIB) when output compare flag B (OC TCSR is set to 1. 0: OCIB requested by OCFB is disabled 1: OCIB requested by OCFB is enabled
1	OVIE	0	R/W	Timer Overflow Interrupt Enable Selects whether to enable a free-running timer o request interrupt (FOVI) when the timer overflow (OVF) in TCSR is set to 1. 0: FOVI requested by OVF is disabled 1: FOVI requested by OVF is enabled
0	—	0	R	Reserved This bit is always read as 0 and cannot be modi

				[Setting condition] When FRC = OCRA [Clearing condition] Read OCFA when OCFA = 1, then write 0 to OC
2	OCFB	0	R/(W)*	Output Compare Flag B Indicates that the FRC value matches the OCRB [Setting condition] When FRC = OCRB [Clearing condition] Read OCFB when OCFB = 1, then write 0 to OC
1	OVF	0	R/(W)*	Overflow Flag Indicates that the FRC has overflowed. [Setting condition] When FRC overflows (changes from H'FFFF to H [Clearing condition] Read OVF when OVF = 1, then write 0 to OVF
0	CCLRA	0	R/W	Counter Clear A Selects whether the FRC is to be cleared on com match A (when the FRC and OCRA values match 0: FRC clearing is disabled 1: FRC is cleared on compare-match A

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

0	CKS0	0	R/W	Select clock source for FRC. 00: ϕ/2 internal clock source 01: ϕ/8 internal clock source 10: ϕ/32 internal clock source 11: Reserved

				Specifies whether OCRA is used in the normal op mode or in the operating mode using OCRAR and OCRAF. 0: The normal operating mode is specified for OC 1: The operating mode using OCRAR and OCRA specified for OCRA
5	—	0	R	Reserved This bit is always read as 0 and cannot be modifi
4	OCRS	0	R/W	Output Compare Register Select OCRA and OCRB share the same address. Whe address is accessed, the OCRS bit selects which is accessed. The operation of OCRA or OCRB is affected. 0: OCRA is selected 1: OCRB is selected
3 to 0	—	All 0	R	Reserved These bits are always read as 0 and cannot be m

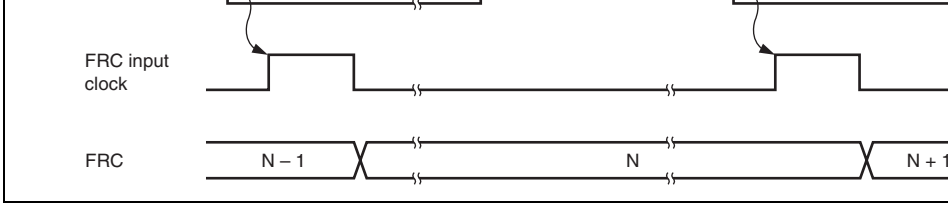


Figure 10.2 Increment Timing with Internal Clock Source

10.3.2 Output Compare Output Timing

A compare-match signal occurs at the last state when the FRC and OCR values match (a timing when the FRC updates the counter value). When a compare-match signal occurs, selected by the OLVL bit in TOCR is output at the output compare pin (FTOA or FTOB 10.3 shows the timing of this operation for compare-match A.

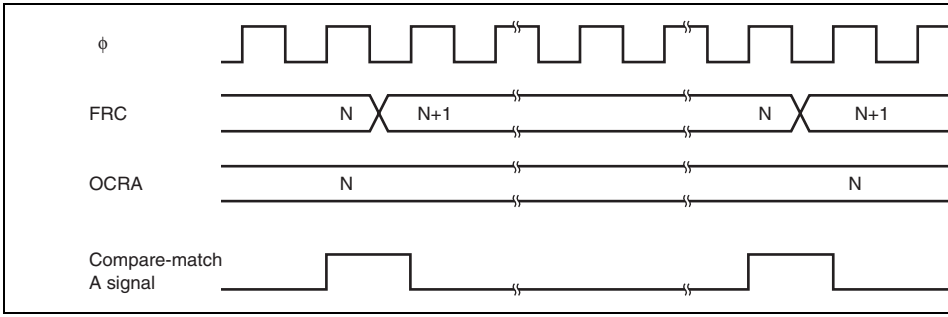


Figure 10.3 Timing of Output Compare A Output

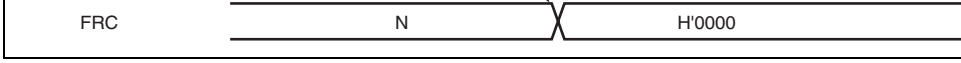


Figure 10.4 Clearing of FRC by Compare-Match A Signal

10.3.4 Timing of Output Compare Flag (OCF) Setting

The output compare flag, OCFA or OCFB, is set to 1 by a compare-match signal generate
the FRC value matches the OCRA or OCRB value. This compare-match signal is generat
last state in which the two values match, just before FRC increments to a new value. Whe
FRC and OCRA or OCRB value match, the compare-match signal is not generated until t
cycle of the clock source. Figure 10.5 shows the timing of setting the OCFA or OCFB fla

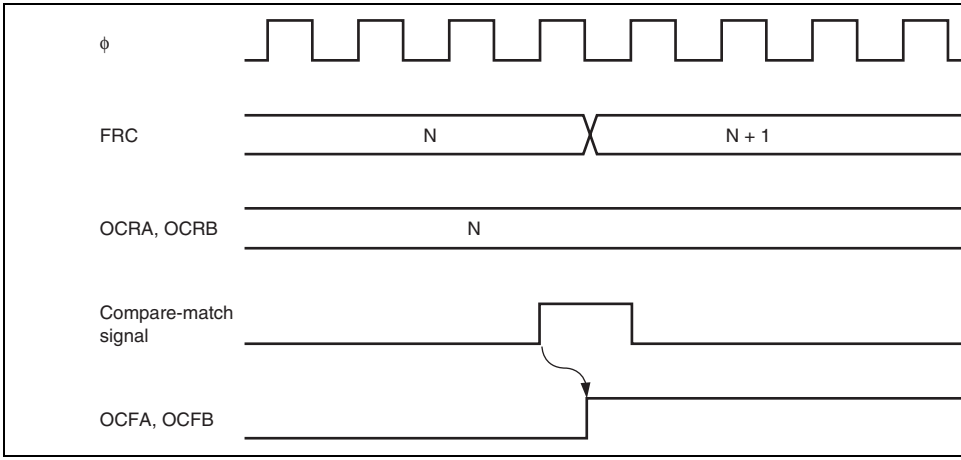


Figure 10.5 Timing of Output Compare Flag (OCFA or OCFB) Setting

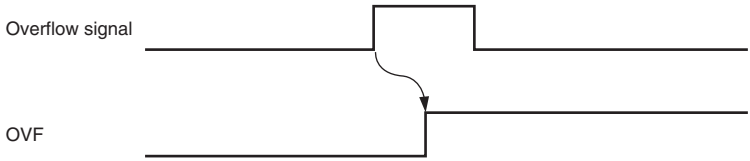


Figure 10.6 Timing of Overflow Flag (OVF) Setting

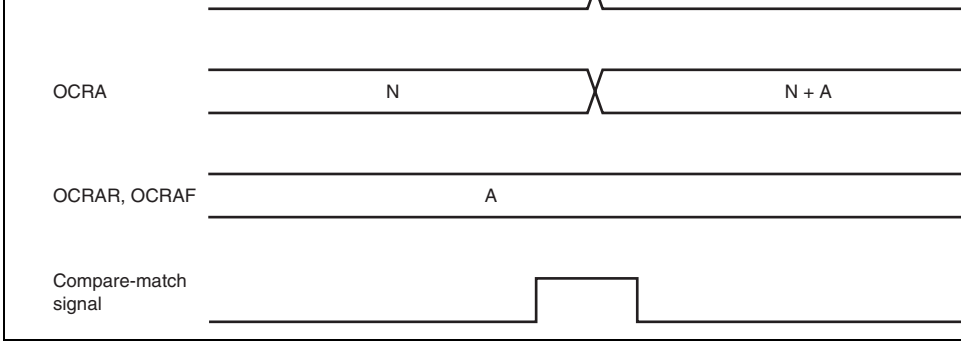


Figure 10.7 OCRA Automatic Addition Timing

10.4 Interrupt Sources

The free-running timer can request three interrupts: OCIA, OCIB, and FOVI. Each interr
be enabled or disabled by an enable bit in TIER. Independent signals are sent to the interr
controller for each interrupt. Table 10.1 lists the sources and priorities of these interrupts.

The OCIA and OCIB interrupts can be used as the on-chip DTC activation sources.

Table 10.1 FRT Interrupt Sources

Interrupt	Interrupt Source	Interrupt Flag	DTC Activation	Prio
OCIA	Compare match of OCRA	OCFA	Possible	High
OCIB	Compare match of OCRB	OCFB	Possible	↑
FOVI	Overflow of FRC	OVF	Not possible	Low

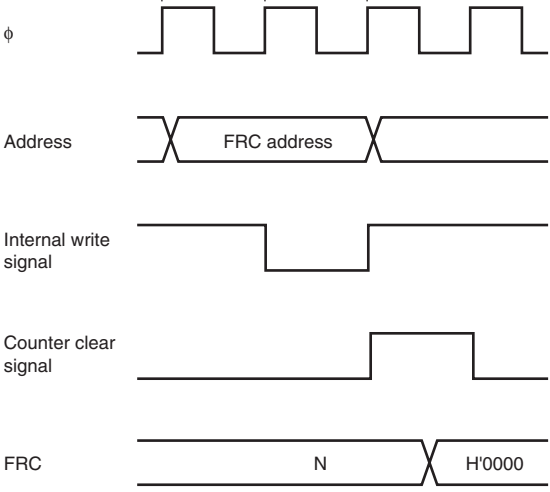


Figure 10.8 Conflict between FRC Write and Clear

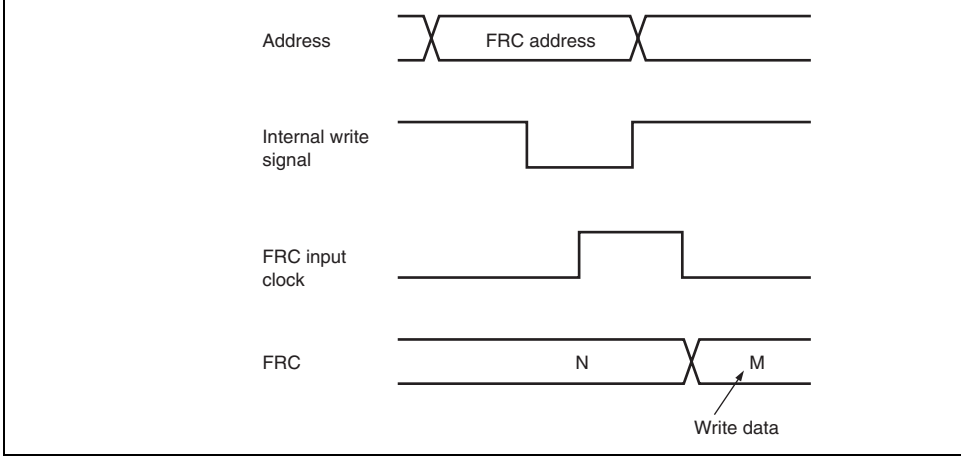


Figure 10.9 Conflict between FRC Write and Increment

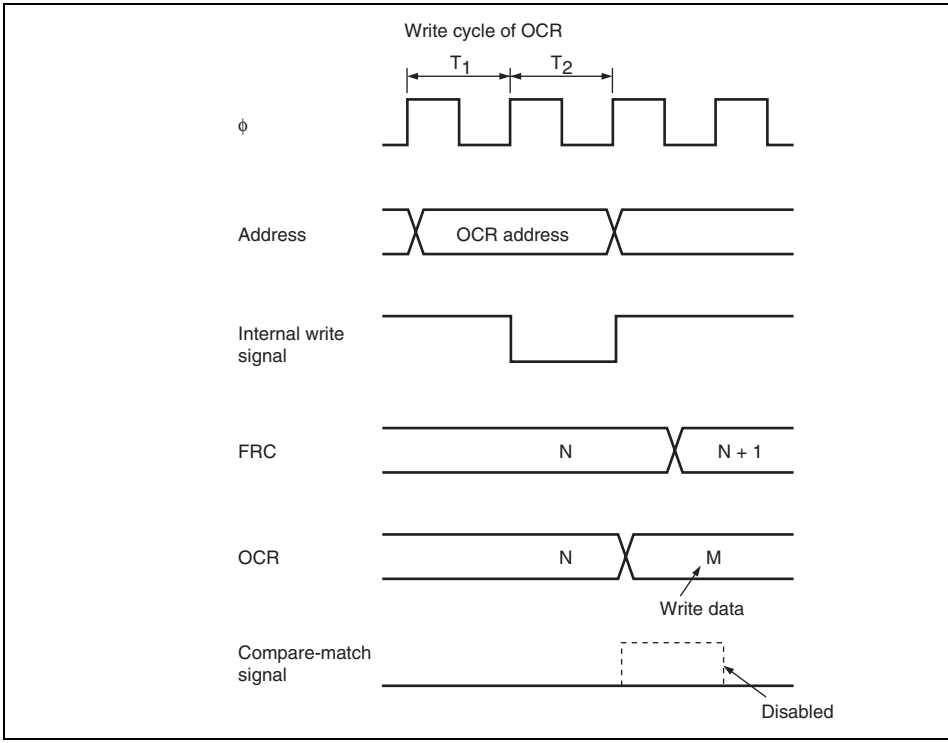


Figure 10.10 Conflict between OCR Write and Compare-Match (When Automatic Addition Function is Not Used)

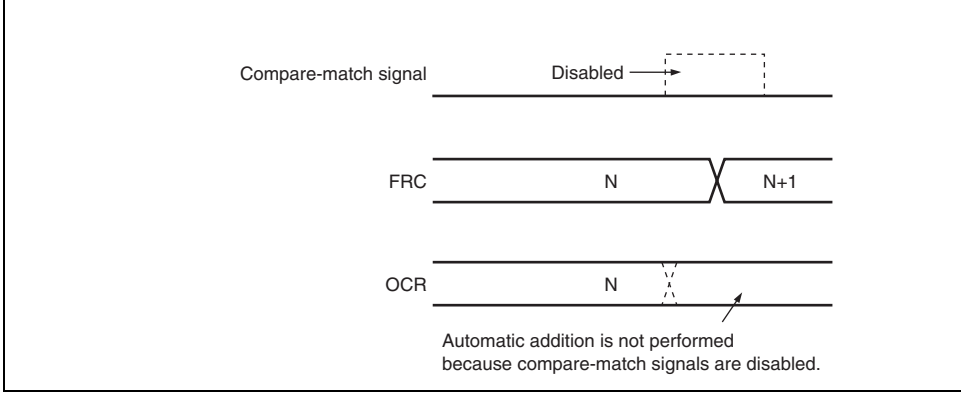
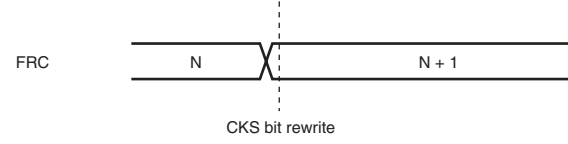


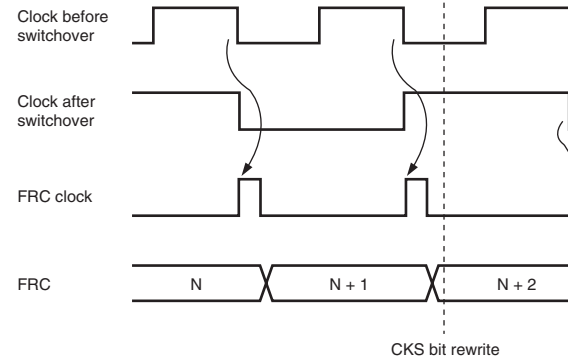
Figure 10.11 Conflict between OCR Write and Compare-Match (When Automatic Addition Function is Used)

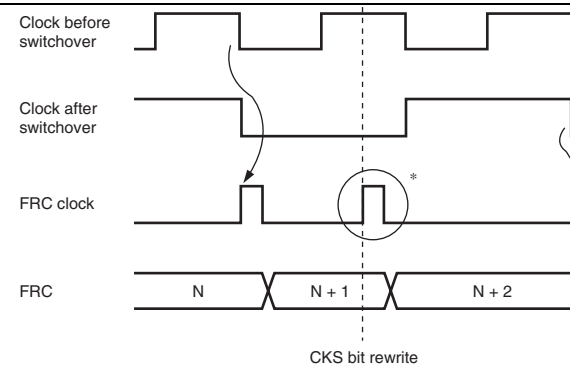
10.5.4 Switching of Internal Clock and FRC Operation

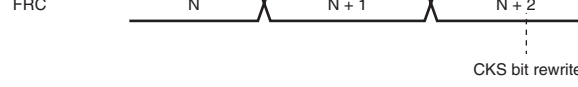
When the internal clock is changed, the changeover may source FRC to increment. This c
on the time at which the clock is switched (bits CKS1 and CKS0 are rewritten), as shown
10.2.

When an internal clock is used, the FRC clock is generated on detection of the falling edg
internal clock scaled from the system clock (ϕ). If the clock is changed when the old sour
and the new source is low, as in case no. 3 in table 10.2, the changeover is regarded as a f
edge that triggers the FRC clock, and FRC is incremented. Switching between an internal
and external clock can also source FRC to increment.



2	Switching from low to high	

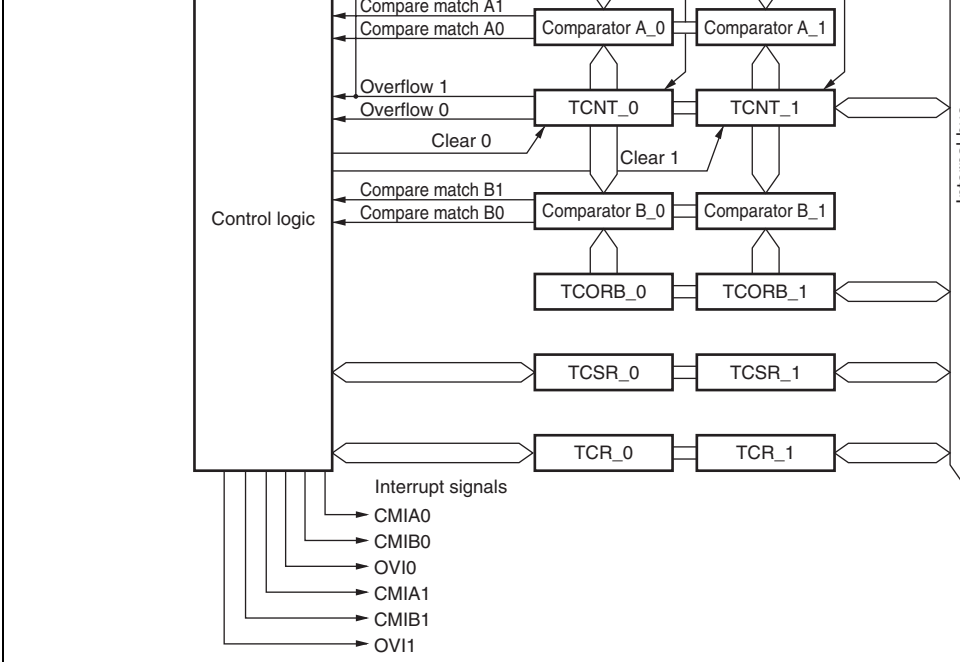
3	Switching from high to low	



Note: * Generated because the switchover is assumed to take place on a falling edge, FRC is incremented.

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

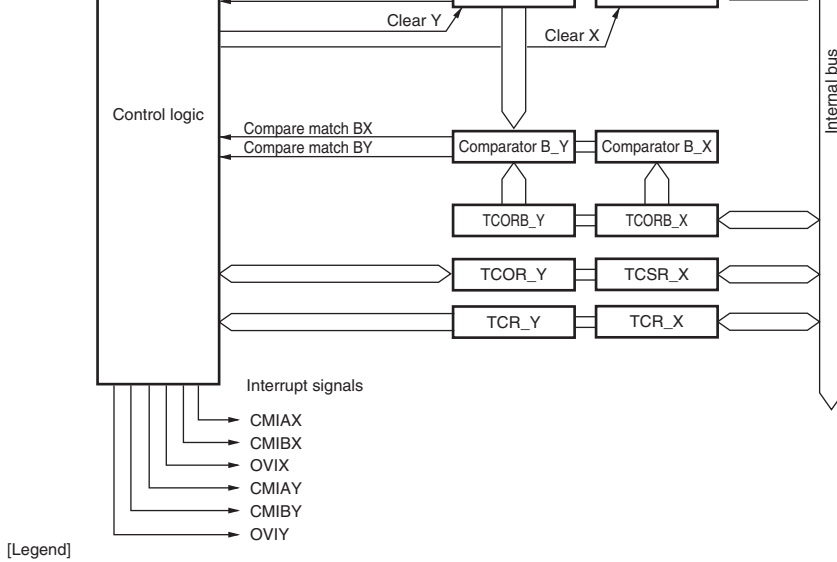
- Selection of two ways to clear the counters
  — The counters can be cleared on compare-match A and compare-match B.
- Cascading of TMR_0 and TMR_1
  (Cascading of TMR_Y and TMR_X is not allowed)
  — Operation as a 16-bit timer can be performed using TMR_0 as the upper half and as the lower half (16-bit count mode). TMR_1 can be used to count TMR_0 com match occurrences (compare-match count mode).
- Multiple interrupt sources for each channel
  — TMR_0, TMR_1, TMR_Y and TMR_X: Three interrupts: Compare-match compare-match B, and overflow



[Legend]

TCORA_0:	Time constant register A_0	TCORA_1:	Time constant register A_1
TCORB_0:	Time constant register B_0	TCORB_1:	Time constant register B_1
TCNT_0:	Timer counter_0	TCNT_1:	Timer counter_1
TCSR_0:	Timer control/status register_0	TCSR_1:	Timer control/status register_1
TCR_0:	Timer control register_0	TCR_1:	Timer control register_1

Figure 11.1 Block Diagram of 8-Bit Timer (TMR_0 and TMR_1)



[Legend]

TCORA_Y:	Time constant register A_Y	TCORA_X:	Time constant register A_X
TCORB_Y:	Time constant register B_Y	TCORB_X:	Time constant register B_X
TCNT_Y:	Timer counter_Y	TCNT_X:	Timer counter_X
TCSR_Y:	Timer control / status register_Y	TCSR_X:	Timer control / status register_X
TCR_Y:	Timer control register_Y	TCR_X:	Timer control register_X
		TCORC:	Tme constant registerC

Figure 11.2 Block Diagram of 8-Bit Timer (TMR_Y and TMR_X)

- Timer connection register S (TCONRS)*

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

* TCONRS is only provided for TMR_X

11.2.1 Timer Counter (TCNT)

Each TCNT is an 8-bit readable/writable up-counter. TCNT_0 and TCNT_1 comprise a s bit register, so they can be accessed together by word access. The clock source is selected CKS2 to CKS0 bits in TCR. TCNT can be cleared by a compare-match A signal or comp match B signal. The method of clearing can be selected by the CCLR1 and CCLR0 bits i When TCNT overflows (changes from H'FF to H'00), the OVF bit in TCSR is set to 1. T initialized to H'00.

TCNT_Y can be accessed when the TMRX/Y bit in TCONRS is 1. TCNT_X can be acce when the TMRX/Y bit in TCONRS is 0. See section 11.2.6, Timer Connection Register S (TCONRS).

11.2.3 Time Constant Register B (TCORB)

TCORB is an 8-bit readable/writable register. TCORB_0 and TCORB_ 1 comprise a sir
register, so they can be accessed together by word access. TCORB is continually compa
the value in TCNT. When a match is detected, the corresponding compare-match flag B
in TCSR is set to 1. However, comparison is disabled during the T2 state of a TCORB w
TCORB is initialized to H'FF.

TCORB_Y can be accessed when the TMRX/Y bit in TCONRS is 1. TCORB_X can be
when the TMRX/Y bit in TCONRS is 0. See section 11.2.6, Timer Connection Register
(TCONRS).

				Selects whether the CMFB interrupt request (CM enabled or disabled when the CMFB flag in TCSR 1. 0: CMFB interrupt request (CMIB) is disabled 1: CMFB interrupt request (CMIB) is enabled
6	CMIEA	0	R/W	Compare-Match Interrupt Enable A Selects whether the CMFA interrupt request (CM enabled or disabled when the CMFA flag in TCSR 1. 0: CMFA interrupt request (CMIA) is disabled 1: CMFA interrupt request (CMIA) is enabled
5	OVIE	0	R/W	Timer Overflow Interrupt Enable Selects whether the OVF interrupt request (OVI) enabled or disabled when the OVF flag in TCSR 1. 0: OVF interrupt request (OVI) is disabled 1: OVF interrupt request (OVI) is enabled
4	CCLR1	0	R/W	Counter Clear 1 and 0
3	CCLR0	0	R/W	Specify the clearing conditions of TCNT. 00: Counter clear is disabled 01: Counter clear is enabled on compare-match A 10: Counter clear is enabled on compare-match B 11: Setting prohibited
2 to 0	CKS2 to CKS0	All 0	R/W	Clock Select 2 to 0 Select the clock input to TCNT and count conditio together with the ICKS1 and ICKS0 bits in STCR. details, see table 11.1.

0	1	1	1	Increments at falling edge of internal clock
1	0	0	X	Increments at overflow signal from TCNT_
1	0	1	X	Setting prohibited
1	1	X	X	Setting prohibited

Note: * If the TMR_0 clock input is set as the TCNT_1 overflow signal and the TMR_ input is set as the TCNT_0 compare-match signal simultaneously, a count-up cannot be generated. Simultaneous setting of these conditions should be avo

[Legend] X: Don't care

0	1	1	1	Increments at falling edge of internal clock
1	0	0	X	Increments at compare-match A from TCN
1	0	1	X	Setting prohibited
1	1	X	X	Setting prohibited

Note: * If the TMR_0 clock input is set as the TCNT_1 overflow signal and the TMR_1 input is set as the TCNT_0 compare-match signal simultaneously, a count-up cannot be generated. Simultaneous setting of these conditions should be avoid

[Legend] X: Don't care

Table 11.1 (3) Clock Input to TCNT and Count Condition (TMR_X, TMR_Y)

	TCR			
Channel	**CKS2**	**CKS1**	**CKS0**	**Description**
TMR_Y	0	0	0	Disables clock input
	0	0	1	Increments at falling edge of internal clock
	0	1	0	Increments at falling edge of internal clock
	0	1	1	Increments at falling edge of internal clock
	1	X	X	Setting prohibited
TMR_Y	0	0	0	Disables clock input
	0	0	1	Increments at falling edge of internal clock
	0	1	0	Increments at falling edge of internal clock
	0	1	1	Increments at falling edge of internal clock
	1	X	X	Setting prohibited

[Legend] X: Don't care

				[Clearing condition]
				Read CMFB when CMFB = 1, then write 0 in CM
6	CMFA	0	R/(W)*	Compare-Match Flag A
				[Setting condition]
				When the values of TCNT_0 and TCORA_0 ma
				[Clearing condition]
				Read CMFA when CMFA = 1, then write 0 in CM
5	OVF	0	R/(W)*	Timer Overflow Flag
				[Setting condition]
				When TCNT_0 overflows from H′FF to H′00
				[Clearing condition]
				Read OVF when OVF = 1, then write 0 in OVF
4	ADTE	0	R/W	A/D Trigger Enable
				Selects whether the A/D conversion start reques compare match A is enabled or disabled.
				0: A/D conversion start request is disabled
				1: A/D conversion start request is enabled
3 to 0	—	All 1	R	Reserved
				These bits are always read as 1 and cannot be

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

				When the values of TCNT_1 and TCORA_1 mat
				[Clearing condition]
				Read CMFA when CMFA = 1, then write 0 in CM
5	OVF	0	R/(W)*	Timer Overflow Flag
				[Setting condition]
				When TCNT_1 overflows from H′FF to H′00
				[Clearing condition]
				Read OVF when OVF = 1, then write 0 in OVF
4 to 0	—	All 1	R	Reserved
				These bits are always read as 1 and cannot be m

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

				[Setting condition] When the values of TCNT_Y and TCORA_Y ma [Clearing condition] Read CMFA when CMFA = 1, then write 0 in CM
5	OVF	0	R/(W)*	Timer Overflow Flag [Setting condition] When TCNT_Y overflows from H'FF to H'00 [Clearing condition] Read OVF when OVF = 1, then write 0 in OVF
4 to 0	—	All 1	R	Reserved These bits are always read as 1 and cannot be

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

				[Setting condition]
				When the values of TCNT_X and TCORA_X mat
				[Clearing condition]
				Read CMFA when CMFA = 1, then write 0 in CM
5	OVF	0	R/(W)*	Timer Overflow Flag
				[Setting condition]
				When TCNT_X overflows from H'FF to H'00
				[Clearing condition]
				Read OVF when OVF = 1, then write 0 in OVF
4 to 0	—	All 1	R	Reserved
				These bits are always read as 1 and cannot be m

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

				1. The TMR_Y registers are accessed at addres H'FFFFF0 to H'FFFFF5
6 to 0	—	All 0	R/W	Reserved The initial values should not be changed.

Table 11.2 Registers Accessible by TMR_X/TMR_Y

TMRX/Y	H'FFFFF0	H'FFFFF1	H'FFFFF2	H'FFFFF3	H'FFFFF4	H'FFFFF5	H'FFFFF6
0	TMR_X	TMR_X	TMR_X	TMR_X	TMR_X	TMR_X	TMR_X
	TCR_X	TCSR_X			TCNT_X		TCORA_X
1	TMR_Y	TMR_Y	TMR_Y	TMR_Y	TMR_Y	TMR_Y	
	TCR_Y	TCSR_Y	TCORA_Y	TCORB_Y	TCNT_Y		

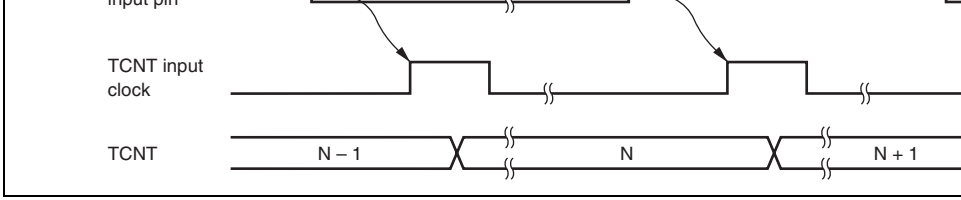


Figure 11.3 Count Timing for Internal Clock Input

11.3.2 Timing of CMFA and CMFB Setting at Compare-Match

The CMFA and CMFB flags in TCSR are set to 1 by a compare-match signal generated v
TCNT and TCOR values match. The compare-match signal is generated at the last state i
the match is true, just when the timer counter is updated. Therefore, when TCNT and TC
match, the compare-match signal is not generated until the next TCNT input clock. Figur
shows the timing of CMF flag setting.

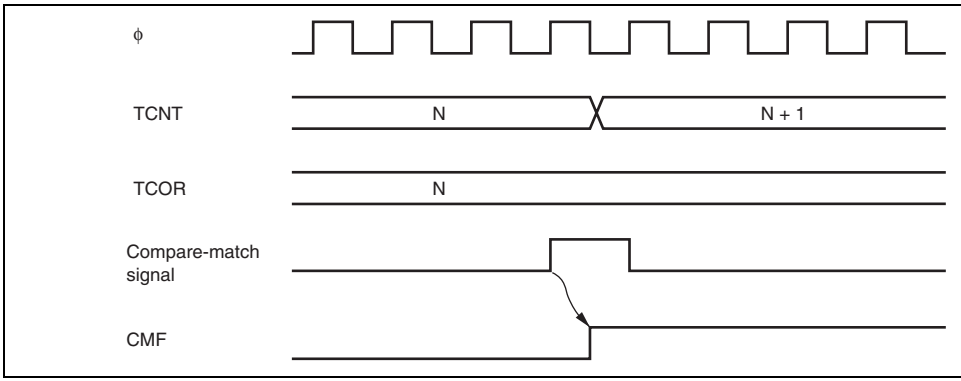


Figure 11.4 Timing of CMF Setting at Compare-Match

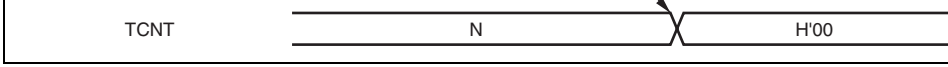


Figure 11.5 Timing of Counter Clear by Compare-Match

11.3.4 Timing of Overflow Flag (OVF) Setting

The OVF bit in TCSR is set to 1 when the TCNT overflows (changes from H'FF to H'00
11.6 shows the timing of OVF flag setting.

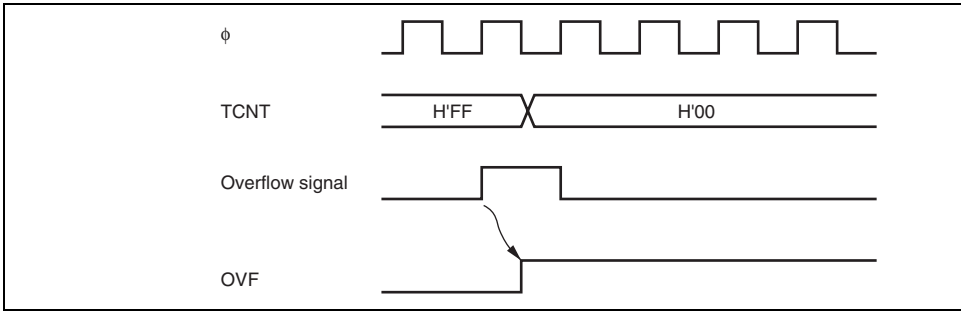


Figure 11.6 Timing of OVF Flag Setting

- Setting of compare-match flags
  - The CMF flag in TCSR_0 is set to 1 when a 16-bit compare-match occurs.
  - The CMF flag in TCSR_1 is set to 1 when a lower 8-bit compare-match occurs.
- Counter clear specification
  - If the CCLR1 and CCLR0 bits in TCR_0 have been set for counter clear at compa the 16-bit counter (TCNT_0 and TCNT_1 together) is cleared when a 16-bit comp match occurs. The 16-bit counter (TCNT_0 and TCNT_1 together) is also cleared counter clear by the TMI0 pin has been set.
  - The settings of the CCLR1 and CCLR0 bits in TCR_1 are ignored. The lower 8 bi be cleared independently.

11.4.2 Compare-Match Count Mode

When bits CKS2 to CKS0 in TCR_1 are B′100, TCNT_1 counts the occurrence of compa A for TMR_0. TMR_0 and TMR_1 are controlled independently. Conditions such as sett CMF flag, generation of interrupts, and counter clearing are in accordance with the settin each channel.

Table 11.3 Interrupt Sources of 8-Bit Timers TMR_0, TMR_1, TMR_Y, and TM

Channel	Name	Interrupt Source	Interrupt Flag	DTC Activation	Int Pri
TMR_X	CMIAX	TCORA_X compare-match	CMFA	Possible	Hig
	CMIBX	TCORB_X compare-match	CMFB	Possible	
	OVIX	TCNT_X overflow	OVF	Not possible	
TMR_0	CMIA0	TCORA_0 compare-match	CMFA	Possible	
	CMIB0	TCORB_0 compare-match	CMFB	Possible	
	OVI0	TCNT_0 overflow	OVF	Not possible	
TMR_1	CMIA1	TCORA_1 compare-match	CMFA	Possible	
	CMIB1	TCORB_1 compare-match	CMFB	Possible	
	OVI1	TCNT_1 overflow	OVF	Not possible	
TMR_Y	CMIAY	TCORA_Y compare-match	CMFA	Possible	
	CMIBY	TCORB_Y compare-match	CMFB	Possible	
	OVIY	TCNT_Y overflow	OVF	Not possible	Lov

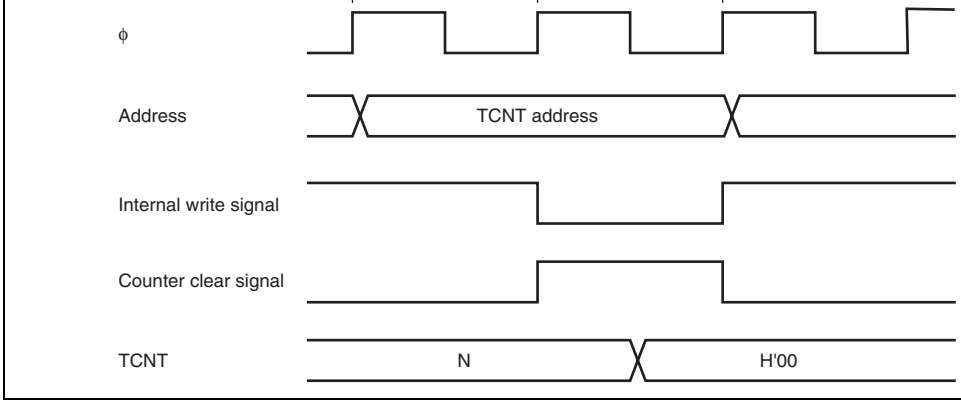


Figure 11.7 Conflict between TCNT Write and Counter Clear

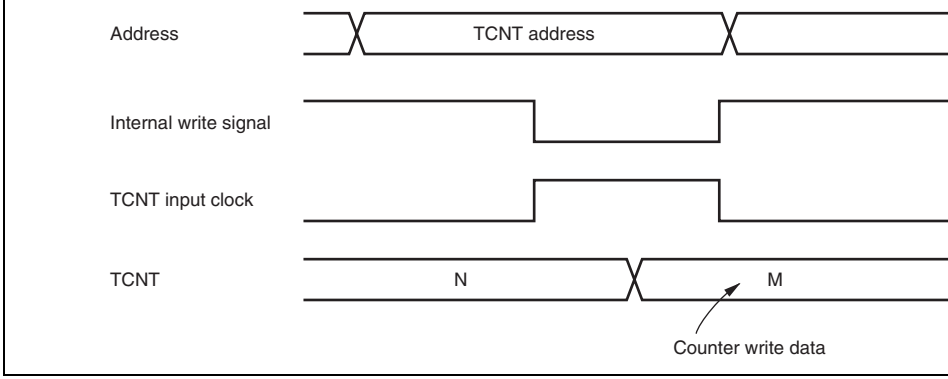


Figure 11.8 Conflict between TCNT Write and Increment

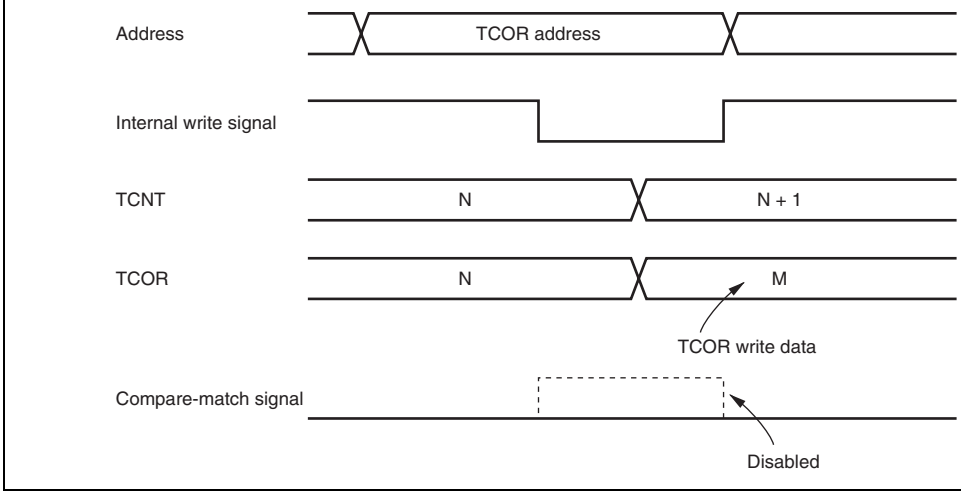
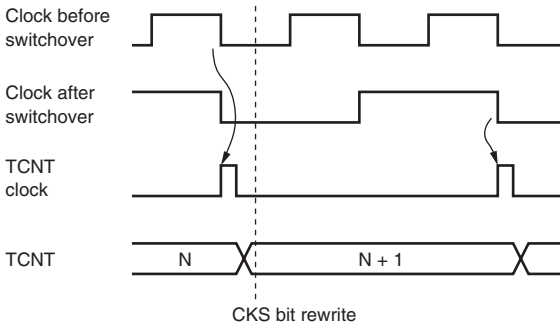
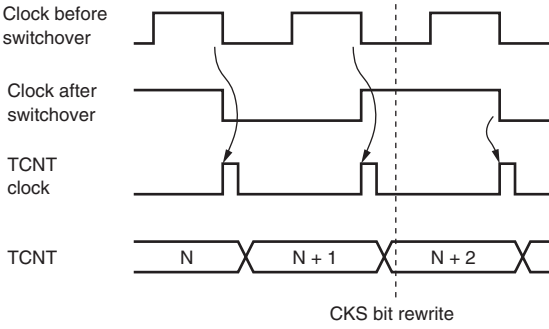
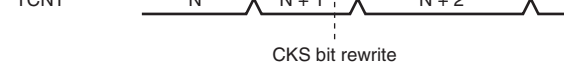


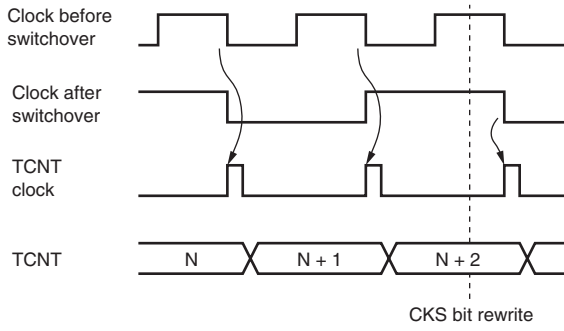
Figure 11.9 Conflict between TCOR Write and Compare-Match

Erroneous incrementation can also happen when switching between internal and externa

Table 11.4 Switching of Internal Clocks and TCNT Operation

No.	Timing of Switchover by Means of CKS1 and CKS0 Bits	TCNT Clock Operation
1	Clock switching from low to low level*[1]	
2	Clock switching from low to high level*[2]	



4	Clock switching from high to high level	

Notes: 1. Includes switching from low to stop, and from stop to low.
2. Includes switching from stop to high.
3. Includes switching from high to stop.
4. Generated on the assumption that the switchover is a falling edge; TCNT is incremented.

11.6.5 Mode Setting with Cascaded Connection

If the 16-bit count mode and compare-match count mode are set simultaneously, the inpu pulses for TCNT_0 and TCNT_1 are not generated, and thus the counters will stop opera Simultaneous setting of these two modes should be avoided.

12.1 Features

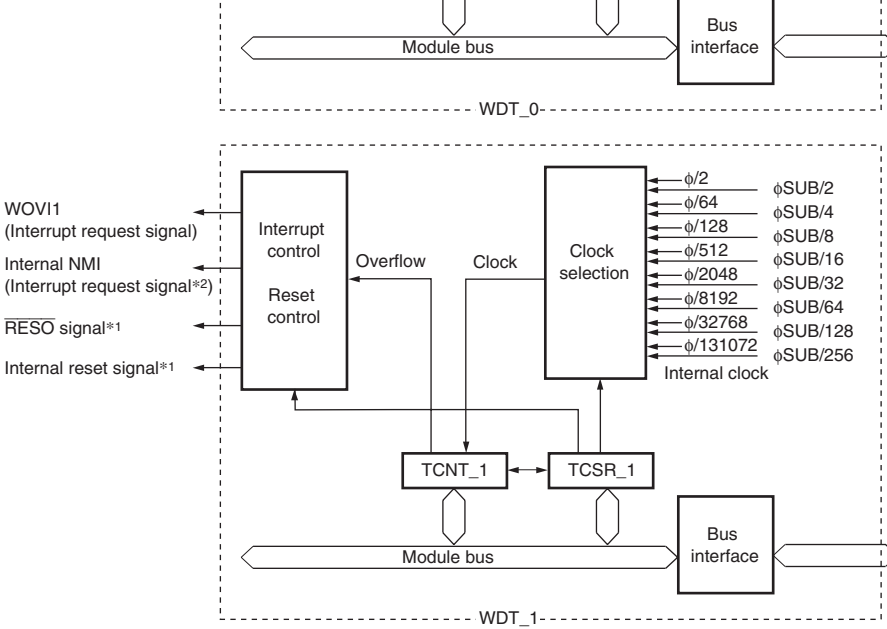
- Selectable from eight (WDT_0) or 16 (WDT_1) counter input clocks.
- Switchable between watchdog timer mode and interval timer mode

Watchdog Timer Mode:

- If the counter overflows, an internal reset or an internal NMI interrupt is generated.
- When the LSI is selected to be internally reset at counter overflow, a low level signa from the $\overline{\text{RESO}}$ pin if the counter overflows.

Internal Timer Mode:

- If the counter overflows, an internal timer interrupt (WOVI) is generated.



[Legend]

TCSR_0: Timer control/status register_0
TCNT_0: Timer counter_0
TCSR_1: Timer control/status register_1
TCNT_1: Timer counter_1

Notes: 1. The RESO signal outputs the low level signal when the internal reset signal is generated due to a TCNT overflow of either WDT_0 or WDT_1. The internal reset signal first resets the WDT in which the overflow has occurred first.

2. The internal NMI interrupt signal can be independently output from either WDT_0 or WDT The interrupt controller does not distinguish the NMI interrupt request from WDT_0 from that from WDT_1.

Figure 12.1 Block Diagram of WDT

12.3 Register Descriptions

The WDT has the following registers. To prevent accidental overwriting, TCSR and TC
to be written to in a method different from normal registers. For details, see section 12.6
on Register Access. For details on the system control register, see section 3.2.2, System
Register (SYSCR).

- Timer counter (TCNT)
- Timer control/status register (TCSR)

12.3.1 Timer Counter (TCNT)

TCNT is an 8-bit readable/writable up-counter. TCNT is initialized to H'00 when the TM
timer control/status register (TCSR) is cleared to 0.

				[Setting conditions] • When TCNT overflows (changes from H'FF to • When internal reset request generation is sele watchdog timer mode, OVF is cleared automa the internal reset. [Clearing conditions] • When TCSR is read when OVF = 1, then 0 is OVF • When 0 is written to TME
6	WT/$\overline{IT}$	0	R/W	Timer Mode Select Selects whether the WDT is used as a watchdog interval timer. 0: Interval timer mode 1: Watchdog timer mode
5	TME	0	R/W	Timer Enable When this bit is set to 1, TCNT starts counting. When this bit is cleared, TCNT stops counting an initialized to H'00.
4	—	0	R/W	Reserved The initial value should not be changed.
3	RST/$\overline{NMI}$	0	R/W	Reset or NMI Selects to request an internal reset or an NMI inte when TCNT has overflowed. 0: An NMI interrupt is requested 1: An internal reset is requested

		101: ϕ/8192 (period: 61.68 ms) 110: ϕ/32768 (period: 246.7 ms) 111: ϕ/131072 (period: 986.9 ms)

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

				the internal reset. [Clearing conditions] • When TCSR is read when OVF = 1*2, then 0 i to OVF • When 0 is written to TME
6	WT/$\overline{IT}$	0	R/W	Timer Mode Select Selects whether the WDT is used as a watchdog interval timer. 0: Interval timer mode 1: Watchdog timer mode
5	TME	0	R/W	Timer Enable When this bit is set to 1, TCNT starts counting. When this bit is cleared, TCNT stops counting an initialized to H'00.
4	PSS	0	R/W	Prescaler Select Selects the clock source to be input to TCNT. 0: Counts the divided cycle of ϕ–based prescaler 1: Counts the divided cycle of ϕSUB–based presc (PSS)
3	RST/$\overline{NMI}$	0	R/W	Reset or NMI Selects to request an internal reset or an NMI inte when TCNT has overflowed. 0: An NMI interrupt is requested 1: An internal reset is requested

011: ϕ/512 (cycle: 3.856 ms)
100: ϕ/2048 (cycle: 15.42 ms)
101: ϕ/8192 (cycle: 61.68 ms)
110: ϕ/32768 (cycle: 246.7 ms)
111: ϕ/131072 (cycle: 986.9 ms)
When PSS = 1:
000: ϕSUB/2 (cycle: 15.6 ms)
001: ϕSUB/4 (cycle: 31.3 ms)
010: ϕSUB/8 (cycle: 62.5 ms)
011: ϕSUB/16 (cycle: 125 ms)
100: ϕSUB/32 (cycle: 250 ms)
101: ϕSUB/64 (cycle: 500 ms)
110: ϕSUB/128 (cycle: 1 s)
111: ϕSUB/256 (cycle: 2 s)

Notes: 1. Only 0 can be written to clear the flag.
2. When OVF is polled with the interval timer interrupt disabled, OVF = 1 must b least twice.

If the RST/NMI bit of TCSR is set to 1, when the TCNT overflows, an internal reset sign
LSI is issued for 518 system clocks, and the low level signal is simultaneously output fro
$\overline{\text{RESO}}$ pin for 132 states, as shown in figure 12.2. If the RST/$\overline{\text{NMI}}$ bit is cleared to 0, whe
TCNT overflows, an NMI interrupt request is generated. Here, the output from the $\overline{\text{RESO}}$
remains high.

An internal reset request from the watchdog timer and a reset input from the $\overline{\text{RES}}$ pin are
processed in the same vector. Reset source can be identified by the XRST bit status in SY
If a reset caused by a signal input to the $\overline{\text{RES}}$ pin occurs at the same time as a reset caused
WDT overflow, the $\overline{\text{RES}}$ pin reset has priority and the XRST bit in SYSCR is set to 1.

An NMI interrupt request from the watchdog timer and an interrupt request from the NM
processed in the same vector. Do not handle an NMI interrupt request from the watchdog
and an interrupt request from the NMI pin at the same time.

WT/$\overline{IT}$: Timer mode select bit
TME: Timer enable bit
OVF: Overflow flag

Note: * After the OVF bit becomes 1, it is cleared to 0 by an internal reset. The XRST bit is also cleared to 0.

Figure 12.2 Watchdog Timer Mode (RST/$\overline{NMI}$ = 1) Operation

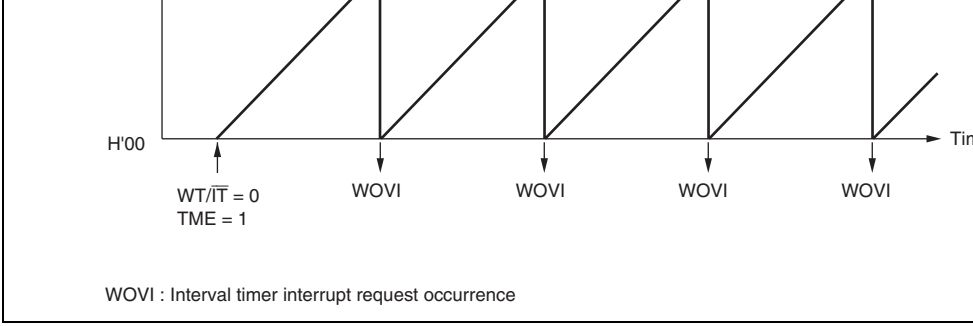


Figure 12.3 Interval Timer Mode Operation

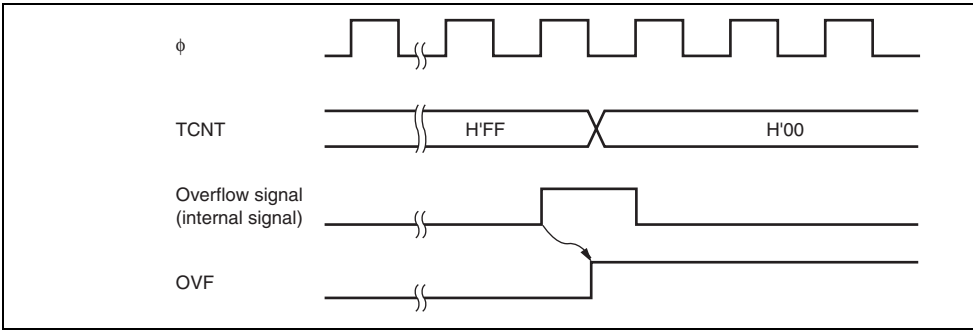


Figure 12.4 OVF Flag Set Timing

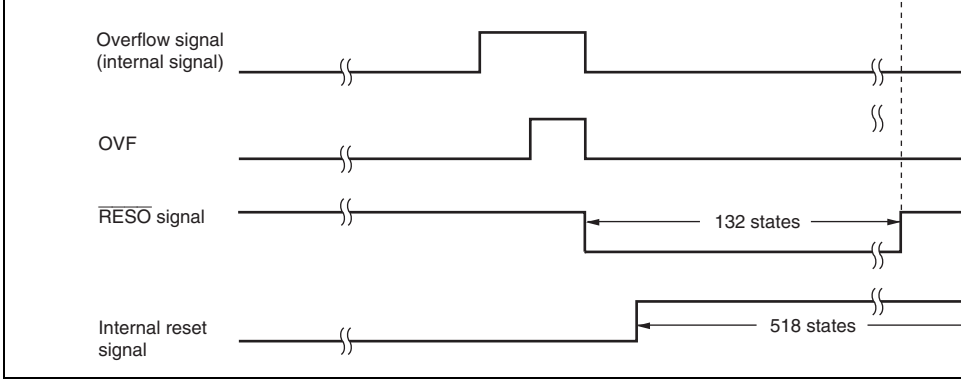


Figure 12.5 Output Timing of RESO signal

This LSI has retain state pins, which are only initialized by a system reset. The outputs o
pins are retained even when an internal reset is generated by the overflow signal of the W
more information, see section 8, I/O Ports.

Name	Interrupt Source	Interrupt Flag	DTC Activa
WOVI	TCNT overflow	OVF	Not possible

These registers must be written to by a word transfer instruction. They cannot be written byte transfer instruction.

TCNT and TCSR both have the same write address. Therefore, satisfy the relative condi shown in figure 12.6 to write to TCNT or TCSR. To write to TCNT, the higher bytes mu the value H'5A and the lower bytes must contain the write data. To write to TCSR, the h bytes must contain the value H'A5 and the lower bytes must contain the write data.

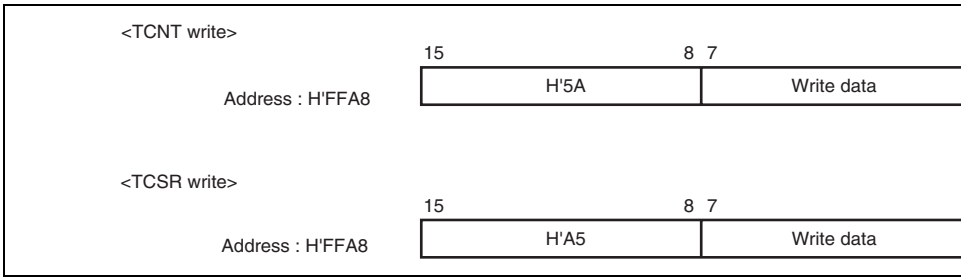


Figure 12.6 Writing to TCNT and TCSR (WDT_0)

Reading from TCNT and TCSR (Example of WDT_0):

These registers are read in the same way as other registers. The read address is H'FFA8 and H'FFA9 for TCNT.

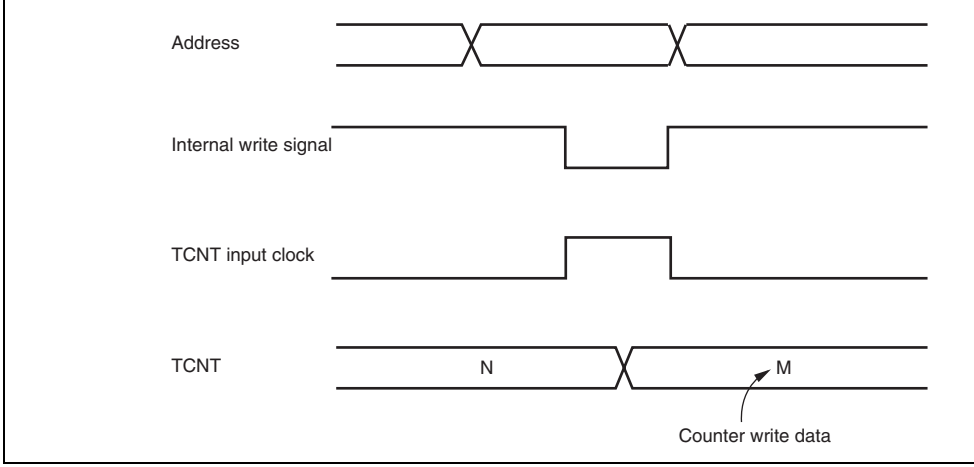


Figure 12.7 Conflict between TCNT Write and Increment

12.6.3 Changing Values of CKS2 to CKS0 Bits

If CKS2 to CKS0 bits in TCSR are written to while the WDT is operating, errors could o
the incrementation. Software must stop the watchdog timer (by clearing the TME bit to 0
changing the values of CKS2 to CKS0 bits.

12.6.4 Changing Value of PSS Bit

If the PSS bit in TCSR_1 is written to while the WDT is operating, errors could occur in
operation. Stop the watchdog timer (by clearing the TME bit to 0) before changing the va
PSS bit.

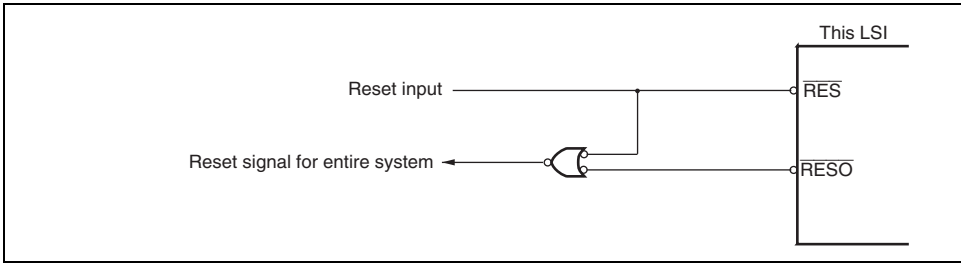


Figure 12.8 Sample Circuit for Resetting the System by the $\overline{\text{RESO}}$ Signa

13.1 Features

- Choice of asynchronous or clock synchronous serial communication mode
- Full-duplex communication capability

  The transmitter and receiver are mutually independent, enabling transmission and re
  be executed simultaneously. Double-buffering is used in both the transmitter and the
  enabling continuous transmission and continuous reception of serial data.
- On-chip baud rate generator allows any bit rate to be selected

  The external clock can be selected as a transfer clock source (except for the smart ca
  interface).
- Choice of LSB-first or MSB-first transfer (except in the case of asynchronous mode
- Four interrupt sources

  Four interrupt sources — transmit-end, transmit-data-empty, receive-data-full, and r
  error — that can issue requests.

  The transmit-data-empty and receive-data-full interrupt sources can activate DTC.
- Module stop mode availability

- Data length: 8 bits
- Receive error detection: Overrun errors

Smart Card Interface:

- An error signal can be automatically transmitted on detection of a parity error during
- Data can be automatically re-transmitted on detection of a error signal during transmi
- Both direct convention and inverse convention are supported

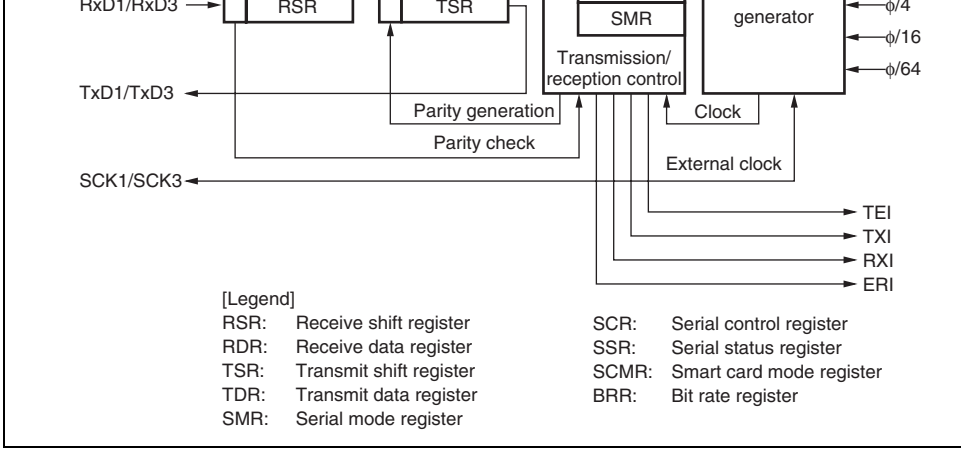


Figure 13.1 Block Diagram of SCI_1 and SCI_3

	TxD1	Output	Channel 1 transmit data output
3	SCK3	Input/Output	Channel 3 clock input/output
	RxD3	Input	Channel 3 receive data input
		Input/Output	Channel 3 transmit/receive data input/output (smart card interface is selected)
	TxD3	Output	Channel 3 transmit data output

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

13.3 Register Descriptions

The SCI has the following registers for each channel. Some bits in the serial mode registe serial status register (SSR), and serial control register (SCR) have different functions in d modes—normal serial communication interface mode and smart card interface mode; the the bits are described separately for each mode in the corresponding register sections.

- Receive shift register (RSR)
- Receive data register (RDR)
- Transmit data register (TDR)
- Transmit shift register (TSR)
- Serial mode register (SMR)
- Serial control register (SCR)
- Serial status register (SSR)
- Smart card mode register (SCMR)
- Bit rate register (BRR)

receive operations be performed. After confirming that the RDRF bit in SSR is set to 1, for only once. RDR cannot be written to by the CPU.

13.3.3 Transmit Data Register (TDR)

TDR is an 8-bit register that stores transmit data. When the SCI detects that TSR is emp transfers the transmit data written in TDR to TSR and starts transmission. The double-bu structures of TDR and TSR enables continuous serial transmission. If the next transmit d already been written to TDR when one frame of data is transmitted, the SCI transfers the data to TSR to continue transmission. Although TDR can be read from or written to by t all times, to achieve reliable serial transmission, write transmit data to TDR for only onc confirming that the TDRE bit in SSR is set to 1.

13.3.4 Transmit Shift Register (TSR)

TSR is a shift register that transmits serial data. To perform serial data transmission, the transfers transmit data from TDR to TSR, then sends the data to the TxD pin. TSR cann directly accessed by the CPU.

				1: Clock synchronous mode
6	CHR	0	R/W	Character Length (enabled only in asynchronous 0: Selects 8 bits as the data length. 1: Selects 7 bits as the data length. LSB-first is fi the MSB of TDR is not transmitted in transmiss In clock synchronous mode, a fixed data length o used.
5	PE	0	R/W	Parity Enable (enabled only in asynchronous mo When this bit is set to 1, the parity bit is added to data before transmission, and the parity bit is che reception. For a multiprocessor format, parity bit and checking are not performed regardless of the setting.
4	O/$\overline{E}$	0	R/W	Parity Mode (enabled only when the PE bit is 1 in asynchronous mode) 0: Selects even parity. 1: Selects odd parity.
3	STOP	0	R/W	Stop Bit Length (enabled only in asynchronous m Selects the stop bit length in transmission. 0: 1 stop bit 1: 2 stop bits In reception, only the first stop bit is checked. If th second stop bit is 0, it is treated as the start bit of transmit frame.
2	MP	0	R/W	Multiprocessor Mode (enabled only in asynchron mode) When this bit is set to 1, the multiprocessor communication function is enabled. The PE bit ar settings are invalid in multiprocessor mode.

				the baud rate, see section 13.3.9, Bit Rate Regis (BRR). n is the decimal display of the value of n (see section 13.3.9, Bit Rate Register (BRR)).

- Bit Functions in Smart Card Interface Mode (when SMIF in SCMR = 1)

Bit	Bit Name	Initial Value	R/W	Description
7	GM	0	R/W	GSM Mode Setting this bit to 1 allows GSM mode operation mode, the TEND set timing is put forward to 11. from the start and the clock output control functi appended. For details, see section 13.7.8, Clock Control.
6	BLK	0	R/W	Setting this bit to 1 allows block transfer mode o For details, see section 13.7.3, Block Transfer M
5	PE	0	R/W	Parity Enable (valid only in asynchronous mode When this bit is set to 1, the parity bit is added t data before transmission, and the parity bit is ch reception. Set this bit to 1 in smart card interfac
4	O/$\overline{E}$	0	R/W	Parity Mode (valid only when the PE bit is 1 in asynchronous mode) 0: Selects even parity 1: Selects odd parity For details on the usage of this bit in smart card mode, see section 13.7.2, Data Format (Except Transfer Mode).

				Timing and Reception Margin. S is described in s 13.3.9, Bit Rate Register (BRR).
1	CKS1	0	R/W	Clock Select 1 and 0
0	CKS0	0	R/W	These bits select the clock source for the baud ra generator. 00: ϕ clock (n = 0) 01: ϕ/4 clock (n = 1) 10: ϕ/16 clock (n = 2) 11: ϕ/64 clock (n = 3) For the relation between the bit rate register settir the baud rate, see section 13.3.9, Bit Rate Regist (BRR). n is the decimal display of the value of n i (see section 13.3.9, Bit Rate Register (BRR)).

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

				When this bit is set to 1, a TXI interrupt request enabled.
6	RIE	0	R/W	Receive Interrupt Enable When this bit is set to 1, RXI and ERI interrupt r are enabled.
5	TE	0	R/W	Transmit Enable When this bit is set to 1, transmission is enablec
4	RE	0	R/W	Receive Enable When this bit is set to 1, reception is enabled.
3	MPIE	0	R/W	Multiprocessor Interrupt Enable (enabled only w MP bit in SMR is 1 in asynchronous mode) When this bit is set to 1, receive data in which th multiprocessor bit is 0 is skipped, and setting of RDRF, FER, and ORER status flags in SSR is d On receiving data in which the multiprocessor bi bit is automatically cleared and normal reception resumed. For details, see section 13.5, Multipro Communication Function.
2	TEIE	0	R/W	Transmit End Interrupt Enable When this bit is set to 1, a TEI interrupt request enabled.

	1x: External clock (Inputs a clock with a frequency 16 times the from the SCK pin.)
	Clock synchronous mode: 0x: Internal clock (SCK pin functions as clock out 1x: External clock (SCK pin functions as clock inp

[Legend]

x: Don't care

4	RE	0	R/W	Receive Enable When this bit is set to 1, reception is enabled.
3	MPIE	0	R/W	Multiprocessor Interrupt Enable (enabled only w bit in SMR is 1 in asynchronous mode) Write 0 to this bit in smart card interface mode.
2	TEIE	0	R/W	Transmit End Interrupt Enable Write 0 to this bit in smart card interface mode.
1	CKE1	0	R/W	Clock Enable 1 and 0
0	CKE0	0	R/W	These bits control the clock output from the SCK mode, clock output can be dynamically switched see section 13.7.8, Clock Output Control. When GM in SMR = 0 00: Output disabled (SCK pin functions as I/O p 01: Clock output 1x: Reserved When GM in SMR = 1 00: Output fixed to low 01: Clock output 10: Output fixed to high 11: Clock output

[Legend]

x: Don't care

				Indicates whether TDR contains transmit data. [Setting conditions] • When the TE bit in SCR is 0 • When data is transferred from TDR to TSR ar ready for data write [Clearing conditions] • When 0 is written to TDRE after reading TDR • When a TXI interrupt request is issued allowir write data to TDR
6	RDRF	0	R/(W)*	Receive Data Register Full Indicates that receive data is stored in RDR. [Setting condition] • When serial reception ends normally and rece transferred from RSR to RDR [Clearing conditions] • When 0 is written to RDRF after reading RDF • When an RXI interrupt request is issued allow read data from RDR The RDRF flag is not affected and retains its prev when the RE bit in SCR is cleared to 0.

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

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

				• When 0 is written to TDRE after reading TDR • When a TXI interrupt request is issued allowin to write data to TDR
6	RDRF	0	R/(W)*[1]	Receive Data Register Full Indicates whether the receive data is stored in RD [Setting condition] • When serial reception ends normally and rece is transferred from RSR to RDR [Clearing conditions] • When 0 is written to RDRF after reading RDR • When an RXI interrupt request is issued allow to read data from RDR The RDRF flag is not affected and retains its prev value when the RE bit in SCR is cleared to 0.
5	ORER	0	R/(W)*[1]	Overrun Error [Setting condition] When the next serial reception is completed while 1 [Clearing condition] When 0 is written to ORER after reading ORER =
4	ERS	0	R/(W)*[1]	Error Signal Status [Setting condition] When a low error signal is sampled [Clearing condition] When 0 is written to ERS after reading ERS = 1

				transferred to TDR. [Setting conditions] • When both TE in SCR and ERS are 0 • When ERS = 0 and TDRE = 1 after a specifi passed after the start of 1-byte data transfer timing depends on the register setting as foll When GM = 0 and BLK = 0, 2.5 etu*2 after transmission start When GM = 0 and BLK = 1, 1.5 etu*2 after transmission start When GM = 1 and BLK = 0, 1.0 etu*2 after transmission start When GM = 1 and BLK = 1, 1.0 etu*2 after transmission start [Clearing conditions] • When 0 is written to TDRE after reading TDRE = 1 • When a TXI interrupt request is issued allow to write the next data to TDR
1	MPB	0	R	Multiprocessor Bit Not used in smart card interface mode.
0	MPBT	0	R/W	Multiprocessor Bit Transfer Write 0 to this bit in smart card interface mode.

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

				0: TDR contents are transmitted with LSB-first. Stores receive data as LSB first in RDR. 1: TDR contents are transmitted with MSB-first. Stores receive data as MSB first in RDR. The SDIR bit is valid only when the 8-bit data form used for transmission/reception; when the 7-bit da format is used, data is always transmitted/receive LSB-first.
2	SINV	0	R/W	Smart Card Data Invert Specifies inversion of the data logic level. The SIN does not affect the logic level of the parity bit. Wh parity bit is inverted, invert the O/$\overline{E}$ bit in SMR. 0: TDR contents are transmitted as they are. Rec is stored as it is in RDR. 1: TDR contents are inverted before being transm Receive data is stored in inverted form in RDR
1	—	1	R	Reserved This bit is always read as 1 and cannot be modifie
0	SMIF	0	R/W	Smart Card Interface Mode Select When this bit is set to 1, smart card interface mod selected. 0: Normal asynchronous or clock synchronous mo 1: Smart card interface mode

Asynchronous mode	$B = \frac{\phi \times 10^6}{64 \times 2^{2n-1} \times (N+1)}$	Error (%) = $\{\frac{\phi \times 10^6}{B \times 64 \times 2^{2n-1} \times (N+$
Clock synchronous mode	$B = \frac{\phi \times 10^6}{8 \times 2^{2n-1} \times (N+1)}$	—
Smart card interface mode	$B = \frac{\phi \times 10^6}{S \times 2^{2n+1} \times (N+1)}$	Error (%) = $\{\frac{\phi \times 10^6}{B \times S \times 2^{2n+1} \times (N$

[Legend]

B: Bit rate (bit/s)

N: BRR setting for baud rate generator ($0 \leq N \leq 255$)

ϕ: Operating frequency (MHz)

n and S: Determined by the SMR settings shown in the following table.

SMR Setting		
CKS1	**CKS0**	**n**
0	0	0
0	1	1
1	0	2
1	1	3

SMR Setting		
BCP1	**BCP0**	**S**
0	0	3
0	1	6
1	0	3
1	1	2

Table 13.3 shows sample N settings in BRR in normal asynchronous mode. Table 13.4 s maximum bit rate settable for each frequency. Table 13.6 and 13.8 show sample N settin BRR in clock synchronous mode and smart card interface mode, respectively. In smart c interface mode, the number of basic clock cycles S in a 1-bit data transfer time can be se For details, see section 13.7.4, Receive Data Sampling Timing and Reception Margin. T and 13.7 show the maximum bit rates with external clock input.

2400	1	64	0.16	1	80	–0.47	1	110	–
4800	0	129	0.16	0	162	0.15	0	220	0
9600	0	64	0.16	0	80	–0.47	0	110	–
19200	0	32	–1.36	0	40	–0.76	0	54	0
31250	0	19	0.00	0	24	0.00	0	33	0
38400	0	15	1.73	0	19	1.73	0	27	–

[Legend]

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

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

Table 13.4 Maximum Bit Rate for Each Frequency (Asynchronous Mode)

ϕ (MHz)	Maximum Bit Rate (bit/s)	n	N
20	625000	0	0
25	781250	0	0
34	1062500	0	0

Table 13.5 Maximum Bit Rate with External Clock Input (Asynchronous Mode)

ϕ (MHz)	External Input Clock (MHz)	Maximum Bit Rate (bi
20	5.0000	312500
25	6.2500	390625
34	8.0000	531250

5 k	1	249	2	74	2	1
10 k	1	124	1	149	1	2
25 k	0	199	0	239	1	8
50 k	0	99	0	119	0	1
100 k	0	49	0	59	0	8
250 k	0	19	0	23	0	3
500 k	0	9	0	11	0	1
1 M	0	4	0	5		
2.5 M	0	1				
5 M	0	0*				

[Legend]

Blank: Setting prohibited.

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

*: Continuous transfer or reception is not possible.

Bit Rate (bit/s)	Operating Frequency φ (MHz)											
	20.00			21.4272			25			34		
	n	N	Error (%)	n	N	Error(%)	n	N	Error (%)	n	N	
9600	0	2	–6.65	0	2	0.00	0	3	–12.49	0	4	

Table 13.9 Maximum Bit Rate for Each Frequency (Smart Card Interface Mode, S

φ (MHz)	Maximum Bit Rate (bit/s)	n	N
20.00	26882	0	0
21.4272	28800	0	0
25.00	33602	0	0
34.00	45699	0	0

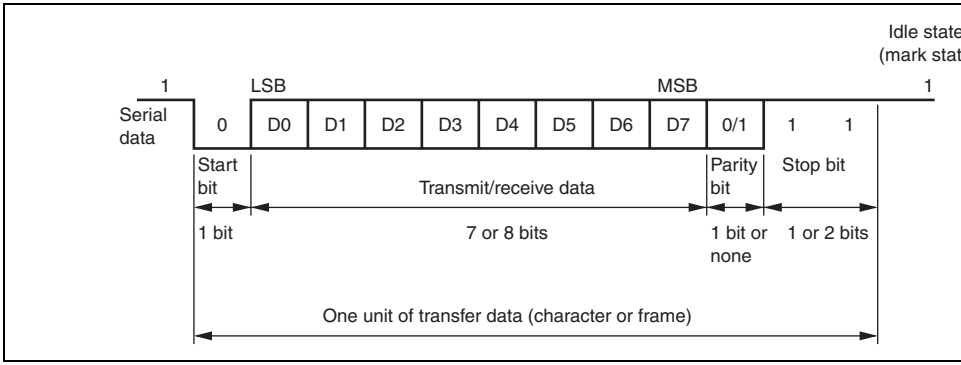


Figure 13.2 Data Format in Asynchronous Communication (Example with 8-Bit Data, Parity, Two Stop Bits)

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

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

N: Ratio of bit rate to clock (N = 16)
D: Clock duty (D = 0.5 to 1.0)
L: Frame length (L = 9 to 12)
F: Absolute value of clock rate deviation

Assuming values of F = 0 and D = 0.5 in formula (1), the reception margin is determined
formula below.

$$M = \{0.5 - 1/(2 \times 16)\} \times 100 \quad [\%] = 46.875\ \%$$

However, this is only the computed value, and a margin of 20% to 30% should be allow
system design.

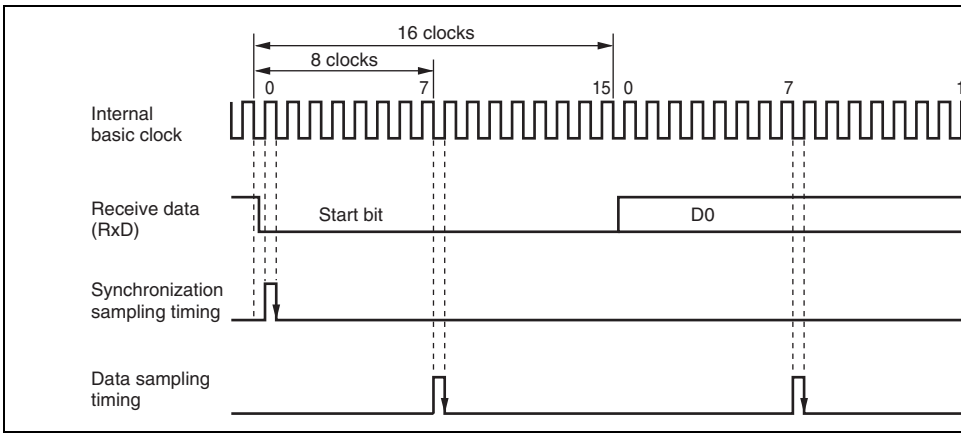


Figure 13.3 Receive Data Sampling Timing in Asynchronous Mode

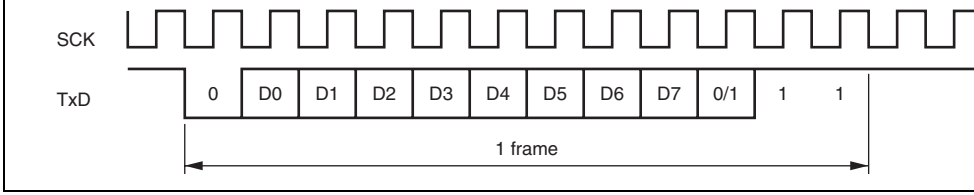
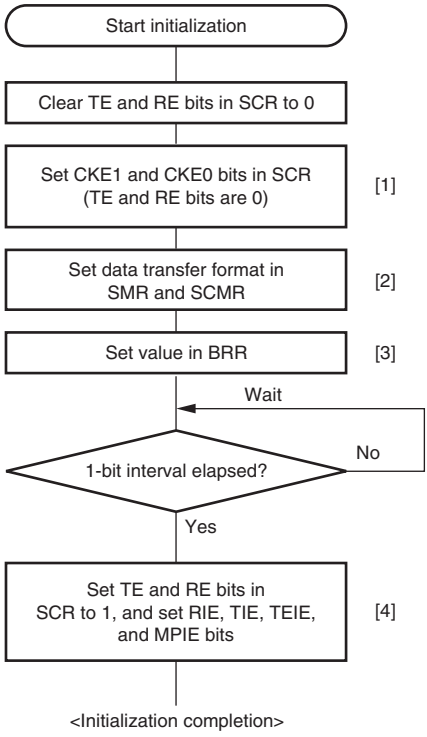


Figure 13.4 Relation between Output Clock and Transmit Data Phase (Asynchronous Mode)



[1] Set the clock selection in SCR. Be sure to clear bits RIE, TIE, TEIE, and MPIE, and bits TE and RE, to 0.

When the clock is selected in asynchronous mode, it is output immediately after SCR settings are made.

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

[3] Write a value corresponding to the bit rate to BRR. Not necessary if an external clock is used.

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

Setting the TE and RE bits enables the TxD and RxD pins to be used.

Figure 13.5 Sample SCI Initialization Flowchart

be enabled.

3. Data is sent from the TxD pin in the following order: start bit, transmit data, parity bit multiprocessor bit (may be omitted depending on the format), and stop bit.
4. The SCI checks the TDRE flag at the timing for sending the stop bit.
5. If the TDRE flag is 0, the data is transferred from TDR to TSR, the stop bit is sent, an serial transmission of the next frame is started.
6. If the TDRE flag is 1, the TEND flag in SSR is set to 1, the stop bit is sent, and then t state" is entered in which 1 is output. If the TEIE bit in SCR is set to 1 at this time, a T interrupt request is generated.

Figure 13.7 shows a sample flowchart for transmission in asynchronous mode.

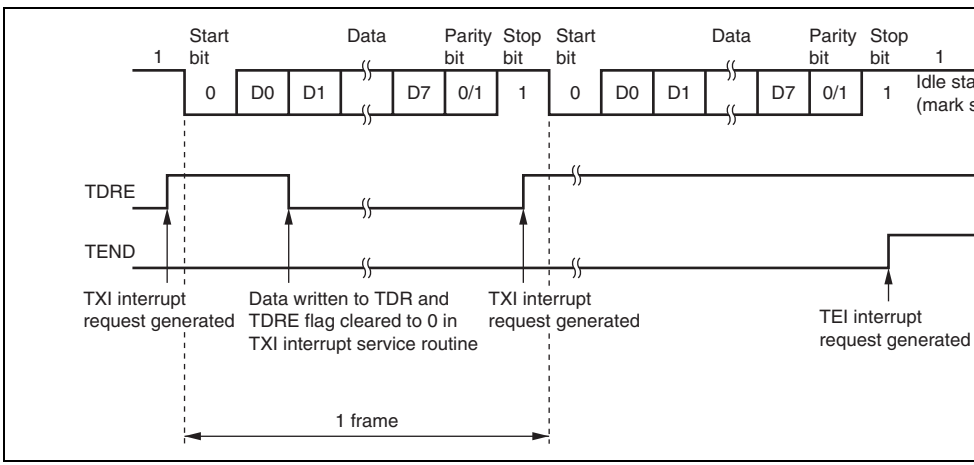
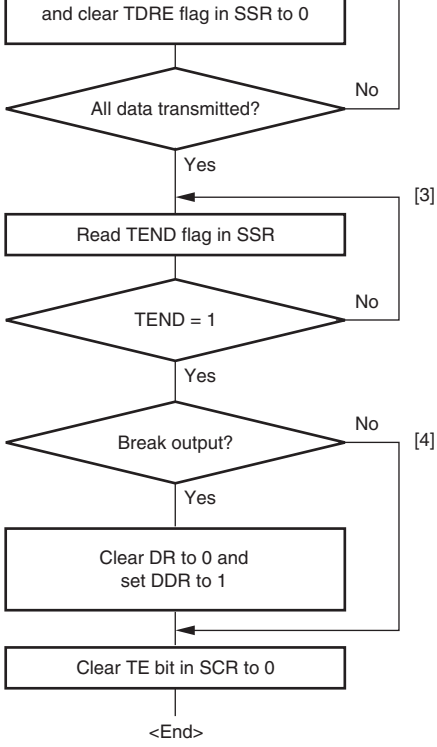


Figure 13.6 Example of Operation in Transmission in Asynchronous Mode (Example with 8-Bit Data, Parity, One Stop Bit)



[3] Serial transmission continuation procedure:

To continue serial transmission, read 1 from the TDRE flag to confirm that writing is possible, then write data to TDR, and clear the TDRE flag to 0. However, the TDRE flag is checked and cleared automatically when the DTC is initiated by a transmit data empty interrupt (TXI) request and writes data to TDR.

[4] Break output at the end of serial transmission:
To output a break in serial transmission, set DDR for the port corresponding to the TxD pin to 1, clear DR to 0, then clear the TE bi in SCR to 0.

Figure 13.7 Sample Serial Transmission Flowchart

3. If a parity error is detected, the PER bit in SSR is set to 1 and receive data is transferr RDR. If the RIE bit in SCR is set to 1 at this time, an ERI interrupt request is generate
4. If a framing error (when the stop bit is 0) is detected, the FER bit in SSR is set to 1 an data is transferred to RDR. If the RIE bit in SCR is set to 1 at this time, an ERI interru request is generated.
5. If reception finishes successfully, the RDRF bit in SSR is set to 1, and receive data is transferred to RDR. If the RIE bit in SCR is set to 1 at this time, an RXI interrupt requ generated. Because the RXI interrupt routine reads the receive data transferred to RDI reception of the next receive data has finished, continuous reception can be enabled.

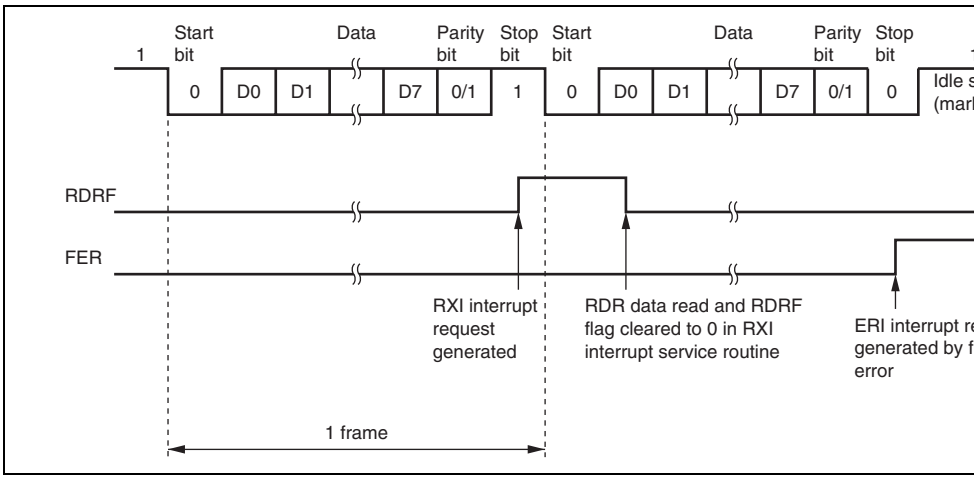
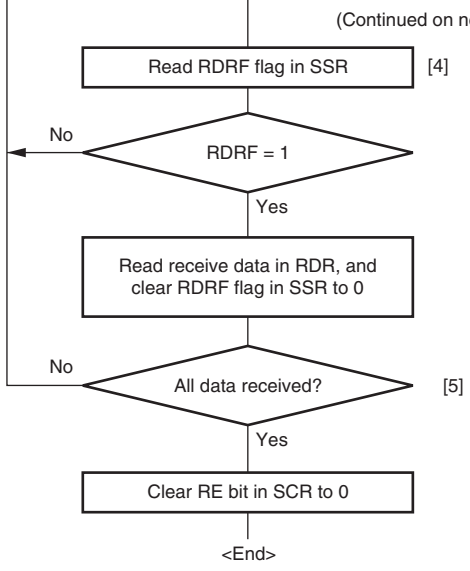


Figure 13.8 Example of SCI Operation in Reception (Example with 8-Bit Data, Parity, One Stop Bit)

0	0	1	0	Transferred to RDR	Framing error
0	0	0	1	Transferred to RDR	Parity error
1	1	1	0	Lost	Overrun error + fram
1	1	0	1	Lost	Overrun error + parit
0	0	1	1	Transferred to RDR	Framing error + parit
1	1	1	1	Lost	Overrun error + fram parity error

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



1. In the case of a framing error, a break can be detected by reading t value of the input port correspondir the RxD pin.

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

[5] Serial reception continuation proce To continue serial reception, before stop bit for the current frame is received, read the RDRF flag, read RDR, and clear the RDRF flag to 0 However, the RDRF flag is cleared automatically when the DTC is initia by an RXI interrupt and reads data RDR.

[Legend]
∨ : Logical add (OR)

Figure 13.9 Sample Serial Reception Flowchart (1)

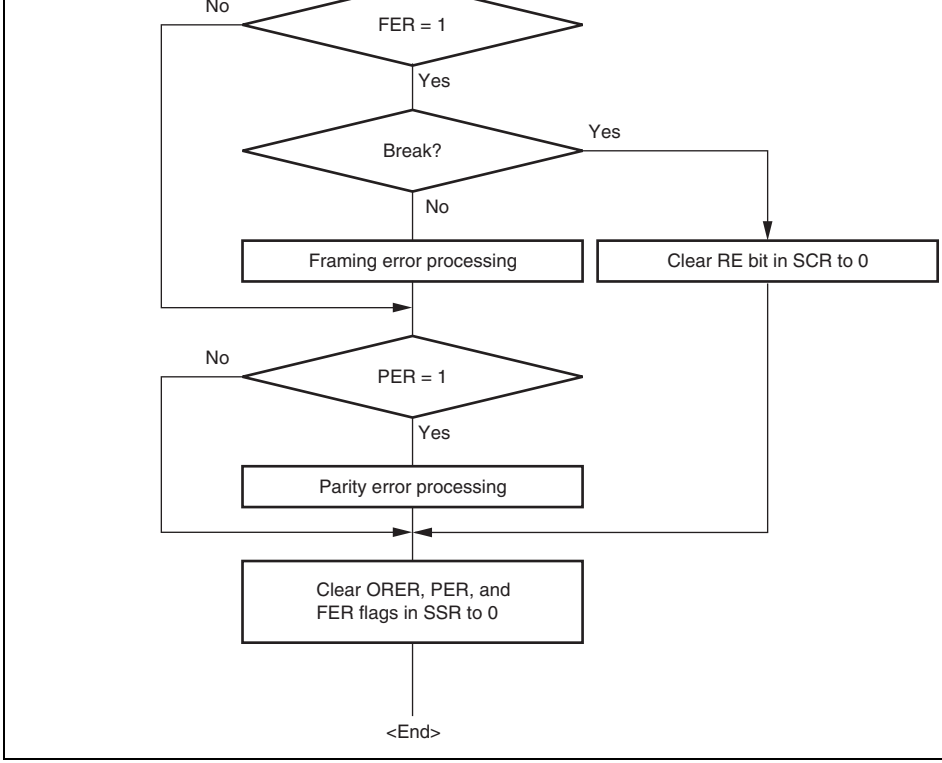


Figure 13.9 Sample Serial Reception Flowchart (2)

transmission cycle, and if the multiprocessor bit is 0, the cycle is a data transmission cycl
13.10 shows an example of inter-processor communication using the multiprocessor form
transmitting station first sends the ID code of the receiving station with which it wants to
serial communication as data with a 1 multiprocessor bit added. It then sends transmit dat
with a 0 multiprocessor bit added. When data with a 1 multiprocessor bit is received, the
station compares that data with its own ID. The station whose ID matches then receives th
sent next. Stations whose ID does not match continue to skip data until data with a 1
multiprocessor bit is again received.

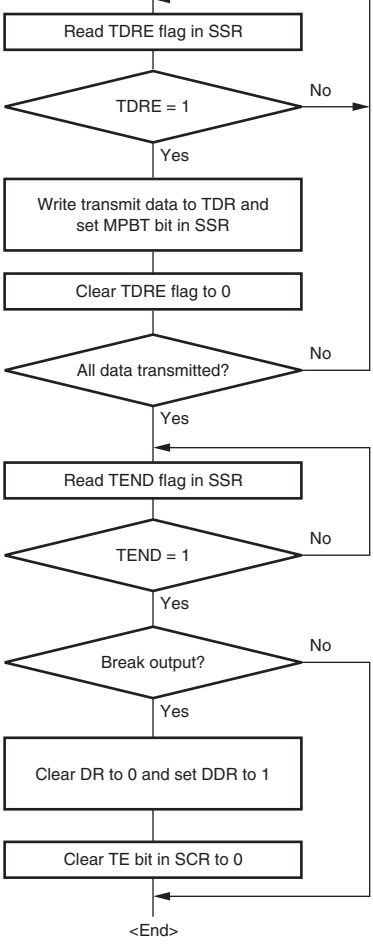
The SCI uses the MPIE bit in SCR to implement this function. When the MPIE bit is set t
transfer of receive data from RSR to RDR, error flag detection, and setting the RDRF, FE
ORER status flags in SSR to 1 are prohibited until data with a 1 multiprocessor bit is rece
reception of a receive character with a 1 multiprocessor bit, the MPB bit in SSR is set to
MPIE bit is automatically cleared, thus normal reception is resumed. If the RIE bit in SCI
1 at this time, an RXI interrupt is generated.

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

		ID transmission cycle = receiving station specification	Data transmission cycle = Data transmission to receiving station specified by ID

[Legend]
MPB: Multiprocessor bit

Figure 13.10 Example of Communication Using Multiprocessor Format (Transmission of Data H'AA to Receiving Station A)



After the TE bit is set to 1, a frame of 1s is output, and transmission is enabled.

[2] SCI status check and transmit data write:
Read SSR and check that the TDRE flag is set to 1, then write transmit data to TDR. Set the MPBT bit in SSR to 0 or 1. Finally, clear the TDRE flag to 0.

[3] Serial transmission continuation procedure:
To continue serial transmission, be sure to read 1 from the TDRE flag to confirm that writing is possible, then write data to TDR, and then clear the TDRE flag to 0. However, the TDRE flag is checked and cleared automatically when the DTC is initiated by a transmit data empty interrupt (TXI) request and writes data to TDR.

[4] Break output at the end of serial transmission:
To output a break in serial transmission, set port DDR to 1, clear DR to 0, and then clear the TE bit in SCR to 0.

Figure 13.11 Sample Multiprocessor Serial Transmission Flowchart

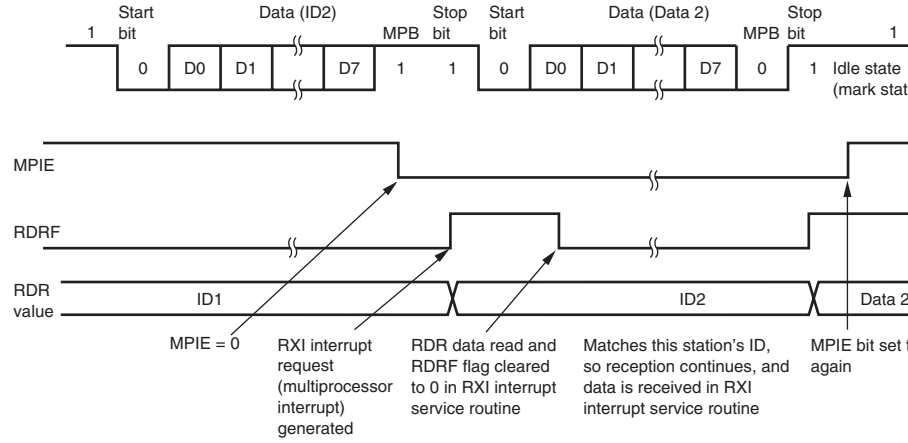
request (multiprocessor interrupt) generated

and RDRF flag cleared to 0 in RXI interrupt service routine

MPIE bit is set to 1 again

not generated, and retains its state

(a) Data does not match station's ID



(b) Data matches station's ID

Figure 13.12 Example of SCI Operation in Reception (Example with 8-Bit Data, Multiprocessor Bit, One Stop Bit)

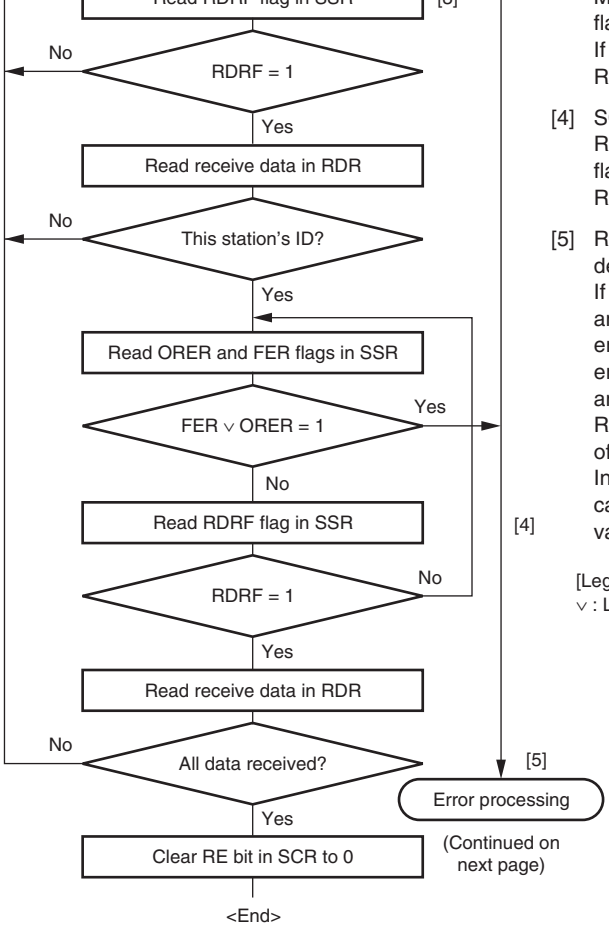


Figure 13.13 Sample Multiprocessor Serial Reception Flowchart (1)

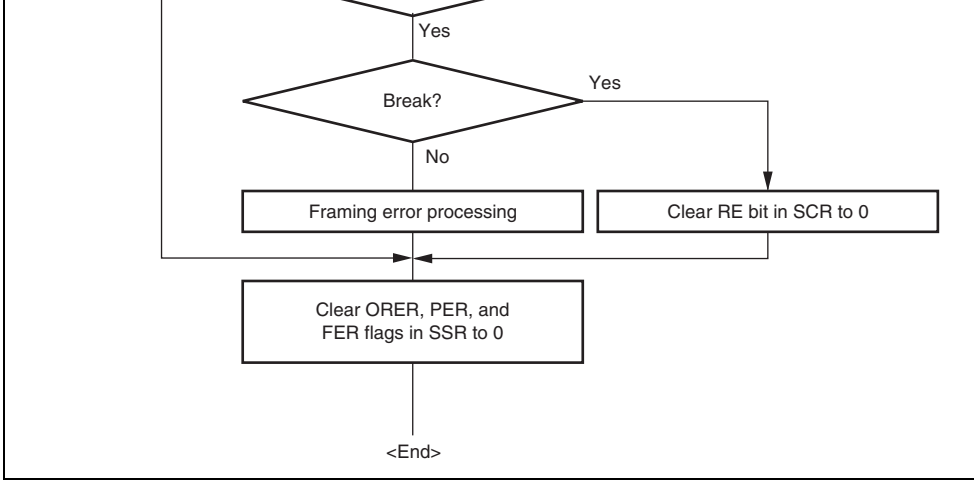


Figure 13.13 Sample Multiprocessor Serial Reception Flowchart (2)

double-buffered structure, so that the next transmit data can be written during transmiss
previous receive data can be read during reception, enabling continuous data transfer.

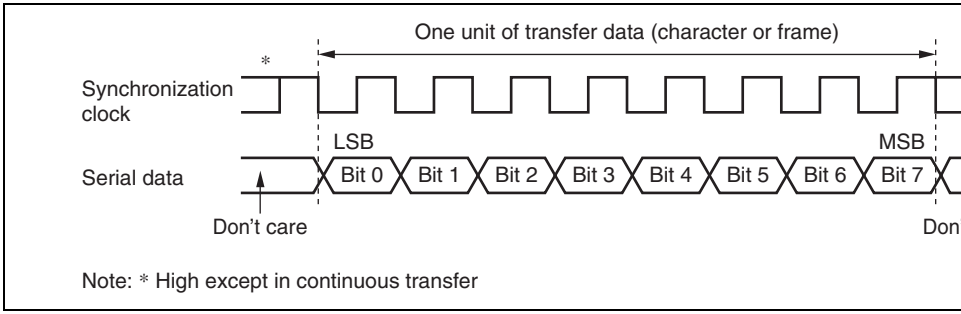
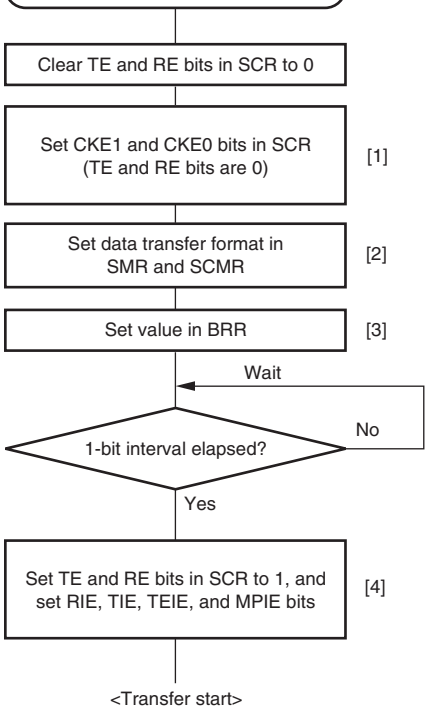


Figure 13.14 Data Format in Synchronous Communication (LSB-First)

13.6.1 Clock

Either an internal clock generated by the on-chip baud rate generator or an external synchronization clock input at the SCK pin can be selected, according to the setting of t 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 transfe character, and when no transfer is performed the clock is fixed high.



to clear bits RIE, TIE, TEIE, and MPIE
TE and RE to 0.

[2] Set the data transfer format in SMR 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 13.15 Sample SCI Initialization Flowchart

3. 8-bit data is sent from the TxD pin synchronized with the output clock when output
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 tra
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 main
output state of the last bit. If the TEIE bit in SCR is set to 1 at this time, a TEI interr
is generated. The SCK pin is fixed high.

Figure 13.17 shows a sample flowchart for serial data transmission. Even if the TDRE f
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.

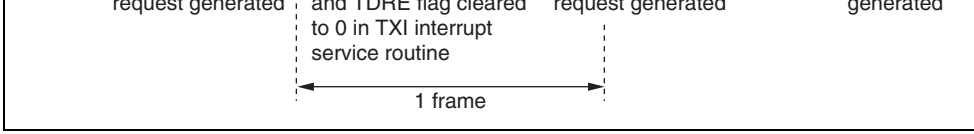
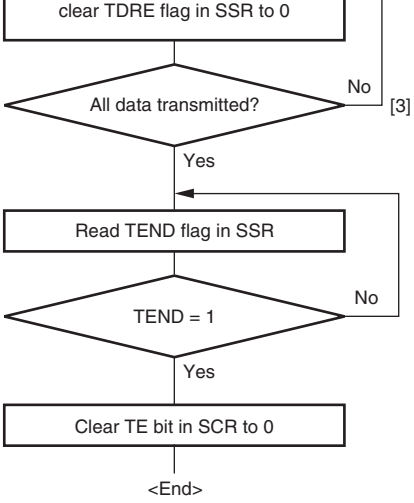


Figure 13.16 Sample SCI Transmission Operation in Clock Synchronous Mo



confirm that writing is possible, then write data to TDR, and then clear the TDRE flag to 0.
However, the TDRE flag is checked and cleared automatically when the DTC is initiated by a transmit data empty interrupt (TXI) request and writes data to TDR.

Figure 13.17 Sample Serial Transmission Flowchart

3. If reception finishes successfully, the RDRF bit in SSR is set to 1, and receive data is transferred to RDR. If the RIE bit in SCR is set to 1 at this time, an RXI interrupt requ generated. Because the RXI interrupt routine reads the receive data transferred to RDI reception of the next receive data has finished, continuous reception can be enabled.

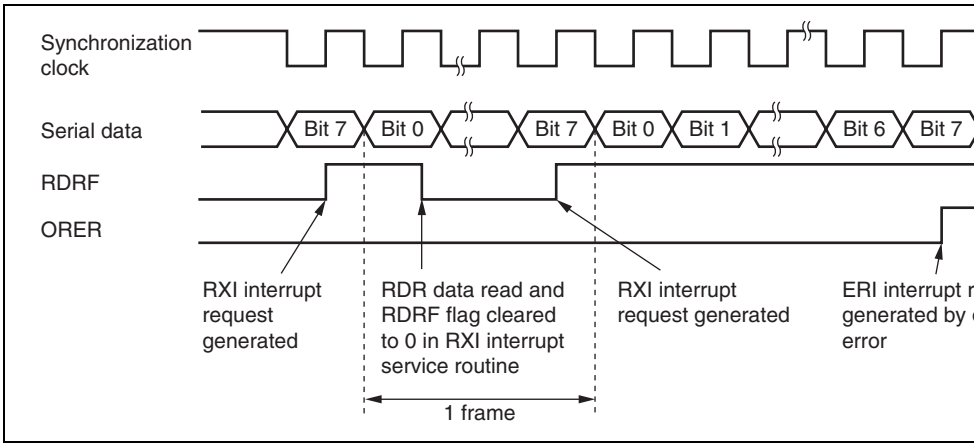
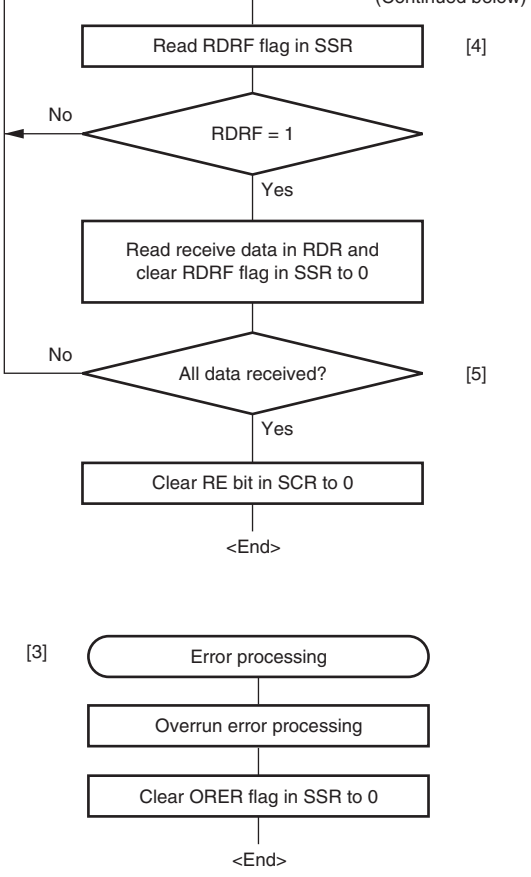


Figure 13.18 Example of SCI Receive Operation in Clock Synchronous Mod

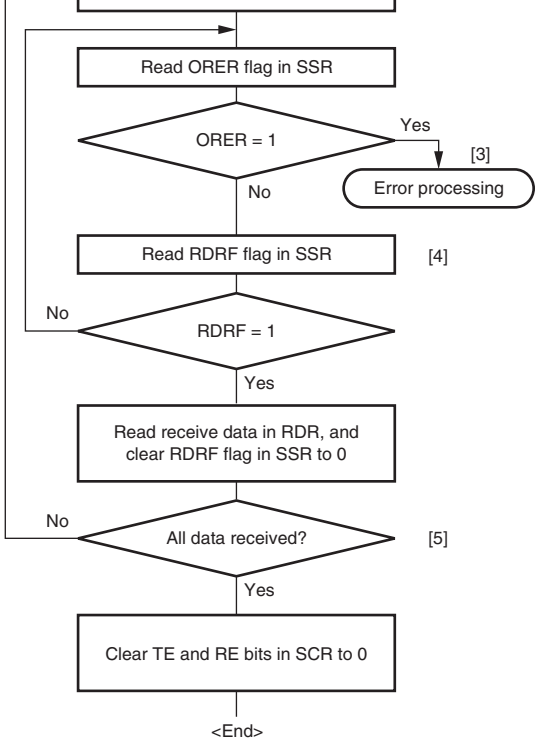
Reception cannot be resumed while a receive error flag is set to 1. Accordingly, clear the FER, PER, and RDRF bits to 0 before resuming reception. Figure 13.19 shows a sample for serial data reception.



Read SSR and check that the R flag is set to 1, then read the rec data in RDR and clear the RDR to 0.
Transition of the RDRF flag from can also be identified by an RXI interrupt.

[5] Serial reception continuation procedure:
To continue serial reception, be the MSB (bit 7) of the current fra received, reading the RDRF flag reading RDR, and clearing the R flag to 0 should be finished.
However, the RDRF flag is clea automatically when the DTC is initiated by a receive data full in (RXI) and reads data from RDR

Figure 13.19 Sample Serial Reception Flowchart



Note: When switching from transmit or receive operation to simultaneous transmit and receive operations, first clear the TE bit and RE bit to 0, then set both these bits to 1 simultaneously.

[3] Receive error processing:
If a receive error occurs, read t
ORER flag in SSR, and after
performing the appropriate erro
processing, clear the ORER fla
Transmission/reception cannot
resumed if the ORER flag is se

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

[5] Serial transmission/reception
continuation procedure:
To continue serial transmission
reception, before the MSB (bit
the current frame is received, f
reading the RDRF flag, reading
and clearing the RDRF flag to
before the MSB (bit 7) of the cu
frame is transmitted, read 1 fro
TDRE flag to confirm that writin
possible. Then write data to TD
clear the TDRE flag to 0.
However, the TDRE flag is che
and cleared automatically whe
DTC is initiated by a transmit d
empty interrupt (TXI) request a
writes data to TDR. Similarly,
RDRF flag is cleared automati
when the DTC is initiated by a
data full interrupt (RXI) and rea
data from RDR.

Figure 13.20 Sample Flowchart of Simultaneous Serial Transmission and Rec

function as an I/O pin. Pull up the data transmission line to Vcc using a resistor. Setting th
and TE bits in SCR to 1 with the IC card not connected enables closed transmission/recep
allowing self diagnosis. To supply the IC card with the clock pulses generated by the SCI
the SCK pin output to the CLK pin of the IC card. A reset signal can be supplied via the c
port of this LSI.

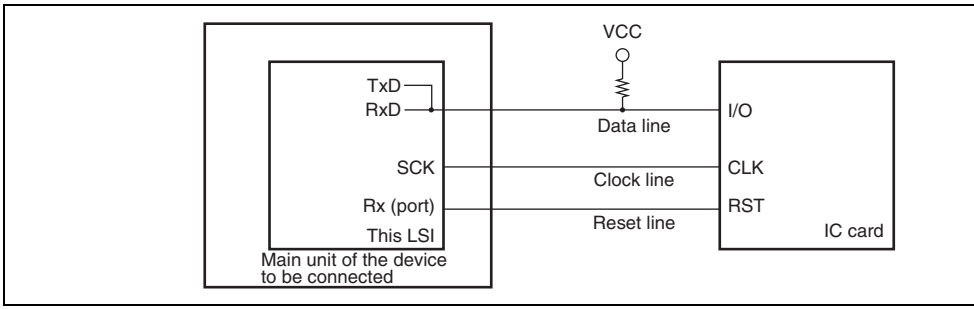


Figure 13.21 Pin Connection for Smart Card Interface

13.7.2 Data Format (Except in Block Transfer Mode)

Figure 13.22 shows the data transfer formats in smart card interface mode.

- One frame contains 8-bit data and a parity bit in asynchronous mode.
- During transmission, at least 2 etu (elementary time unit: time required for transferrin
is secured as a guard time after the end of the parity bit before the start of the next fra
- If a parity error is detected during reception, a low error signal is output for 1 etu after
has passed from the start bit.
- If an error signal is sampled during transmission, the same data is automatically re-tra
after two or more etu.

Figure 13.22 Data Formats in Normal Smart Card Interface Mode

For communication with the IC cards of the direct convention and inverse convention ty
follow the procedure below.

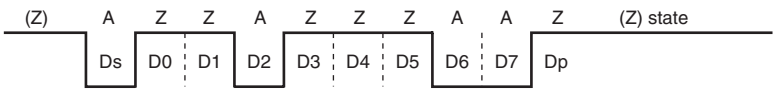


Figure 13.23 Direct Convention (SDIR = SINV = O/$\overline{E}$ = 0)

For the direct convention type, logic levels 1 and 0 correspond to states Z and A, respect
data is transferred with LSB-first as the start character, as shown in figure 13.23. Theref
in the start character in the figure is H'3B. When using the direct convention type, write
the SDIR and SINV bits in SCMR. Write 0 to the O/$\overline{E}$ bit in SMR in order to use even p
which is prescribed by the smart card standard.

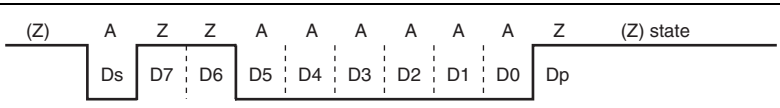


Figure 13.24 Inverse Convention (SDIR = SINV = O/$\overline{E}$ = 1)

- If a parity error is detected during reception, no error signal is output. Since the PER b
  SSR is set by error detection, clear the bit before receiving the parity bit of the next fr
- During transmission, at least 1 etu is secured as a guard time after the end of the parity
  before the start of the next frame.
- Since the same data is not re-transmitted during transmission, the TEND flag in SSR i
  11.5 etu after transmission start.
- Although the ERS flag in block transfer mode displays the error signal status as in nor
  smart card interface mode, the flag is always read as 0 because no error signal is trans

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

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.

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

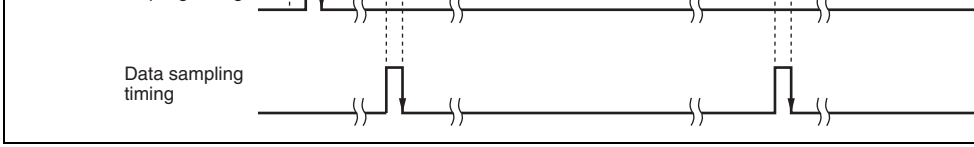


**Figure 13.25 Receive Data Sampling Timing in Smart Card Interface Mod
(When Clock Frequency is 372 Times the Bit Rate)**

13.7.5 Initialization

Before starting transmitting and receiving data, initialize the SCI using the following 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.

re-transmitted. Figure 13.26 shows the data re-transfer operation during transmission.

1. If an error signal from the receiving end is sampled after one frame of data has been transmitted, the ERS bit in SSR is set to 1. Here, an ERI interrupt request is generate RIE bit in SCR is set to 1. Clear the ERS bit to 0 before the next parity bit is sample
2. For the frame in which an error signal is received, the TEND bit in SSR is not set to 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 13.28 shows a sample flowchart for transmission. All the processing steps are automatically performed using a TXI interrupt request to activate the DTC. In transmiss TEND and TDRE flags in SSR are simultaneously set to 1, thus generating a TXI interr 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. See section 7, Data Transfer Controller (DTC) for DTC settings.

Figure 13.26 Data Re-transfer Operation in SCI Transmission Mode

Note that the TEND flag is set in different timings depending on the GM bit setting in SM
which is shown in figure 13.27.

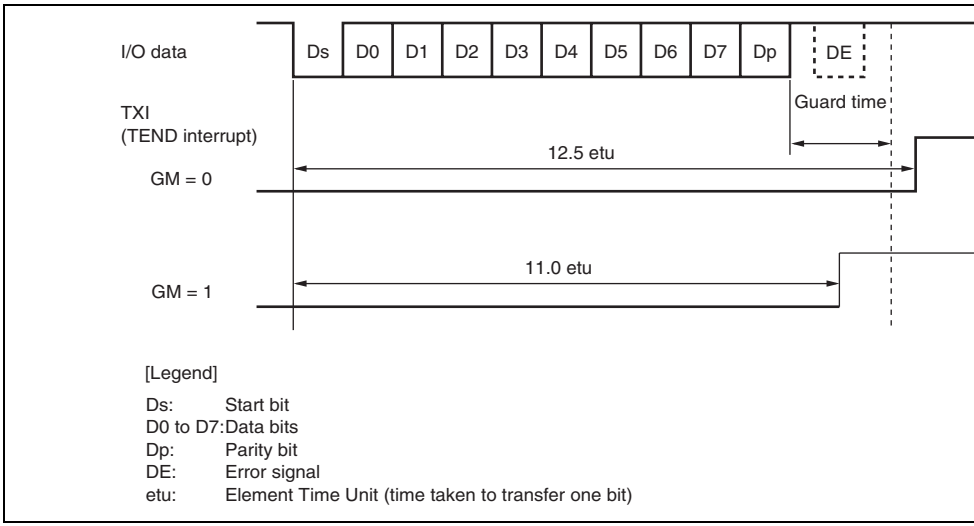


Figure 13.27 TEND Flag Set Timings during Transmission

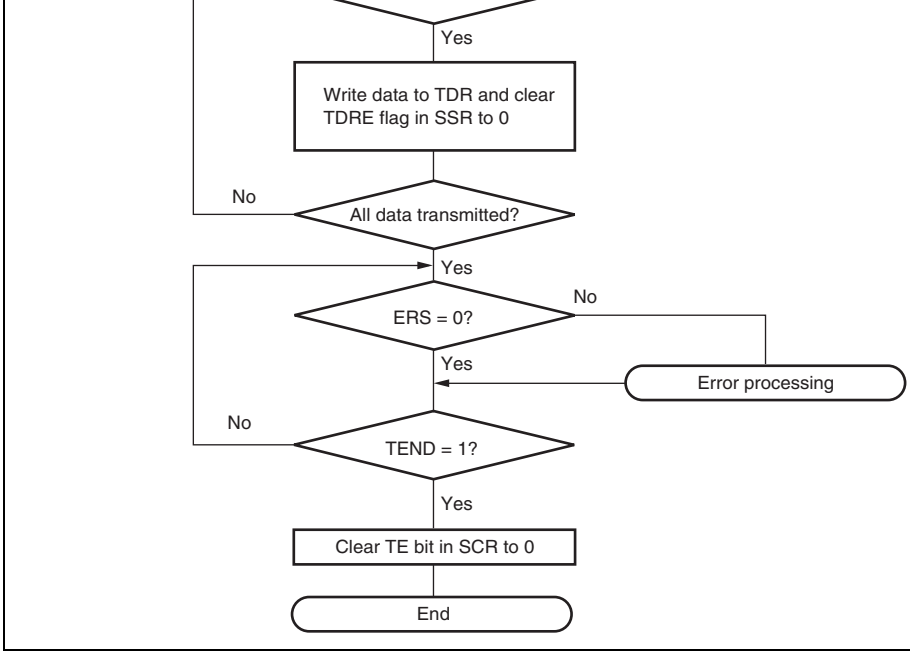


Figure 13.28 Sample Transmission Flowchart

to have been received successfully, and the RDRF bit in SSR is set to 1. Here, an RXI request is generated if the RIE bit in SCR is set.

Figure 13.30 shows a sample flowchart for reception. All the processing steps are 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 13.4, Operation in Asynchronc

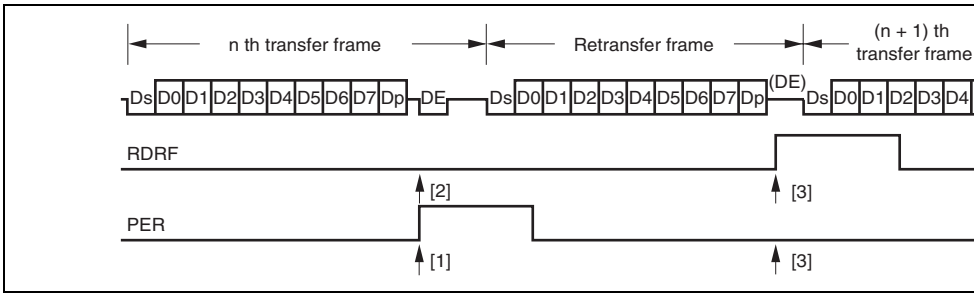


Figure 13.29 Data Re-transfer Operation in SCI Reception Mode

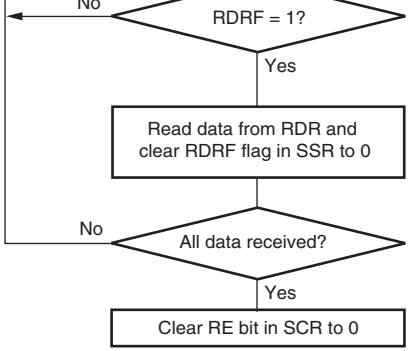


Figure 13.30 Sample Reception Flowchart

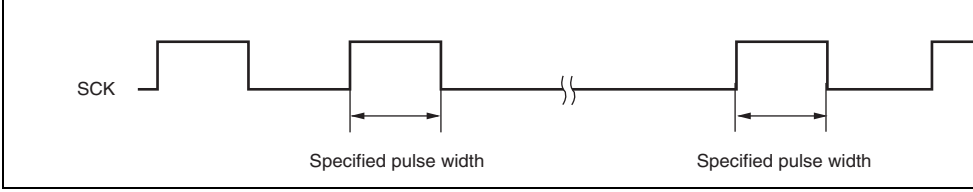


Figure 13.31 Clock Output Fixing Timing

At power-on and transitions to/from software standby mode, use the following procedure the appropriate clock duty ratio.

At Power-On:

To secure the appropriate clock duty ratio simultaneously with power-on, use the followi procedure.

1. Initially, port input is enabled in the high-impedance state. To fix the potential level, u pull-up or pull-down resistor.
2. Fix the SCK pin to the specified output using the CKE1 bit in SCR.
3. Set SMR and SCMR to enable smart card interface mode.
4. Set the CKE0 bit in SCR to 1 to start clock output.

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

1. Cancel software standby mode.
2. Write 1 to the CKE0 bit in SCR to start clock output. A clock signal with the approp ratio is then generated.

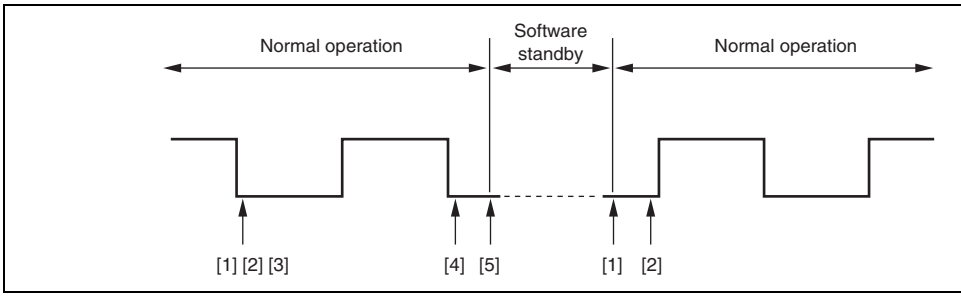


Figure 13.32 Clock Stop and Restart Procedure

allow data transfer. The TDRE flag is automatically cleared to 0 at data transfer by the D

When the RDRF flag in SSR is set to 1, an RXI interrupt request is generated. When the
PER, or FER flag in SSR is set to 1, an ERI interrupt request is generated. An RXI 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 13.12 SCI Interrupt Sources

Channel	Name	Interrupt Source	Interrupt Flag	DTC Activation
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
3	ERI3	Receive error	ORER, FER, PER	Not possible
	RXI3	Receive data full	RDRF	Possible
	TXI3	Transmit data empty	TDRE	Possible
	TEI3	Transmit end	TEND	Not possible

	TXI1	Transmit data empty	TEND	Possible
3	ERI3	Receive error, error signal detection	ORER, PER, ERS	Not possible
	RXI3	Receive data full	RDRF	Possible
	TXI3	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.

When framing error detection is performed, a break can be detected by reading the RxD p
directly. In a break, the input from the RxD pin becomes all 0s, and so the FER flag in SS
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.

13.9.3 Mark State and Break Sending

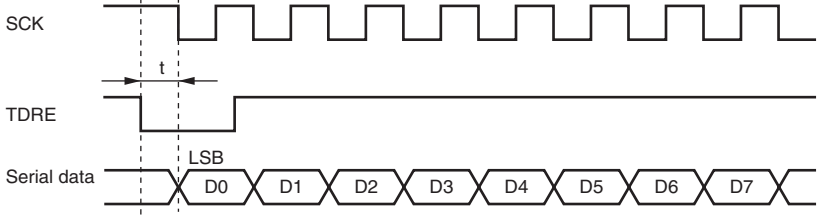
When the TE bit in SCR is 0, the TxD pin is used as an I/O port whose direction (input or
and level are determined by DR and DDR of the port. This can be used to set the TxD pin
state (high level) or send a break during serial data transmission. To maintain the commu
line at mark state until TE is set 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.

13.9.4 Receive Error Flags and Transmit Operations (Clock Synchronous Mode

Transmission cannot be started when a receive error flag (ORER, FER, or RER) in SSR 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.

13.9.5 Relation between Writing to TDR and TDRE Flag

Data can be written to TDR irrespective of the TDRE flag status in SSR. However, if the
is written to TDR when the TDRE flag is 0, that is, when the previous data has not been
transferred to TSR yet, the previous data in TDR is lost. Be sure to write transmit data to
after verifying that the TDRE flag is set to 1.



Note: When external clock is supplied, t must be more than four clock cycles.

Figure 13.33 Sample Transmission using DTC in Clock Synchronous Mod

Figure 13.34 shows a sample flowchart for mode transition during transmission. Figures
13.36 show the pin states during transmission.

Before making the transition from the transmission mode using DTC transfer to module s
software standby mode, stop all transmit operations (TE = TIE = TEIE = 0). Setting TE a
1 after mode cancellation generates a TXI interrupt request to start transmission using the

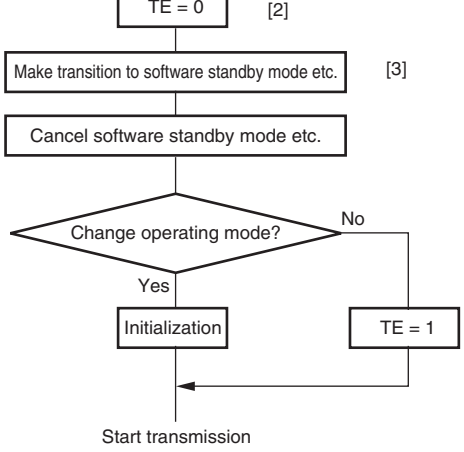


Figure 13.34 Sample Flowchart for Mode Transition during Transmissio

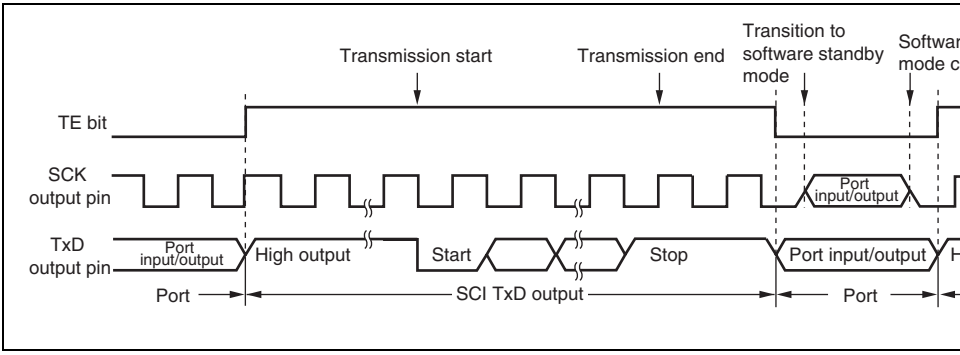


Figure 13.35 Pin States during Transmission in Asynchronous Mode (Internal Clock)

Figure 13.36 Pin States during Transmission in Clock Synchronous Mode (Internal Clock)

Reception: Before making the transition to module stop or software standby mode, stop r (RE = 0). RSR, RDR, and SSR are reset. If transition is made during data reception, the d received will be invalid.

To receive data in the same reception mode after mode cancellation, set RE to 1, and then reception. To receive data in a different reception mode, initialize the SCI first.

Figure 13.37 shows a sample flowchart for mode transition during reception.

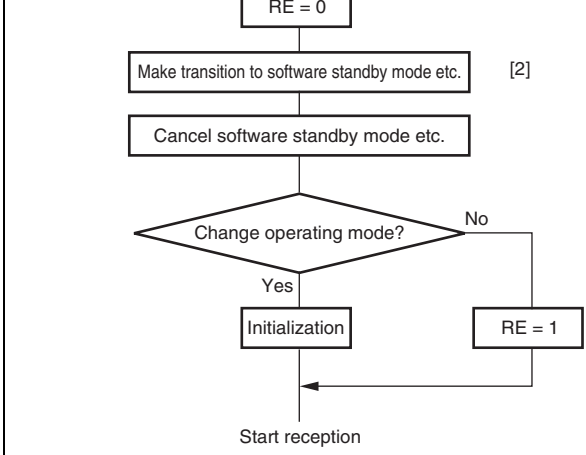


Figure 13.37 Sample Flowchart for Mode Transition during Reception

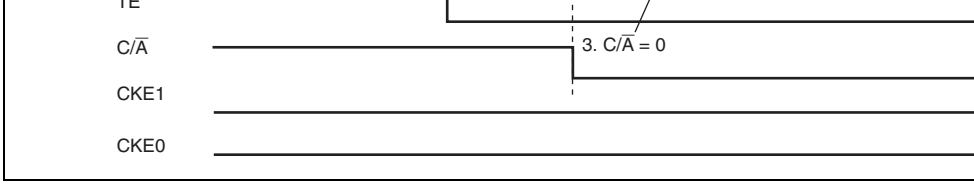


Figure 13.38 Switching from SCK Pins to Port Pins

To prevent the low pulse output that is generated when switching the SCK pins to the por specify the SCK pins for input (pull up the SCK/port pins externally), and follow the proc below with DDR = 1, DR = 1, C/$\overline{A}$ = 1, CKE1 = 0, CKE1 = 0, and TE = 1.

1. End serial data transmission
2. TE bit = 0
3. CKE1 bit = 1
4. C/$\overline{A}$ bit = 0 (switch to port output)
5. CKE1 bit = 0

Figure 13.39 Prevention of Low Pulse Output at Switching from SCK Pins to P

- CRC code generated for any desired data length in an 8-bit unit
- CRC operation executed on eight bits in parallel
- One of three generating polynomials selectable
- CRC code generation for LSB-first or MSB-first communication selectable

Figure 14.1 is a block diagram of the CRC operation circuit.

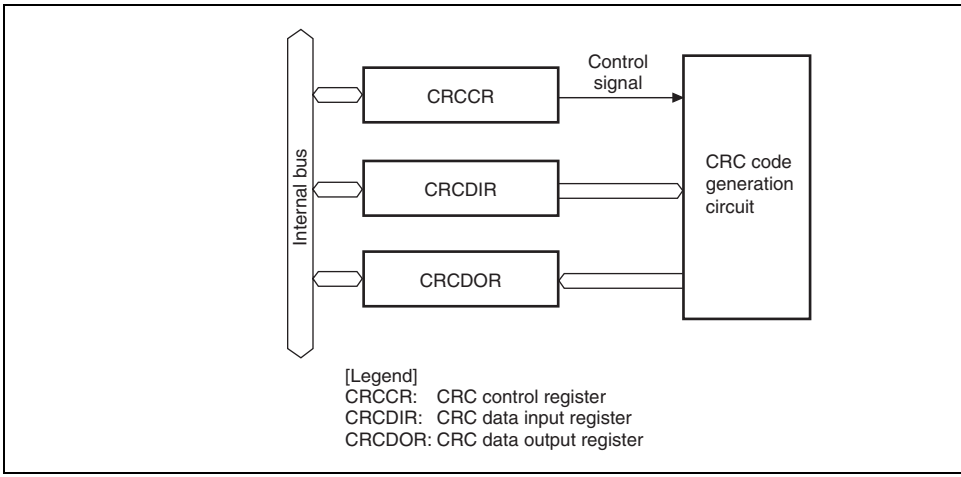


Figure 14.1 Block Diagram of CRC Operation Circuit

CRCCR initializes the CRC operation circuit, switches the operation mode, and selects th
generating polynomial.

Bit	Bit Name	Initial Value	R/W	Description
7	DORCLR	0	W	CRCDOR Clear Setting this bit to 1 clears CRCDOR to H′0000.
6 to 3	—	All 0	R	Reserved The initial value should not be changed.
2	LMS	0	R/W	CRC Operation Switch Selects CRC code generation for LSB-first or MS communication. 0: Performs CRC operation for LSB-first commun The lower byte (bits 7 to 0) is first transmitted w CRCDOR contents (CRC code) are divided int bytes to be transmitted in two parts. 1: Performs CRC operation for MSB-first commu The upper byte (bits 15 to 8) is first transmitted CRCDOR contents (CRC code) are divided int bytes to be transmitted in two parts.
1	G1	0	R/W	CRC Generating Polynomial Select
0	G0	0	R/W	These bits select the polynomial. 00: Reserved 01: $X^8 + X^2 + X + 1$ 10: $X^{16} + X^{15} + X^2 + 1$ 11: $X^{16} + X^{12} + X^5 + 1$

CRCCR are set to G1 = 0 and G0 = 1, respectively, the lower byte of this register contai
result.

14.3 CRC Operation Circuit Operation

The CRC operation circuit generates a CRC code for LSB-first/MSB-first communicatio
example in which a CRC code for hexadecimal data H'F0 is generated using the $X^{16} + X$
polynomial with the G1 and G0 bits in CRCCR set to B'11 is shown below.

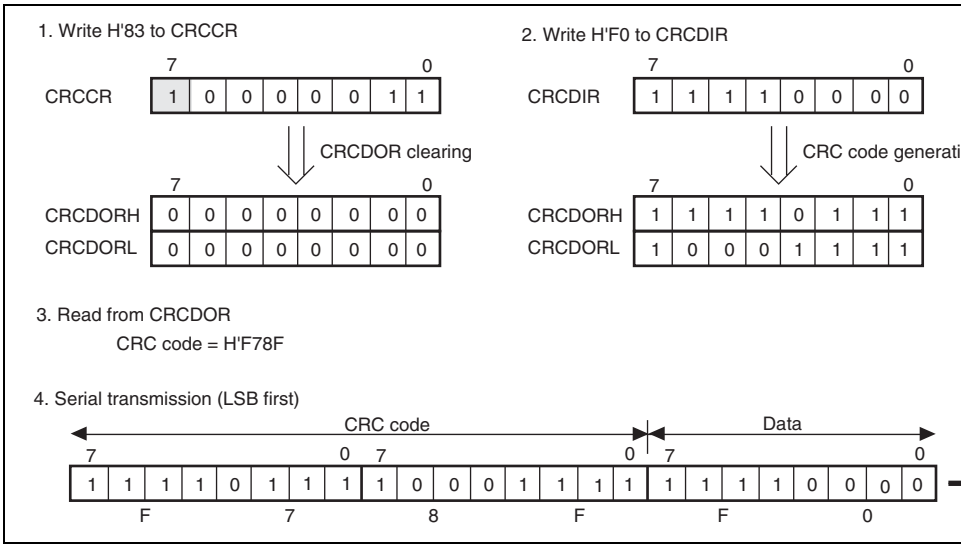


Figure 14.2 LSB-First Data Transmission

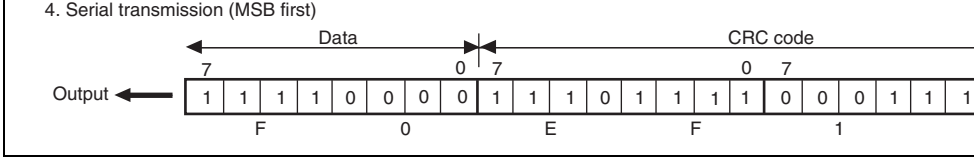


Figure 14.3 MSB-First Data Transmission

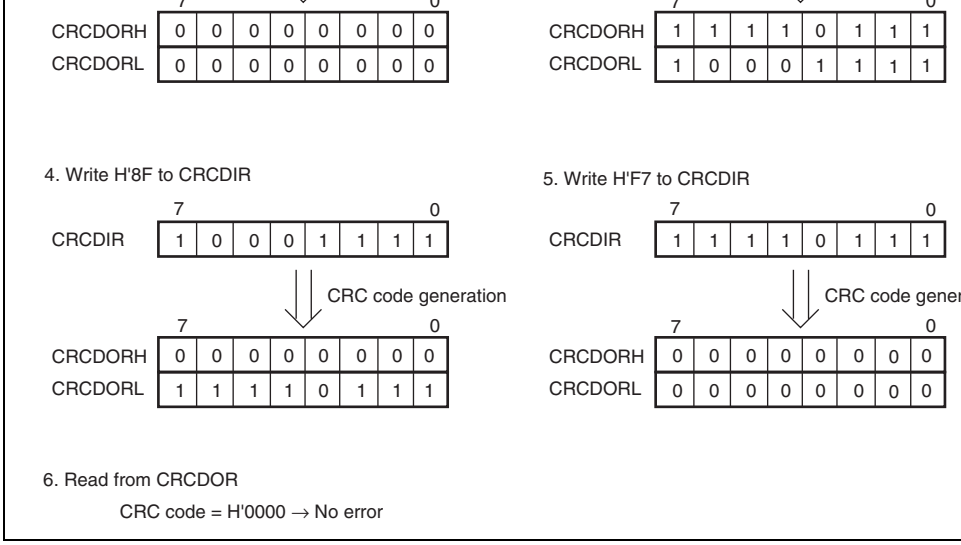


Figure 14.4 LSB-First Data Reception

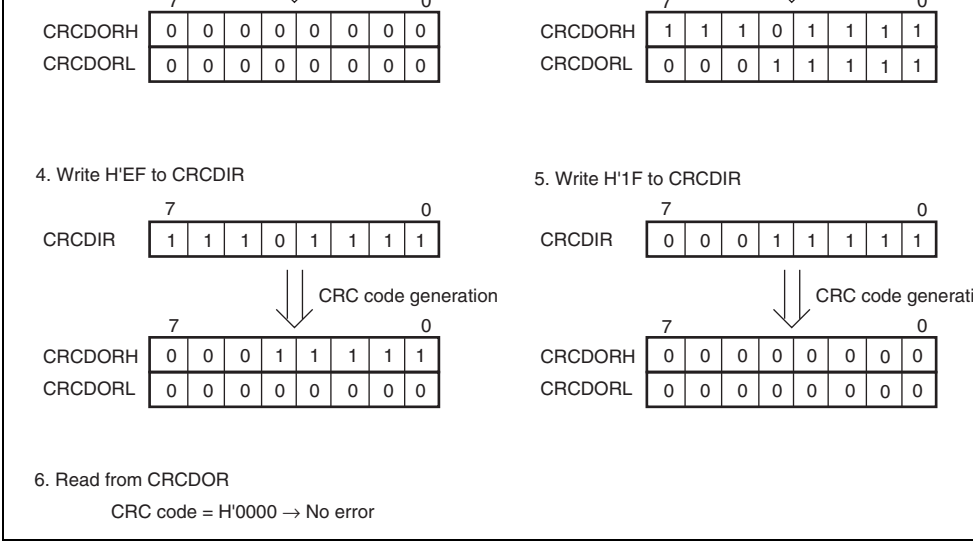


Figure 14.5 MSB-First Data Reception

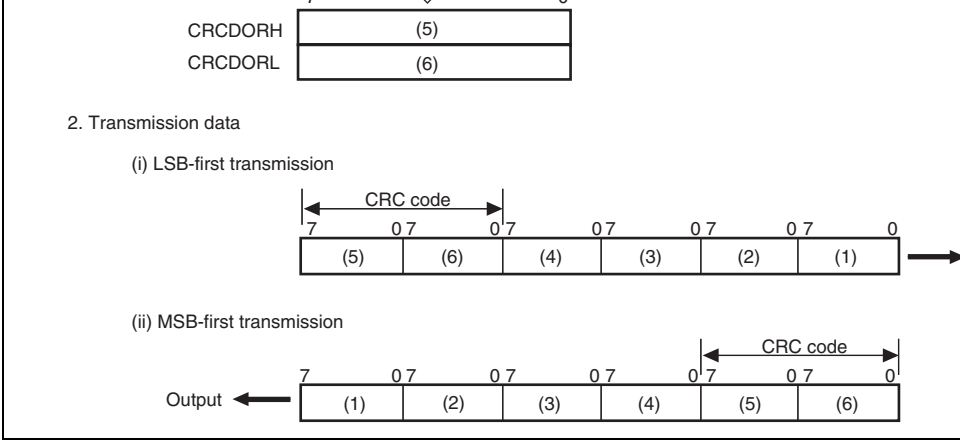


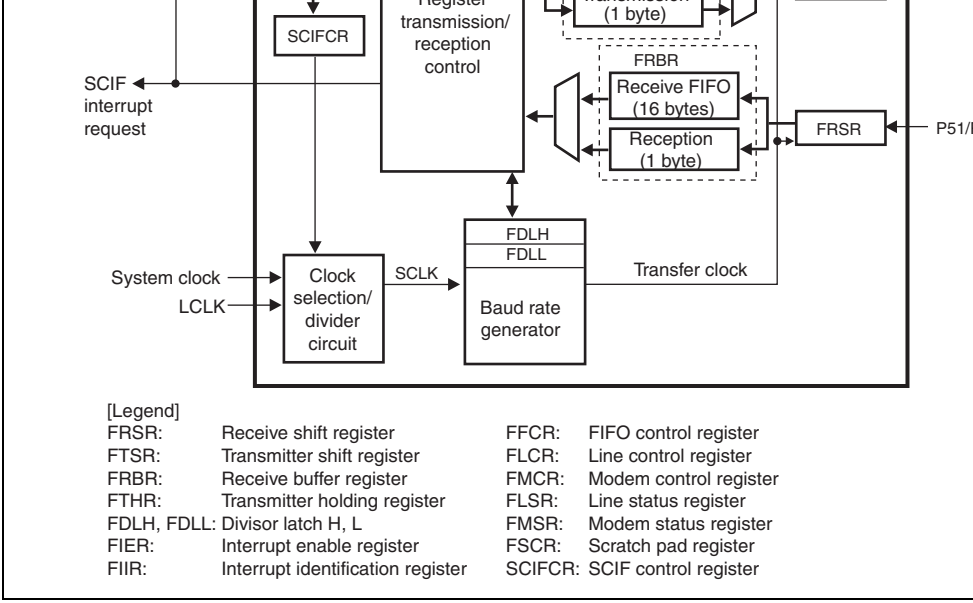
Figure 14.6 LSB-First and MSB-First Transmit Data

In addition, the SCIF can be connected to the LPC interface for direct control from the L

15.1 Features

- Full-duplex communication:

  The transmitter and receiver are independent, enabling transmission and reception to executed simultaneously. Both the transmitter and receiver use 16-stage FIFO buffer enabling continuous transmission and continuous reception of serial data.
- On-chip baud rate generator allows any bit rate to be selected
- Modem control function (only for $\overline{\text{CTS}}$ and $\overline{\text{RTS}}$)
- Data length: Selectable from 5, 6, 7, and 8 bits
- Parity: Selectable from even parity, odd parity, and no parity
- Stop bit length: Selectable from 1, 1.5, and 2 bits
- Receive error detection: Parity, overrun, and framing errors
- Break detection



[Legend]

FRSR:	Receive shift register	FFCR:	FIFO control register
FTSR:	Transmitter shift register	FLCR:	Line control register
FRBR:	Receive buffer register	FMCR:	Modem control register
FTHR:	Transmitter holding register	FLSR:	Line status register
FDLH, FDLL:	Divisor latch H, L	FMSR:	Modem status register
FIER:	Interrupt enable register	FSCR:	Scratch pad register
FIIR:	Interrupt identification register	SCIFCR:	SCIF control register

Figure 15.1 Block Diagram of SCIF

RTS	P65	Output	Transmission request output

- Divisor latch L (FDLL)
- Interrupt enable register (FIER)
- Divisor latch H (FDLH)
- Interrupt identification register (FIIR)
- FIFO control register (FFCR)
- Line control register (FLCR)
- Modem control register (FMCR)
- Line status register (FLSR)
- Modem status register (FMSR)
- Scratch pad register (FSCR)
- SCIF control register (SCIFCR)
- SCIF address register H (SCIFADRH)
- SCIF address register L (SCIFADRL)
- SERIRQ control register 4 (SIRQCR4)

Table 15.2 Register Access

SCIFE Bit in HICR5	**0**		**1**	
Bit 3 in SUBMSTPBL	**0**	**1**	**0**	**1**
SCIFCR	H8S CPU access*2	Access disabled	H8S CPU access*2	Access d
Other than SCIFCR	H8S CPU access*2	Access disabled	LPC access*1	LPC acc

Notes: 1. When LPC access is set, writing from the H8S CPU is disabled. The read valu
2. When H8S CPU access is set, writing from the LPC is disabled. The read valu

the DR bit in FLSR is set.

When the FIFO is disabled, the data in FRBR must be read before the next data is receiv
data is received before the remaining data is read, the data is overwritten, resulting in an
error.

When this register is read with the FIFO enabled, the first buffer of the receive FIFO is r
When the receive FIFO becomes full, the subsequent receive data is lost, resulting in an
error.

Bit	Bit Name	Initial Value	R/W	Description
7 to 0	Bit 7 to bit 0	All 0	R	Stores received serial data. The data is 16 bytes when the FIFO is ena

15.3.3 Transmitter Shift Register (FTSR)

FTSR is a register that converts parallel data from the TxDF pin to serial data and then t
the serial data. When one frame transmission of serial data is completed, the next data is
transferred from FTHR. The serial data is transmitted from the LSB (bit 0).

FTSR cannot be written from the H8S CPU/LPC interface.

Bit	Bit Name	Initial Value	R/W	Description
7 to 0	Bit 7 to bit 0	—	W	Stores serial data to be transmitted. The data is 16 bytes when the FIFO is ena

15.3.5 Divisor Latch H, L (FDLH, FDLL)

The FDLH and FDLL are registers used to set the baud rate. They are accessible when th bit in FLCR is 1. Frequency division ranging from 1 to ($2^{16} - 1$) can be set with these regi The frequency divider circuit stops when both of FDLH and FDLL are 0 (initial value).

- FDLH

Bit	Bit Name	Initial Value	R/W	Description
7 to 0	Bit 7 to bit 0	All 0	R/W	Upper 8 bits of divisor latch

- FDLL

Bit	Bit Name	Initial Value	R/W	Description
7 to 0	Bit 7 to bit 0	All 0	R/W	Lower 8 bits of divisor latch

Baud rate = (Clock frequency input to baud rate generator) / (16 × divisor value)

				0: Modem status interrupt disabled 1: Modem status interrupt enabled
2	ELSI	0	R/W	Receive Line Status Interrupt Enable 0: Receive line status interrupt disabled 1: Receive line status interrupt enabled
1	ETBEI	0	R/W	FTHR Empty Interrupt Enable 0: FTHR empty interrupt disabled 1: FTHR empty interrupt enabled
0	ERBFI	0	R/W	Receive Data Ready Interrupt Enable A character timeout interrupt is included w FIFO is enabled. 0: Receive data ready interrupt disabled 1: Receive data ready interrupt enabled

5, 4	—	All 0	R	Reserved These bits are always read as 0. The initial should not be changed.
3	INTID2	0	R	Interrupt ID2, ID1, ID0
2	INTID1	0	R	These bits Indicate the interrupt of the high
1	INTID0	0	R	priority among the pending interrupts. 000: Modem status 001: FTHR empty 010: Receive data ready 011: Receive line status 110: Character timeout (when the FIFO is e
0	INTPEND	1	R	Interrupt Pending Indicates whether one or more interrupts a pending. 0: Interrupt pending 1: No interrupt pending

						remaining, FIFO trigger level	receive below level.
1	1	0	0	2	Character timeout (with FIFO enabled)	No data is input to or output from the receive FIFO for the 4-character time period while one or more characters remain in the receive FIFO.	FRBR
0	0	1	0	3	FTHR empty	FTHR empty	FIIR re FTHR
0	0	0	0	4 (low)	Modem status	CTS	FMSR

				10: 8 bytes 11: 14 bytes
5, 4	—	—	—	Reserved These bits cannot be modified.
3	DMAMODE	0	—	DMA Mode This bit is not supported. The initial value shou be changed.
2	XMITFRST	0	W	Transmit FIFO Reset The transmit FIFO data is cleared when 1 is w However, FTSR data is not cleared. This bit is automatically cleared.
1	RCVRFRST	0	W	Receive FIFO Reset The receive FIFO data is cleared when 1 is wr However, FRSR data is not cleared. This bit is automatically cleared.
0	FIFOE	0	W	FIFO Enable 0: Transmit/receive FIFOs disabled All bytes of these FIFOs are cleared. 1: Transmit/receive FIFOs enabled

				1: FDLL and FDLH access enabled
6	BREAK	0	R/W	Break Control Generates a break by driving the serial outpu TxDF low. The break state is released by clearing this b 0: Break released 1: Break generated
5	STICK PARITY	0	R	Stick Parity This bit is not supported in this LSI. This bit is always read as 0. The initial value not be changed.
4	EPS	0	R/W	Parity Select Selects even or odd parity when the PEN bit 0: Odd parity 1: Even parity
3	PEN	0	R/W	Parity Enable Selects whether to add a parity bit for data transmission and whether to perform a parity data reception. 0: No parity bit added/parity check disabled 1: Parity bit added/parity check enabled

				length. 00: Data length is 5 bits 01: Data length is 6 bits 10: Data length is 7 bits 11: Data length is 8 bits

15.3.10 Modem Control Register (FMCR)

FMCR controls output signals.

Bit	Bit Name	Initial Value	R/W	Description
7 to 5	—	All 0	R	Reserved These bits are always read as 0. The initial va should not be changed.
4	LOOP BACK	0	R/W	Loopback Test The transmit data output is internally connect receive data input, and the transmit data outp (RxDF) becomes 1. The receive data input pi disconnected from external sources. The mod control input pin, $\overline{\text{CTS}}$, is disconnected from t external sources, and the pin is internally con to the modem control output signal, $\overline{\text{RTS}}$. The transmit data is received immediately in loopb mode. Enabling/disabling of interrupts is set b OUT2LOOP bit in SCIFCR and FIER. 0: Loopback function disabled 1: Loopback function enabled

1	RTS	0	R/W	Request to Send Controls the $\overline{\text{RTS}}$ output. 0: $\overline{\text{RTS}}$ output is high level 1: $\overline{\text{RTS}}$ output is low level
0	—	—	—	Reserved

				[Clearing condition] When FRBR is read or FLSR is read while tl no remaining data that could cause an error FIFO clear. 1: A receive FIFO error [Setting condition] When at least one data error (parity error, fra error, or break interrupt) has occurred in the
6	TEMT	1	R	Transmitter Empty Indicates whether transmit data remains. • When the FIFO is disabled 0: Transmit data remains in FTHR or FTSR. [Clearing condition] Transmit data is written to FTHR. 1: No transmit data remains in FTHR and FT [Setting condition] When no transmit data remains in FTHR and • When the FIFO is enabled 0: Transmit data remains in the transmit FIF FTSR. [Clearing condition] Transmit data is written to FTHR. 1: No transmit data remains in the transmit F FTSR. [Setting condition] When no transmit data remains in the transm and FTSR

				[Setting condition] When the transmit FIFO becomes empty • When the FIFO is disabled 0: Transmit data remains in FTHR. [Clearing condition] Transmit data is written to FTHR 1: No transmit data in FTHR [Setting condition] When data transfer from FTHR to FTSR is completed
4	BI	0	R	Break Interrupt Indicates detection of the receive data brea When the FIFO is enabled, a break interrup in any receive data in the FIFO, and this bi when the receive data is in the first FIFO b Reception of the next data starts after the i receive data becomes mark and a valid sta received. 0: Break signal not detected [Clearing condition] FLSR read 1: Break signal detected [Setting condition] When input receive data stays at space (lo for a reception time exceeding the length o frame

				[Clearing condition] FLSR read 1: A framing error [Setting condition] Invalid stop bit in the receive data
2	PE	0	R	Parity Error This bit indicates a parity error in the receive when the PEN bit in FLCR is 1. When the FI enabled, this error occurs in any receive dat FIFO, and this bit is set when the receive da the first FIFO buffer. 0: No parity error [Clearing condition] FLSR read If this bit is set during an overrun error, read twice. 1: A parity error [Setting condition] Detection of parity error in receive data

				lost. 0: No overrun error [Clearing condition] FLSR read 1: An overrun error [Setting condition] Occurrence of an overrun error
0	DR	0	R	Data Ready Indicates that receive data is stored in FRB FIFO. 0: No receive data [Clearing condition] FRBR is read or all of the FIFO data is rea 1: Receive data remains. [Setting condition] Reception of data

				Indicates a change in the $\overline{\text{CTS}}$ input signal a DCTS bit is read. 0: No change in the $\overline{\text{CTS}}$ input signal after F read [Clearing condition] FMSR read 1: A change in the $\overline{\text{CTS}}$ input signal after FM [Setting condition] A change in the $\overline{\text{CTS}}$ input signal

15.3.13 Scratch Pad Register (FSCR)

FSCR is not used for SCIF control, but is used to temporarily store program data.

Bit	Bit Name	Initial Value	R/W	Description
7 to 0	Bit 7 to bit 0	All 0	R/W	Temporarily stores program data.

				Do not change the initial value.
4	OUT2LOOP	0	R/W	Enables or disables interrupts during a loop test. 0: Interrupt enabled 1: Interrupt disabled
3 2	CKSEL1 CKSEL0	0 0	R/W R/W	These bits select the clock (SCLK) to be in baud rate generator. 00: LCLK divided by 18 01: System clock divided by 11 10: Reserved for LCLK (not selectable) 11: Reserved for system clock (not selecta
1	SCIFRST	0	R/W	Resets the baud rate generator, FRSR, an 0: Normal operation 1: Reset
0	REGRST	0	R/W	Resets registers (except SCIFCR) accessil the H8S CPU or LPC interface. 0: Normal operation 1: Reset

CKSEL1, CKSEL0	LSLR (33 MHz) divided by 18		System Clock (34 MHz) divided by 1	
Baud rate	**FDLH, FDLL (Hex)**	**Error (%)**	**FDLH, FDLL (Hex)**	**Error (%)**
50	0900	-0.54 %	H'0F18	-0.01%
75	0600	-0.54 %	H'0A10	-0.01%
110	0417	-0.51 %	H'06DC	0.01%
300	0180	-0.54 %	H'0284	-0.01%
600	00C0	-0.54 %	H'0142	-0.01%
1200	0060	-0.54 %	H'00A1	-0.01%
1800	0040	-0.54 %	H'006B	0.30%
2400	0030	-0.54 %	H'0050	0.62%
4800	0018	-0.54 %	H'0028	0.62%
9600	000C	-0.54 %	H'0014	0.62%
14400	0008	-0.54 %	H'000D	—
19200	0006	-0.54 %	H'000A	0.62%
38400	0003	-0.54 %	H'0005	0.62%
57600	0002	-0.54 %	H'0003	—
115200	0001	-0.54 %	H'0002	—

enabling continuous data transmission and reception.

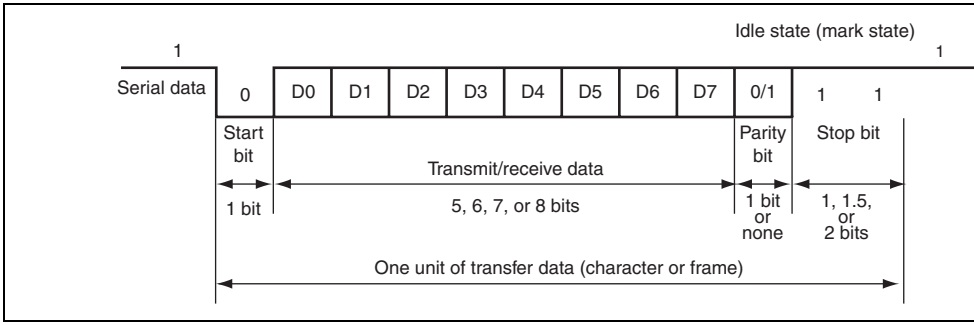


Figure 15.2 Data Format in Serial Transmission/Reception (Example with 8-Bit Data, Parity and 2 Stop Bits)

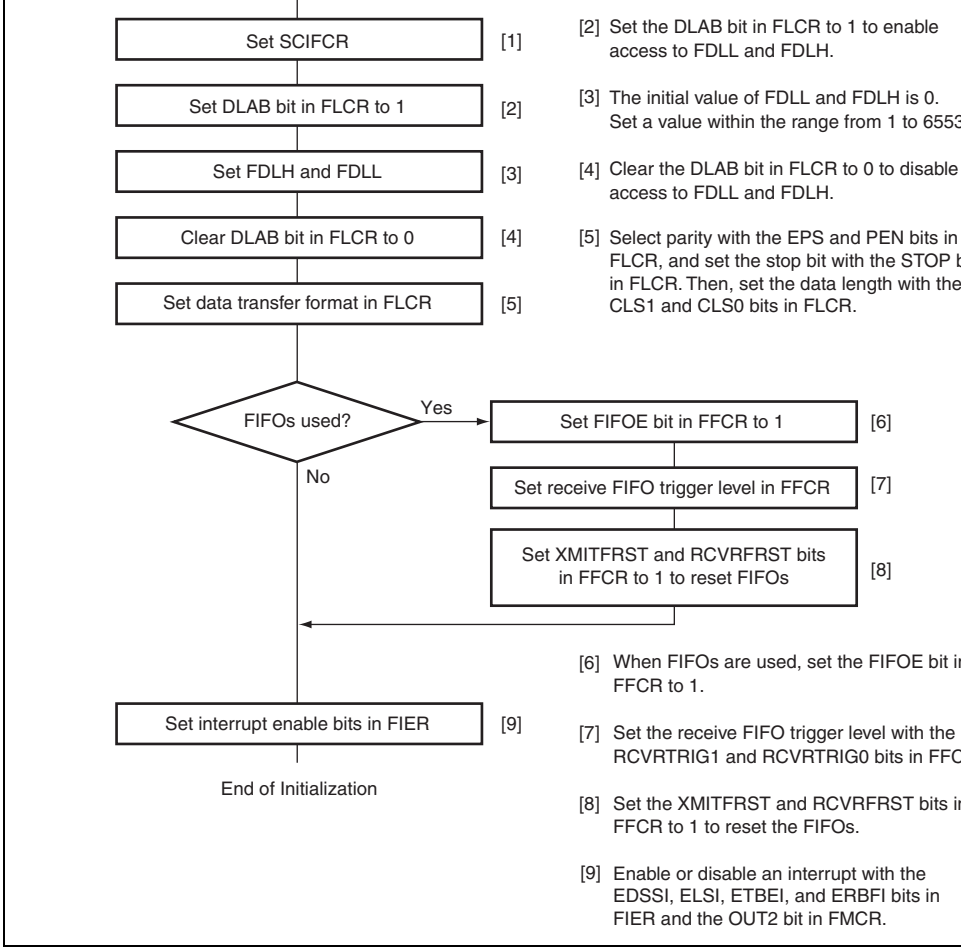
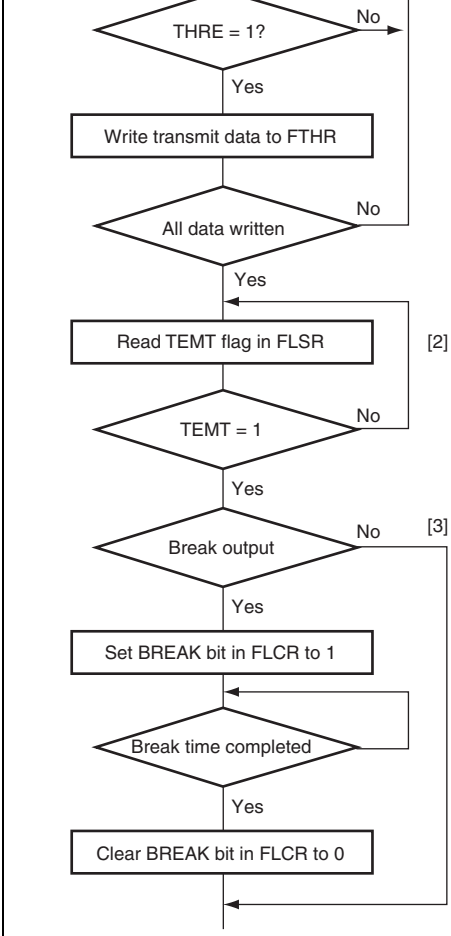
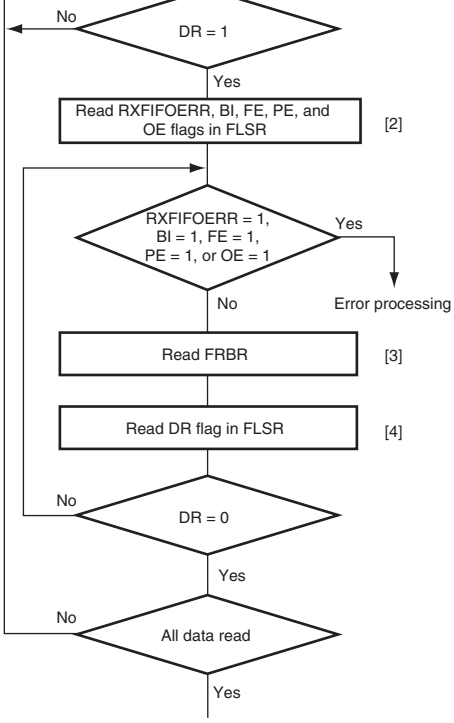


Figure 15.3 Example of Initialization Flowchart



[3] To output a break at the end of serial transmission, s
BREAK bit in FLCR to 1. After completion of the bre
clear the BREAK bit in FLCR to 0 to clear the break

Figure 15.4 Example of Data Transmission Flowchart

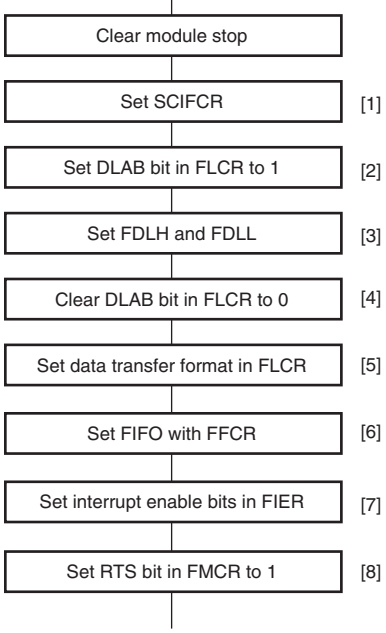


set to 1, a receive line status interrupt occurs.

[3] Read the receive data in FRBR.

[4] Check the DR flag in FLSR. When the DR flag is cleared to 0 and all data has been read, data recep is complete.

Figure 15.5 Example of Data Reception Flowchart



bits in SCIFCR. Set the SCIF input/output pins wi
the SCIFOE1 and SCIFOE0 bits in SCIFCR.

[2] Set the DLAB bit in FLCR to 1 to enable access to
FDLL and FDLH.

[3] The initial value of FDLL and FDLH is 0. Set a val
within the range from 1 to 65535.

[4] Clear the DLAB bit in FLCR to 0 to disable access
FDLL and FDLH.

[5] Select parity with the EPS and PEN bits in FLCR,
set the stop bit with the STOP bit in FLCR. Then,
the data length with the CLS1 and CLS0 bits in FL
Set the FIFOE bit in FFCR to 1 to enable the FIFO

[6] Set the receive FIFO trigger level with the RCVRT
and RCVRTRIG0 bits in FFCR. Select the best tri
level to prevent an overflow of the receive FIFO.

[7] Set the EDSSI and ERBFI bits in FIER to 1 to ena
modem status interrupt and receive data ready int

[8] Set the RTS bit in FMCR to 1.

Figure 15.6 Example of Initialization Flowchart

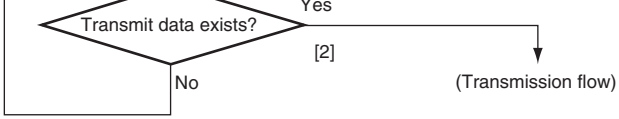


Figure 15.7 Example of Data Transmission/Reception Standby Flowchar

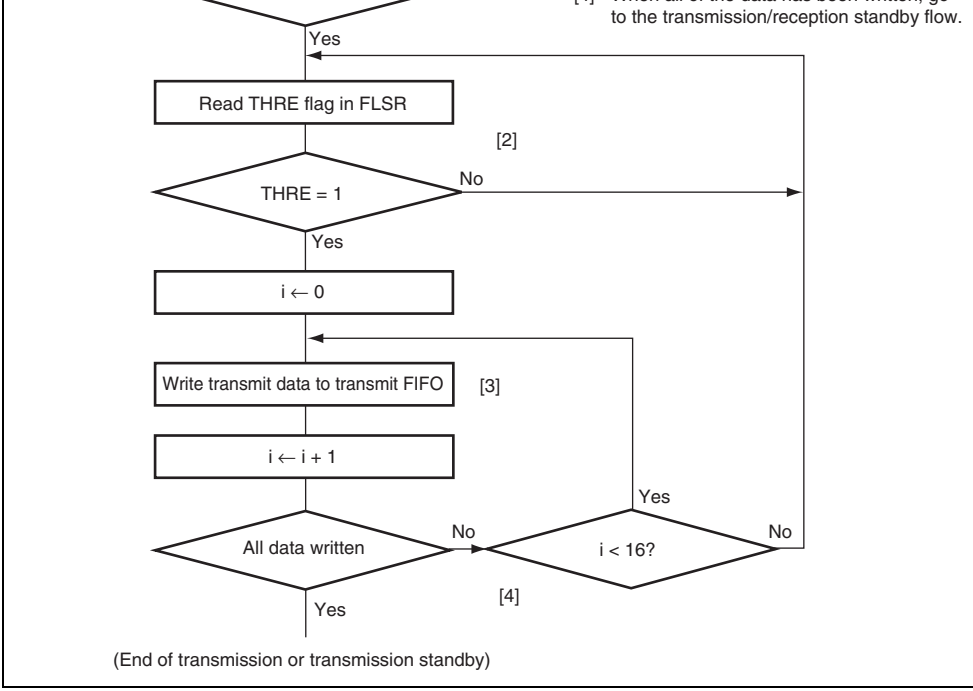
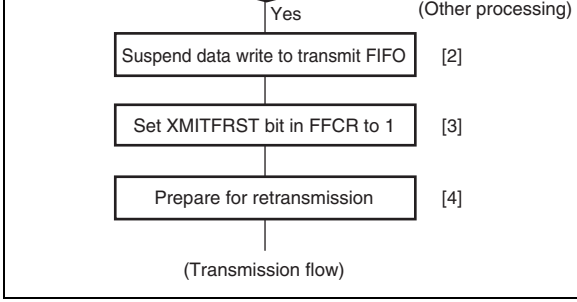
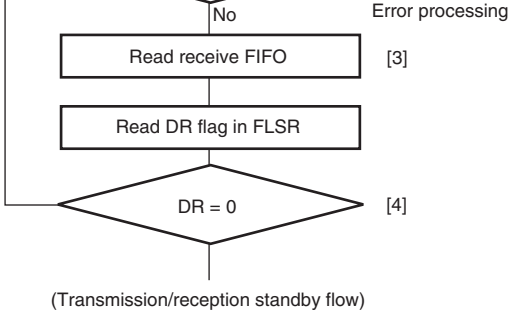


Figure 15.8 Example of Data Transmission Flowchart



[4] Prepare for retransmission of data and go
the transmission flow.

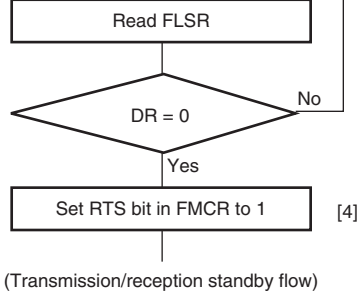
Figure 15.9 Example of Data Transmission Suspension Flowchart



[3] Read the receive FIFO.

[4] Check the DR flag in FLSR. When the DR is cleared and all of the data has been rea reception is complete.

Figure 15.10 Example of Data Reception Flowchart



[4] Set the RTS bit in FMCR to 1, and then go to the transmission/reception standby flow.

Figure 15.11 Example of Data Reception Suspension Flowchart

LPC Interface I/O Address						SCIF
Bits 15 to 3	**Bit 2**	**Bit 1**	**Bit 0**	**R/W**	**Condition**	**Reg**
SCIFADR (bits 15 to 3)	0	0	0	R	FLCR[7] = 0	FRB
				W	FLCR[7] = 0	FTH
				R/W	FLCR[7] = 1	FDL
SCIFADR (bits 15 to 3)	0	0	1	R/W	FLCR[7] = 0	FIER
				R/W	FLCR[7] = 1	FDL
SCIFADR (bits 15 to 3)	0	1	0	R	—	FIIR
				W	—	FFC
SCIFADR (bits 15 to 3)	0	1	1	R/W	—	FLC
SCIFADR (bits 15 to 3)	1	0	0	R/W	—	FMC
SCIFADR (bits 15 to 3)	1	0	1	R	—	FLS
SCIFADR (bits 15 to 3)	1	1	0	R	—	FMS
SCIFADR (bits 15 to 3)	1	1	1	R/W	—	FSC

SCSIRQ3	Initialized	Retained	Retained	Reta
SCSIRQ2	Initialized	Retained	Retained	Reta
SCSIRQ1	Initialized	Retained	Retained	Reta
SCSIRQ0	Initialized	Retained	Retained	Reta

Receive line status	Overrun error, parity error, framing error, break interrupt
Receive data ready	Acceptance of receive data, FIFO trigger level
Character timeout (when FIFO is enabled)	No data is input to or output from the receive FIFO for the 4-character time period while one or more characters remain in the receive FIFO.
FTHR empty	FTHR empty
Modem status	CTS

Table 15.9 shows the interrupt source, vector address, and interrupt priority.

Table 15.9 Interrupt Source, Vector Address, and Interrupt Priority

Interrupt Origin of Interrupt Source	Interrupt Name	Vector Number	Vector Address
SCIF	SCIF	82	H'000148

15.6 Usage Note

15.6.1 Power-Down Mode When LCLK is Selected for SCLK

To switch to software standby mode when LCLK divided by 18 has been selected for SC
the shutdown function of the LPC interface to stop LCLK.

for IPMI applications.

- Five serial pin multiplexed modes
  - Mode 0: Each COM port is used for its respective serial communication module: for SCIF, COM2 for SCI_1 and COM3 for SCI_3 (default mode)
  - Mode 1: COM1 snoop mode with use of SCI_1 and internal registers
  - Mode 2: SCIF-and-SCI_1 bridge mode in which internal registers provide softwa control.
  - Mode 3: COM port switched mode in which COM1 is connected to SCI_1 and C connected to SCIF. Internal registers provide flow control for SCI_1.
  - Mode 4: SCIF-and-SCI_3 bridge mode providing the same functionality as mode

Please refer to section 13, Serial Communication Interface (SCI) for details on SCI_1 ar and section 15, Serial Communication Interface with FIFO (SCIF), for details on SCIF.

RTS	Output	Transmission request	P65

16.3.1 Serial Multiplexed Mode Register 0 (SMR0)

Bit	Bit Name	Initial Value	R/W	Description
7 to 5	—	—	R	Reserved
4	SME	0	R/W	Serial Pin Multiplex Enable 0: Pin multiplexing disabled 1: Pin multiplexing enabled
3	—	0	R	Reserved This bit is always read as 0. The write value always be 0.
2	SM2	0	R/W	Serial Pin Multiplexed Mode Select
1	SM1	0	R/W	These bits select a serial pin multiplexed m selection is only enabled when SME bit is 1
0	SM0	0	R/W	000: Serial multiplexed mode 0 001: Serial multiplexed mode 1 010: Serial multiplexed mode 2 011: Serial multiplexed mode 3 100: Serial multiplexed mode 4 101: Reserved (Do not modify) 110: Reserved (Do not modify) 111: Reserved (Do not modify)

				1: 1 is output
4	CTS3	—	R	Monitors the state of the $\overline{\text{RTS}}$ pin input of the S mode 4.
3	—	—	R	Reserved
2	RTS3	1	R/W	Controls the output on the $\overline{\text{CTS}}$ pin of the SCIF
				0: 0 is output
				1: 1 is output
1,0	—	—	R/W	Reserved

RxD1 and TxD1 of SCI_1 are cross-connected to COM2. RxD3 and TxD3 of SCI_3 are connected to COM3.

Figure 16.1 illustrates the pin connection in serial pin multiplexed mode 0.

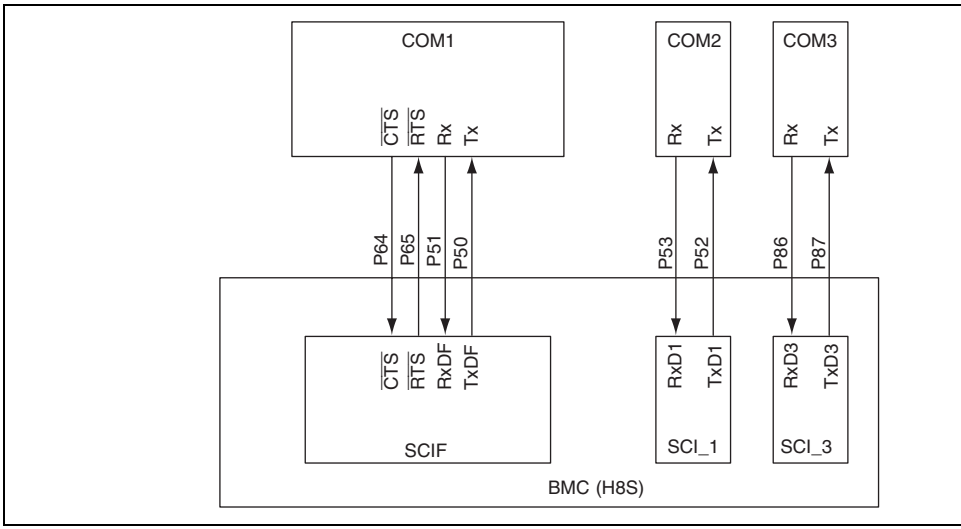


Figure 16.1 Serial Pin Multiplexed Mode 0

Figure 16.2 illustrates the pin connection in serial pin multiplexed mode 1.

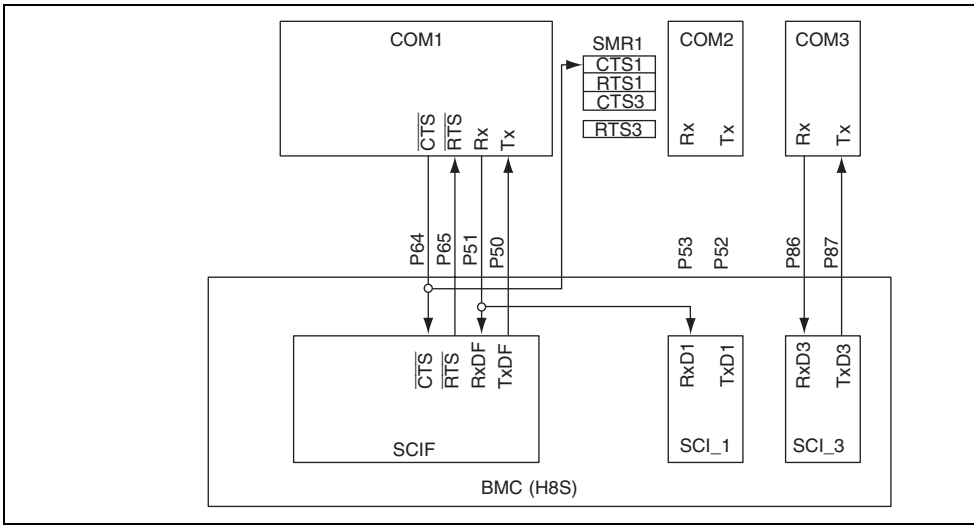


Figure 16.2 Serial Pin Multiplexed Mode 1

state of $\overline{\text{RTS}}$ of SCIF is reflected in bit CTS1 of the SMR1 register.

Figure 16.3 illustrates the pin connection in serial pin multiplexed mode 2.

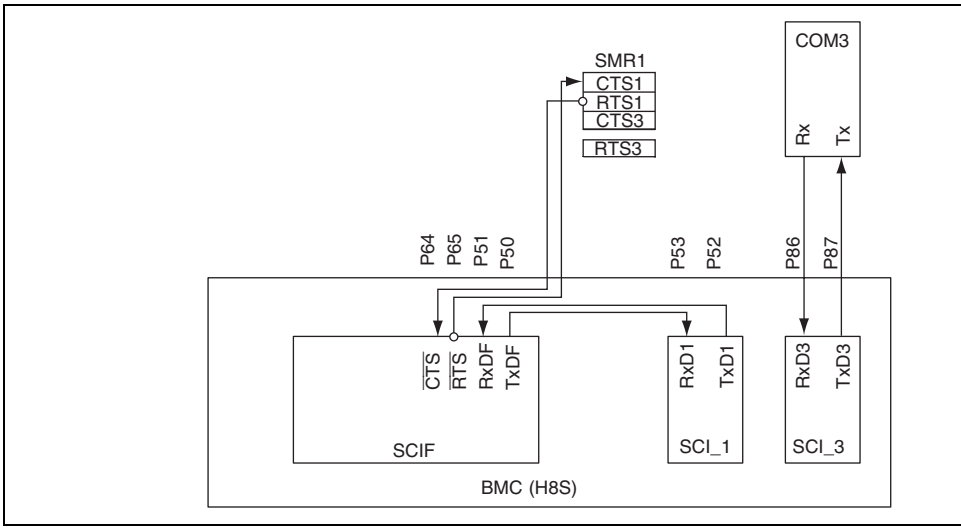


Figure 16.3 Serial Pin Multiplexed Mode 2

The state of $\overline{\text{CTS}}$ of COM1 is reflected in bit CTS1 of the SMR1 register.

The values written to bits DTR1/RTS1 of the SMR1 register are output to $\overline{\text{DTR}}$/$\overline{\text{RTS}}$ of C

Figure 16.4 illustrates the pin connection in serial pin multiplexed mode 3.

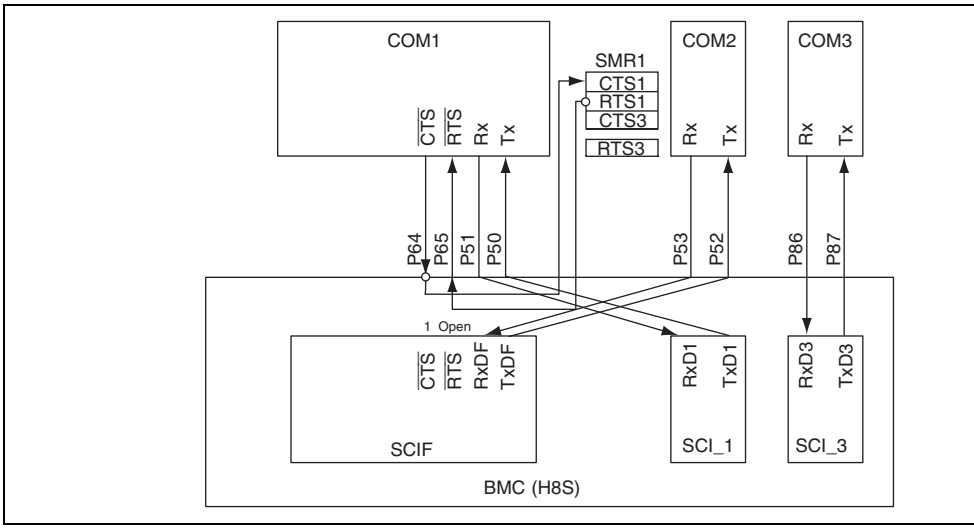


Figure 16.4 Serial Pin Multiplexed Mode 3

The state of $\overline{\text{CTS}}$ of COM1 is reflected to CTS1 bit of SMR1 register.

The values written to bits DTR1/RTS1 of the SMR1 register are output to $\overline{\text{RTS}}$ of COM value written to bit RTS3 of SMR1 is reflected in $\overline{\text{CTS}}$ of SCIF, and the state of $\overline{\text{RTS}}$ of reflected in bit CTS3 of SMR1, allowing SCI_3 and SCIF to communicate each other w flow control.

Figure 16.5 illustrates the pin connection in serial pin multiplexed mode 4.

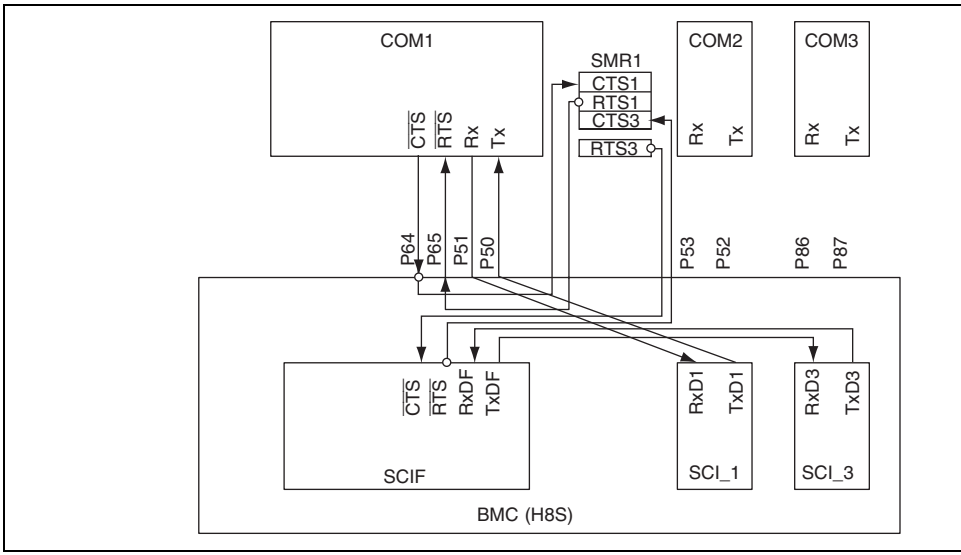


Figure 16.5 Serial Pin Multiplexed Mode 4

  - Clocked synchronous serial format: non-addressing format without acknowledge master operation only
- Conforms to Philips I^2C bus interface (I^2C bus format)
- Two ways of setting slave address (I^2C bus format)
- Start and stop conditions generated automatically in master mode (I^2C bus format)
- Selection of acknowledge output levels when receiving (I^2C bus format)
- Automatic loading of acknowledge bit when transmitting (I^2C bus format)
- Wait function in master mode (I^2C bus format)
  - A wait can be inserted by driving the SCL pin low after data transfer, excluding acknowledgement.
  - The wait can be cleared by clearing the interrupt flag.
- Wait function (I^2C bus format)
  - A wait request can be generated by driving the SCL pin low after data transfer.
  - The wait request is cleared when the next transfer becomes possible.
- Interrupt sources
  - Data transfer end (including when a transition to transmit mode with I^2C bus forn when ICDR data is transferred, or during a wait state)
  - Address match: when any slave address matches or the general call address is rec slave receive mode with I^2C bus format (including address reception after loss of arbitration)
  - Arbitration loss
  - Start condition detection (in master mode)
  - Stop condition detection (in slave mode)
- Selection of 32 internal clocks (in master mode)
- Direct bus drive

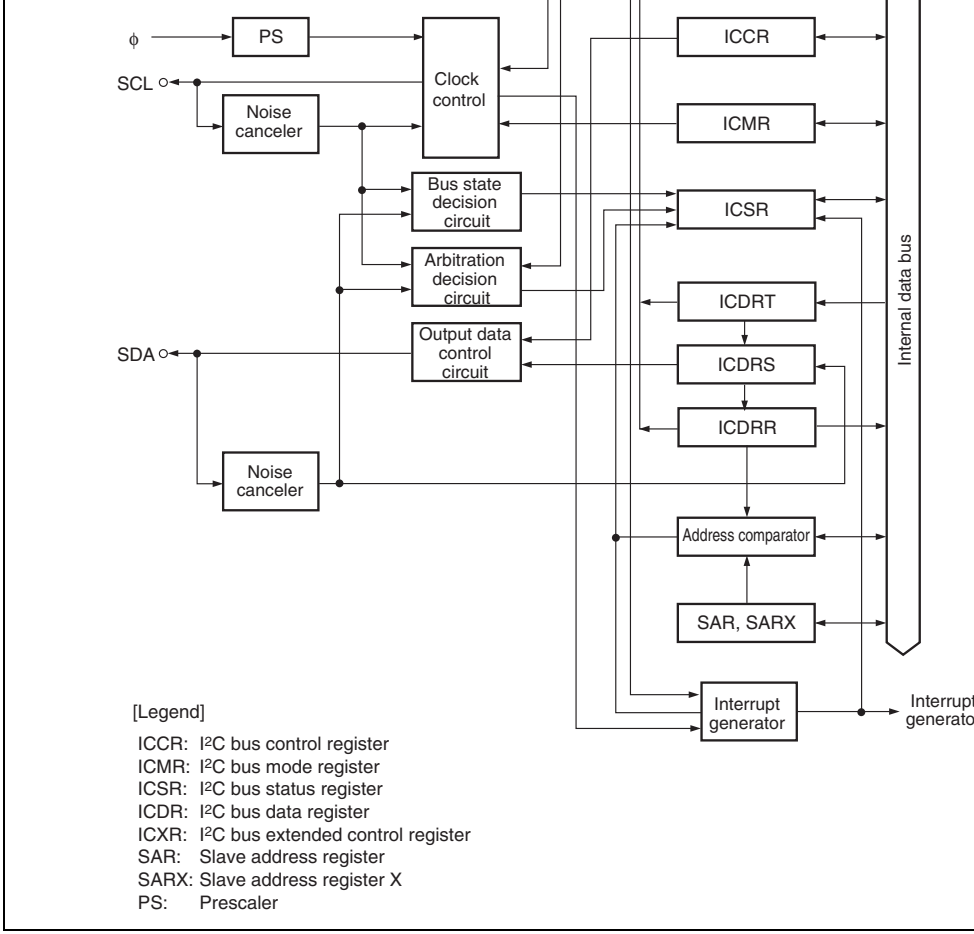


Figure 17.1 Block Diagram of I²C Bus Interface

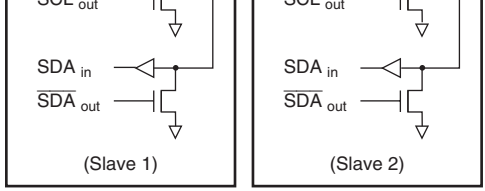


Figure 17.2 I²C Bus Interface Connections (Example: This LSI as Master

	SDA1	Input/Output	Data input/output pin of channel IIC
2	SCL2	Input/Output	Clock input/output pin of channel IIC
	SDA2	Input/Output	Data input/output pin of channel IIC
3	SCL3	Input/Output	Clock input/output pin of channel IIC
	SDA3	Input/Output	Data input/output pin of channel IIC
4	SCL4	Input/Output	Clock input/output pin of channel IIC
	SDA4	Input/Output	Data input/output pin of channel IIC
5	SCL5	Input/Output	Clock input/output pin of channel IIC
	SDA5	Input/Output	Data input/output pin of channel IIC

Note: * In the text, the channel subscript is omitted, and only SCL and SDA are used.

- I²C bus mode register (ICMR)
- I²C bus transfer rate select register (IICX3)
- I²C bus control register (ICCR)
- I²C bus status register (ICSR)
- I²C bus extended control register (ICXR)
- I²C SMbus control register (ICSMBCR)

17.3.1 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

ICDR can be written to and read from only when the ICE bit is set to 1 in ICCR. The initi
of ICDR is undefined.

17.3.2 Slave Address Register (SAR)

SAR sets the slave address and selects the communication format. When the LSI is in sla
with the I^2C bus format selected, if the FS bit is set to 0 and the upper 7 bits of SAR matc
upper 7 bits of the first frame received after a start condition, the LSI operates as the slav
specified by the master device. SAR can be accessed only when the ICE bit in ICCR is cl
0.

Bit	Bit Name	Initial Value	R/W	Description
7	SVA6	All 0	R/W	Slave Addresses 6 to 0
6	SVA5			Set a slave address.
5	SVA4			
4	SVA3			
3	SVA2			
2	SVA1			
1	SVA0			
0	FS	0	R/W	Format Select Selects the communication format together with t bit in SARX. Refer to table 17.2. This bit should be set to 0 when general call addr recognition is performed.

7	SVAX6	All 0	R/W	Second Slave Addresses 6 to 0
6	SVAX5			Set the second slave address.
5	SVAX4			
4	SVAX3			
3	SVAX2			
2	SVAX1			
1	SVAX0			
0	FSX	1	R/W	Format Select X Selects the communication format together with in SAR. Refer to table 17.2.

		• General call address recognized
1	0	I²C 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

- I²C bus format: addressing format with acknowledge bit
- Clocked synchronous serial format: non-addressing format without acknowledge bit, master mode only

				Set this bit to 0 when the I²C bus format is used.
6	WAIT	0	R/W	Wait Insertion Bit This bit is valid only in master mode with the I²C format. 0: Data and the acknowledge bit are transferred consecutively with no wait inserted. 1: After the fall of the clock for the final data bit (clock), the IRIC flag is set to 1 in ICCR, and a begins (with SCL at the low level). When the I is cleared to 0 in ICCR, the wait ends and the acknowledge bit is transferred. For details, refer to section 17.4.7, IRIC Setting and SCL Control.
5	CKS2	All 0	R/W	Transfer Clock Select These bits are used only in master mode. These bits select the required transfer clock rate with bits IICX5 (channel 5), IICX4 (channel 4), a (channel 3) in the IICX3 register and bits IICX2 (2), IICX1 (channel 1), and IICX0 (channel 0) in t register. Refer to table 17.3.
4	CKS1			
3	CKS0			

I²C Bus Format	Clocked Synchronous Serial Format
B'000: 9 bits	B'000: 8 bits
B'001: 2 bits	B'001: 1 bits
B'010: 3 bits	B'010: 2 bits
B'011: 4 bits	B'011: 3 bits
B'100: 5 bits	B'100: 4 bits
B'101: 6 bits	B'101: 5 bits
B'110: 7 bits	B'110: 6 bits
B'111: 8 bits	B'111: 7 bits

				This bit selects a clock rate to be applied to the transfer rate. 0: ϕ/2 1: ϕ/4
2	IICX5	0	R/W	IIC Transfer Rate Select 5, 4, 3
1	IICX4	0	R/W	These bits are used to control IIC_5 to IIC_3 op
0	IICX3	0	R/W	These bits select the transfer rate in master mo together with the CKS2 to CKS0 bits in ICMR. F transfer rate, see table 17.3.

	1	0	0	φ/80	250.0 kHz	312.5 kHz	425.0 kH
			1	φ/100	200.0 kHz	250.0 kHz	340.0 kH
		1	0	φ/112	178.6 kHz	223.2 kHz	303.6 kH
			1	φ/128	156.3 kHz	195.3 kHz	265.6 kH
1	0	0	0	φ/56	357.1 kHz	446.4 kHz*	607.1 kH
			1	φ/80	250.0 kHz	312.5 kHz	425.0 kH
		1	0	φ/96	208.3 kHz	260.4 kHz	354.2 kH
			1	φ/128	156.3 kHz	195.3 kHz	265.6 kH
	1	0	0	φ/160	125.0 kHz	156.3 kHz	212.5 kH
			1	φ/200	100.0 kHz	125.0 kHz	170.0 kH
		1	0	φ/224	89.3 kHz	111.6 kHz	151.8 kH
			1	φ/256	78.1 kHz	97.7 kHz	132.8 kH

Note: * The correct operation cannot be guaranteed since the value is outside the I^2C interface specifications (high-speed mode: max. 400 kHz).
(n = 0 to 5)

	1	0	0	ϕ/160	125.0 kHz	156.3 kHz	212.5 k
			1	ϕ/200	100.0 kHz	125.0 kHz	170.0 k
		1	0	ϕ/224	89.3 kHz	111.6 kHz	151.8 k
			1	ϕ/256	78.1 kHz	97.7 kHz	132.8 k
1	0	0	0	ϕ/112	178.6 kHz	223.2 kHz	303.6 k
			1	ϕ/160	125.0 kHz	156.3 kHz	212.5 k
		1	0	ϕ/190	104.2 kHz	130.2 kHz	177.1 k
			1	ϕ/256	78.1 kHz	97.7 kHz	132.8 k
	1	0	0	ϕ/320	62.5 kHz	78.1 kHz	106.3 k
			1	ϕ/400	50.0 kHz	62.5 kHz	85.0 kH
		1	0	ϕ/448	44.6 kHz	55.8 kHz	75.9 kH
			1	ϕ/512	39.1 kHz	48.8 kHz	66.4 kH

Note: * The correct operation cannot be guaranteed since the value is outside the I^2C interface specifications (high-speed mode: max. 400 kHz).
(n = 0 to 5)

				reception, they are connected to the SCL and S and the I²C bus can be driven. ICMR and ICDR accessed.
6	IEIC	0	R/W	I²C Bus Interface Interrupt Enable 0: Disables interrupts from the I²C bus interface to CPU. 1: Enables interrupts from the I²C bus interface to CPU.
5	MST	0	R/W	Master/Slave Select
4	TRS	0	R/W	Transmit/Receive Select 00: Slave receive mode 01: Slave transmit mode 10: Master receive mode 11: Master transmit mode Both these bits will be cleared by hardware when lose in a bus contention in master mode of the I²C format. In slave receive mode with I²C bus format bit in the first frame immediately after the start co automatically sets these bits in receive mode or tr mode by hardware. Modification of the TRS bit during transfer is defe transfer is completed, and the changeover is mad completion of the transfer.

				MST clearing condition 2) [TRS clearing conditions] (1) When 0 is written by software (except for TR condition 3) (2) When 0 is written in TRS after reading TRS = TRS setting condition 3) (3) When lost in bus contention in I^2C bus format mode [TRS setting conditions] (1) When 1 is written by software (except for TR condition 3) (2) When 1 is written in TRS after reading TRS = TRS clearing condition 3) (3) When 1 is received as the R/$\overline{W}$ bit after the fi address matching in I^2C bus format slave mo
3	ACKE	0	R/W	Acknowledge Bit Decision Selection 0: The value of the acknowledge bit is ignored, continuous transfer is performed. The value received acknowledge bit is not indicated by bit in ICSR, which is always 0. 1: If the acknowledge bit is 1, continuous transf halted. Depending on the receiving device, the acknowl may be significant, in indicating completion of pr of the received data, for instance, or may be fixe have no significance.

- Writing to the BBSY flag is disabled.

[BBSY setting condition]

- When the SDA level changes from high to low the condition of SCL = high, assuming that the condition has been issued.

[BBSY clearing conditions]

- When the SDA level changes from low to high the condition of SCL = high, assuming that the condition has been issued.

To issue a start/stop condition, use the MOV instr

The I^2C bus interface must be set in master transm before the issue of a start condition. Set MST to 1 TRS to 1 before writing 1 in BBSY and 0 in SCP.

The BBSY flag can be read to check whether the (SCL, SDA) is busy or free.

Note: * Even if the BBSY bit is written to, the value of the flag does not change.

- When a start condition is detected in the bus after a start condition is issued (when the ICD set to 1 because of first frame transmission)
- When a wait is inserted between the data and acknowledge bit when the WAIT bit is 1 (fall o transmit/receive clock)
- At the end of data transfer (rise of the 9th transmit/receive clock)
- When a slave address is received after bus m is lost
- If 1 is received as the acknowledge bit (when bit in ICSR is set to 1) when the ACKE bit is 1
- When the AL flag is set to 1 after bus masters while the ALIE bit is 1

I^2C bus format slave mode:

- When the slave address (SVA or SVAX) matc 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 conditio detection (rise of the 9th clock)
- When the general call address is detected (w is received for R/$\overline{W}$ bit, and ADZ flag in ICSR and at the end of data reception up to the sub retransmission start condition or stop conditio detection (rise of the 9th receive clock)
- When 1 is received as an acknowledge bit wh ACKE bit is 1 (when the ACKB bit is set to 1)
- When a stop condition is detected while the S is 0 (when the STOP or ESTP flag in ICSR is

- When transmitting the data in the ICDR registe
  (when data is transferred from ICDRT to ICDR
  transmit mode and the ICDRE flag is set to 1,
  is transferred from ICDRS to ICDRR in receive
  and the ICDRF flag is set to 1.)

[Clearing conditions]

- When 0 is written in IRIC after reading IRIC =
- When ICDR is accessed by DTC * (This may
  clearing condition. For details, see the descrip
  the DTC operation on the next page.

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

Even when the IRIC flag and IRTR flag are set, the ICDRE or ICDRF flag may not be s
IRIC and IRTR flags are not cleared at the end of the specified number of transfers in co
transfer using the DTC. The ICDRE or ICDRF flag is cleared, however, since the specif
number of ICDR reads or writes have been completed.

Tables 17.4 and 17.5 show the relationship between the flags and the transfer states.

													ICDF
1	1	1	0	0	—	0	0	0	0	0	—	0↓	ICDF the a
1	1	1	0	0	—	0	0	0	0	0	—	1	Tran end ICDF
1	1	1	0	0	—	0	0	0	0	0	—	0↓	ICDF the a or af cond detec
1	1	1	0	0	1↑	0	0	0	0	0	—	1↑	Auto trans ICDF with state
1	0	1	0	0	1↑	0	0	0	0	—	1↑	—	Rece with
1	0	1	0	0	—	0	0	0	0	—	0↓	—	ICDF the a
1	0	1	0	0	—	0	0	0	0	—	1	—	Rece with
1	0	1	0	0	—	0	0	0	0	—	0↓	—	ICDF the a
1	0	1	0	0	1↑	0	0	0	0	—	1↑	—	Auto trans ICDF ICDF abov
0↓	0↓	1	0	0	—	0	1↑	0	0	—	—	—	Arbit
1	—	0↓	0	0	—	0	0	0	0	—	—	0↓	Stop detec

[Legend]

0: 0-state retained 1: 1-state retained —: Previous state retained

0↓: Cleared to 0 1↑: Set to 1

0	1↑/0 *1	1	0	0	1↑	1↑	—	0	0	0	1↑	1	S fi (
0	1	1	0	0	—	—	—	—	0	1↑	—	—	T e a
0	1	1	0	0	1↑/0 *1	—	—	—	0	0	—	1↑	T e I
0	1	1	0	0	—	—	0↓	0↓	0	0	—	0↓	I w s
0	1	1	0	0	—	—	—	—	0	0	—	1	T e I
0	1	1	0	0	—	—	0↓	0↓	0	0	—	0↓	I w s
0	1	1	0	0	1↑/0 *2	—	0	0	0	0	—	1↑	A c fr I tl s
0	0	1	0	0	1↑/0 *2	—	—	—	—	—	1↑	—	P w
0	0	1	0	0	—	—	0↓	0↓	0↓	—	0↓	—	I w s

*3 *3

[Legend]

0: 0-state retained 1: 1-state retained —: Previous state retained

0↓: Cleared to 0 1↑: Set to 1

Notes: 1. Set to 1 when 1 is received as a R/$\overline{W}$ bit following an address.

2. Set to 1 when the AASX bit is set to 1.

3. When ESTP=1, STOP is 0, or when STOP=1, ESTP is 0.

				• When 0 is written in ESTP after reading EST • When the IRIC flag in ICCR is cleared to 0
6	STOP	0	R/(W)*	Normal Stop Condition Detection Flag This bit is valid in I^2C bus format slave mode. [Setting condition] When a stop condition is detected after frame tra completed. [Clearing conditions] • When 0 is written in STOP after reading STO • When the IRIC flag is cleared to 0
5	IRTR	0	R/(W)*	I^2C Bus Interface Continuous Transfer Interrupt I Flag Indicates that the I^2C bus interface has issued ar request to the CPU, and the source is completio reception/transmission of one frame in continuou transmission/reception for which DTC activation possible. When the IRTR flag is set to 1, the IRI also set to 1 at the same time. [Setting conditions] I^2C bus format slave mode: • When the ICDRE or ICDRF flag in ICDR is s when AASX = 1 I^2C bus format master mode or clocked synchron format mode: • When the ICDRE or ICDRF flag is set to 1 [Clearing conditions] • When 0 is written after reading IRTR = 1 • When the IRIC flag is cleared to 0 while ICE

				• When 0 is written in AASX after reading AASX = 1 • When a start condition is detected • In master mode
3	AL	0	R/(W)*	Arbitration Lost Flag Indicates that arbitration was lost in master mode. [Setting conditions] When ALSL=0 • If the internal SDA and SDA pin disagree at th SCL in master transmit mode • If the internal SCL line is high at the fall of SCL master mode When ALSL=1 • If the internal SDA and SDA pin disagree at th SCL in master transmit mode • If the SDA pin is driven low by another device the I^2C bus interface drives the SDA pin low, a start condition instruction was executed in ma transmit mode [Clearing conditions] • When ICDR is written to (transmit mode) or re (receive mode) • When 0 is written in AL after reading AL = 1

				[Clearing conditions] • When ICDR is written to (transmit mode) or r (receive mode) • When 0 is written in AAS after reading AAS = • In master mode
1	ADZ	0	R/(W)*	General Call Address Recognition Flag In I^2C bus format slave receive mode, this flag is the first frame following a start condition is the g address (H'00). [Setting condition] When the general call address (one frame includ R/$\overline{W}$ bit is H'00) is detected in slave receive mod = 0 or FSX = 0 [Clearing conditions] • When ICDR is written to (transmit mode) or r (receive mode) • When 0 is written in ADZ after reading ADZ = • In master mode If a general call address is detected while FS=1 FSX=0, the ADZ flag is set to 1; however, the ge address is not recognized (AAS flag is not set to

ACKE=1 in transmit mode

- When 0 is written to the ACKE bit

Receive mode:

0: Returns 0 as acknowledge data after data rece

1: Returns 1 as acknowledge data after data rece

When this bit is read, the value loaded from the bu
(returned by the receiving device) is read in transm
(when TRS = 1). In reception (when TRS = 0), the
set by internal software is read.

When this bit is written, acknowledge data that is
after receiving is rewritten regardless of the TRS v
the ICSR register bit is written using bit-manipulat
instructions, the acknowledge data should be re-s
the acknowledge data setting is rewritten by the A
reading value.

Write the ACKE bit to 0 to clear the ACKB flag to
transmission is ended and a stop condition is issu
master mode, or before transmission is ended and
released to issue a stop condition by a master dev

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

				the stop condition is detected (STOP = 1 or E in slave mode. 1: Disables IRIC flag setting and interrupt gener when the stop condition is detected.
6	HNDS	0	R/W	Handshake Receive Operation Select Enables or disables continuous receive operatio receive mode. 0: Enables continuous receive operation 1: Disables continuous receive operation When the HNDS bit is cleared to 0, receive oper performed continuously after data has been rec successfully while ICDRF flag is 0. When the HNDS bit is set to 1, SCL is fixed to th level after data has been received successfully ICDRF flag is 0; thus disabling the next data to b transferred. The bus line is released and next re operation is enabled by reading the receive data

- When data is received successfully and transf
  from ICDRS to ICDRR.

(1) When data is received successfully while ICD
(at the rise of the 9th clock pulse).

(2) When ICDR is read successfully in receive m
data was received while ICDRF = 1.

[Clearing conditions]

- When ICDR (ICDRR) is read.
- When 0 is written to the ICE bit.

When ICDRF is set due to the condition (2) above
is temporarily cleared to 0 when ICDR (ICDRR) is
however, since data is transferred from ICDRS to
immediately, ICDRF is set to 1 again.

Note that ICDR cannot be read successfully in tra
mode (TRS = 1) because data is not transferred f
ICDRS to ICDRR. Be sure to read data from ICDF
receive mode (TRS = 0).

- When the start condition is detected from the state in I^2C bus format or serial format.
- When data is transferred from ICDRT to ICDR
  1. When data is transmitted completely while = 0 (at the rise of the 9th clock pulse).
  2. When data is written to ICDR completely mode after data was transmitted while ICE

[Clearing conditions]

- When data is written to ICDR (ICDRT).
- When the stop condition is detected in I^2C bus or serial format.
- When 0 is written to the ICE bit.

Note that if the ACKE bit is set to 1 in I^2C bus form enabling acknowledge bit decision, ICDRE is not data is transmitted completely while the acknowle 1.

When ICDRE is set due to the condition (2) above is temporarily cleared to 0 when data is written to (ICDRT); however, since data is transferred from ICDRS immediately, ICDRF is set to 1 again. Do data to ICDR when TRS = 0 because the ICDRE value is invalid during the time.

				interface outputs at the rise of SCL and the SCL driven low by another device. 1: If the SDA pin state disagrees with the data tha interface outputs at the rise of SCL and the SDA driven low by another device in idle state or afte start condition instruction was executed.
1	FNC1	0	R/W	Function Bit
0	FNC0	0	R/W	These bits cancel some restrictions on usage. For refer to section 17.6, Usage Notes. 00: Restrictions on operation remaining in effect 01: Setting prohibited 10: Setting prohibited 11: Restrictions on operation canceled

Bit	Bit Name	Value	R/W	Description
7	SMB5E	0	R/W	SMBus Enable
6	SMB4E	0	R/W	These bits enable/disable to support the SMBus,
5	SMB3E	0	R/W	combination with bits FSEL1 and FSEL0. Bits SM
4	SMB2E	0	R/W	SMB4E, SMB3E, SMB2E, SMB1E, and SMB0E
3	SMB1E	0	R/W	IIC_5, IIC_4, IIC_3, IIC_2, IIC_1, and IIC_0, resp
2	SMB0E	0	R/W	0: Disables to support the SMBus
				1: Enables to support the SMBus
1	FSEL1	0	R/W	Frequency Selection
0	FSEL0	0	R/W	These bits must be specified to match the systen frequency in order to support the SMBus. For de setting, see table 17.7.

1	0	Min.	300	240*	176*
		Max.	550	440	324
	1	Min.	500	400	294*
		Max.	950	760	559

Notes: n = 0 to 5

* Since the value is outside the SMBus specification, it should not be set.

Table 17.7 ISCMBCR Setting

System Clock	SMBnE	FSEL1	FSEL0
20 MHz	1	1	0
20 to 34 MHz	1	1	1

n = 0 to 5

Figure 17.5 shows the I²C bus timing.

The symbols used in figures 17.3 to 17.5 are explained in table 17.8.

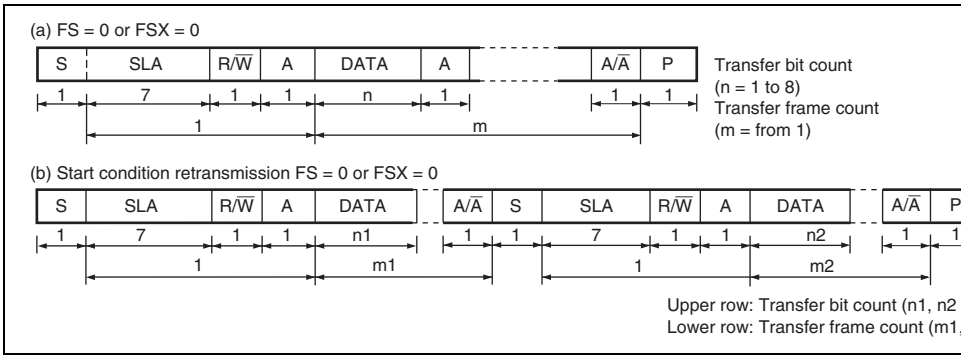


Figure 17.3 I²C Bus Data Formats (I²C Bus Formats)

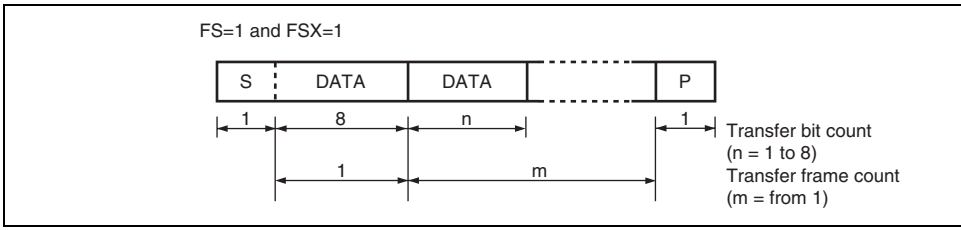


Figure 17.4 I²C Bus Data Formats (Serial Formats)

S	Start condition. The master device drives SDA from high to low while SCL is h
SLA	Slave address. The master device selects the slave device.
R/$\overline{W}$	Indicates the direction of data transfer: from the slave device to the master de when R/$\overline{W}$ is 1, or from the master device to the slave device when R/W is 0
A	Acknowledge. The receiving device drives SDA low to acknowledge a transfe slave device returns acknowledge in master transmit mode, and the master d returns acknowledge in master receive mode.)
DATA	Transferred data. The bit length of transferred data is set with the BC2 to BC0 ICMR. The MSB first or LSB first is switched with the MLS bit in ICMR.
P	Stop condition. The master device drives SDA from low to high while SCL is h

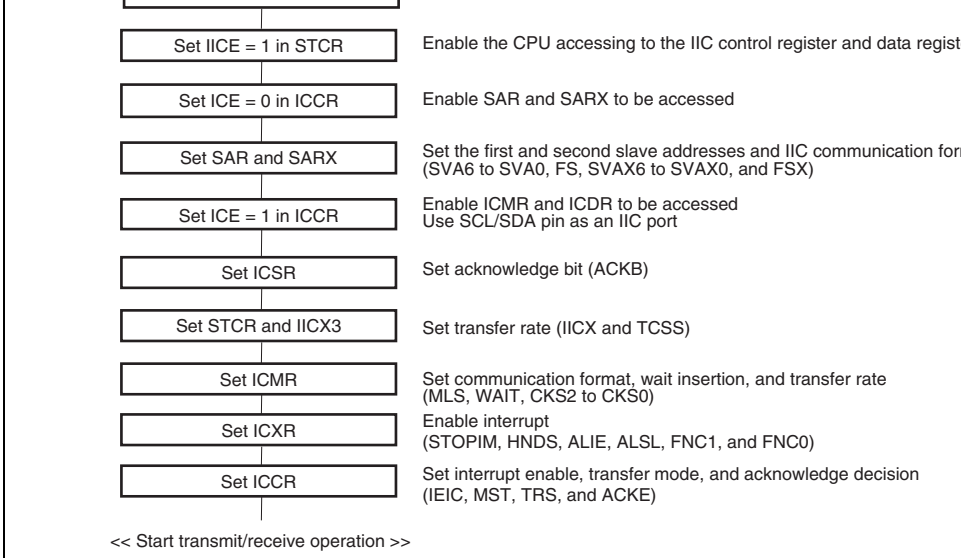


Figure 17.6 Sample Flowchart for IIC Initialization

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.

17.4.3 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.

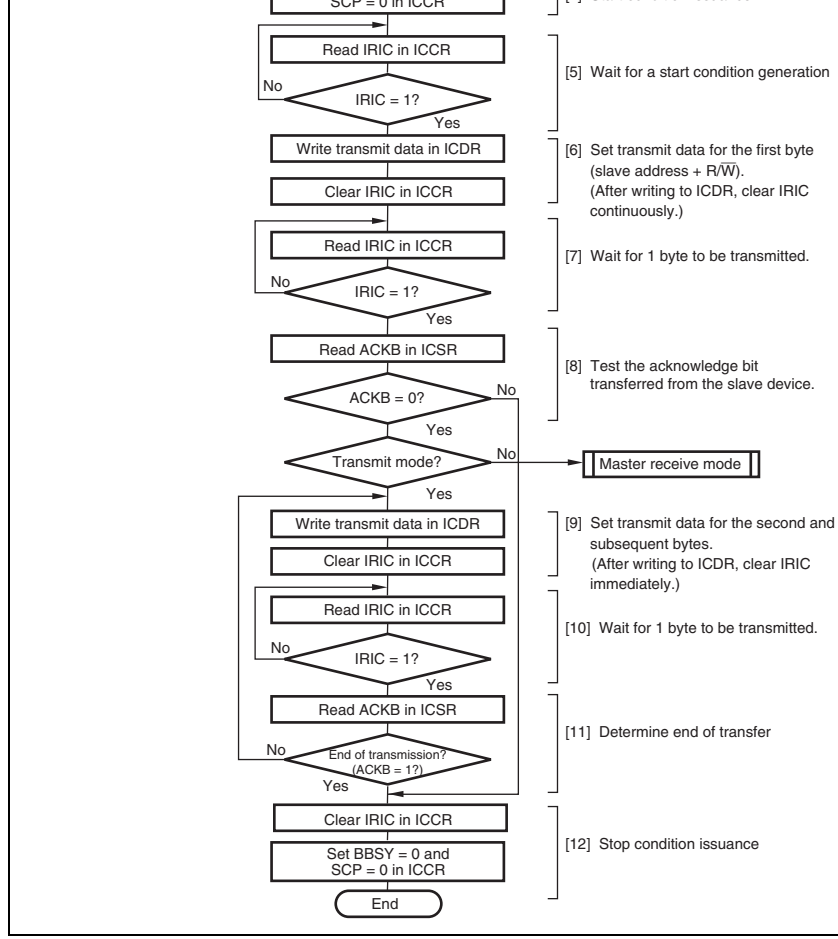


Figure 17.7 Sample Flowchart for Operations in Master Transmit Mode

6. Write the data (slave address + R/W) to ICDR.

   With the I^2C bus format (when the FS bit in SAR or the FSX bit in SARX is 0), the f
   data following the start condition indicates the 7-bit slave address and transmit/recei
   direction (R/$\overline{W}$).

   To determine the end of the transfer, the IRIC flag is cleared to 0. After writing to IC
   IRIC continuously so no other interrupt handling routine is executed. If the time for
   transmission of one frame of data has passed before the IRIC clearing, the end of tra
   cannot be determined. The master device sequentially sends the transmission clock a
   data written to ICDR. The selected slave device (i.e. the slave device with the match
   address) drives SDA low at the 9th transmit clock pulse and returns an acknowledge
7. When one frame of data has been transmitted, the IRIC flag is set to 1 at the rise of t
   transmit clock pulse. After one frame has been transmitted, SCL is automatically fix
   synchronization with the internal clock until the next transmit data is written.
8. Read the ACKB bit in ICSR to confirm that ACKB is cleared to 0. When the slave d
   not acknowledged (ACKB bit is 1), operate step [12] to end transmission, and retry t
   transmit operation.
9. Write the transmit data to ICDR.

   As indicating the end of the transfer, the IRIC flag is cleared to 0. Perform the ICDR
   the IRIC flag clearing sequentially, just as in step [6]. Transmission of the next fram
   performed in synchronization with the internal clock.
10. When one frame of data has been transmitted, the IRIC flag is set to 1 at the rise of t
    transmit clock pulse. After one frame has been transmitted, SCL is automatically fix
    synchronization with the internal clock until the next transmit data is written.
11. Read the ACKB bit in ICSR.

    Confirm that the slave device has been acknowledged (ACKB bit is 0). When there i
    to be transmitted, go to step [9] to continue the next transmission operation. When th
    device has not acknowledged (ACKB bit is set to 1), operate step [12] to end transm

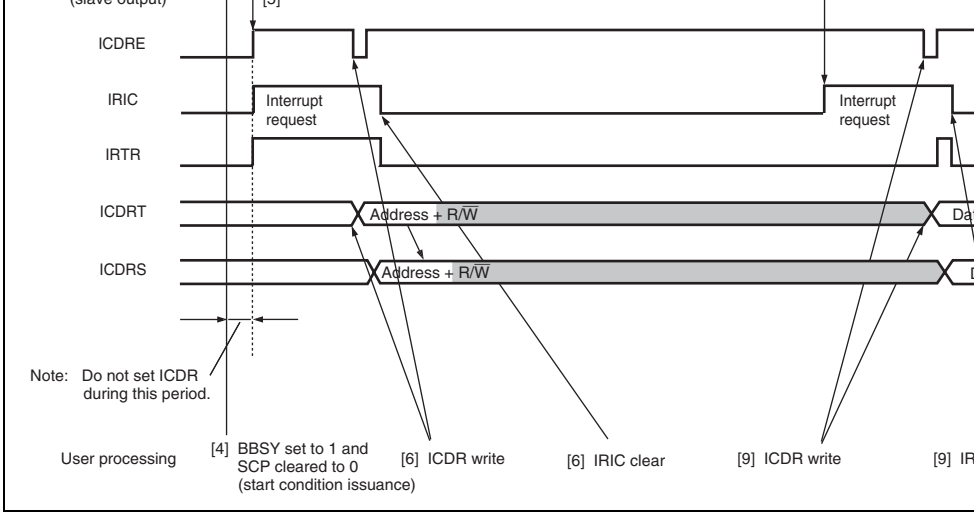


Figure 17.8 Operation Timing Example in Master Transmit Mode (MLS = WAI

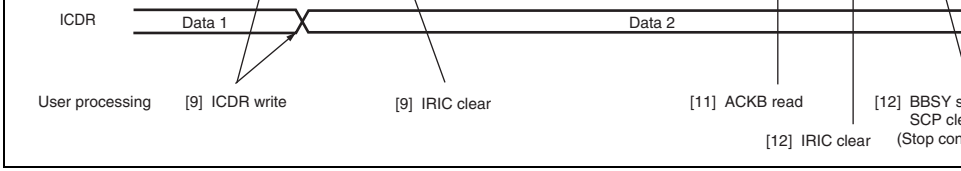


Figure 17.9 Stop Condition Issuance Operation Timing Example in Master Trans (MLS = WAIT = 0)

17.4.4 Master Receive Operation

In I^2C bus format master receive mode, the master device outputs the receive clock, rece and returns an acknowledge signal. The slave device transmits data.

The master device transmits data containing the slave address and R/$\overline{W}$ (1: read) in the f following the start condition issuance in master transmit mode, selects the slave device, switches the mode for receive operation.

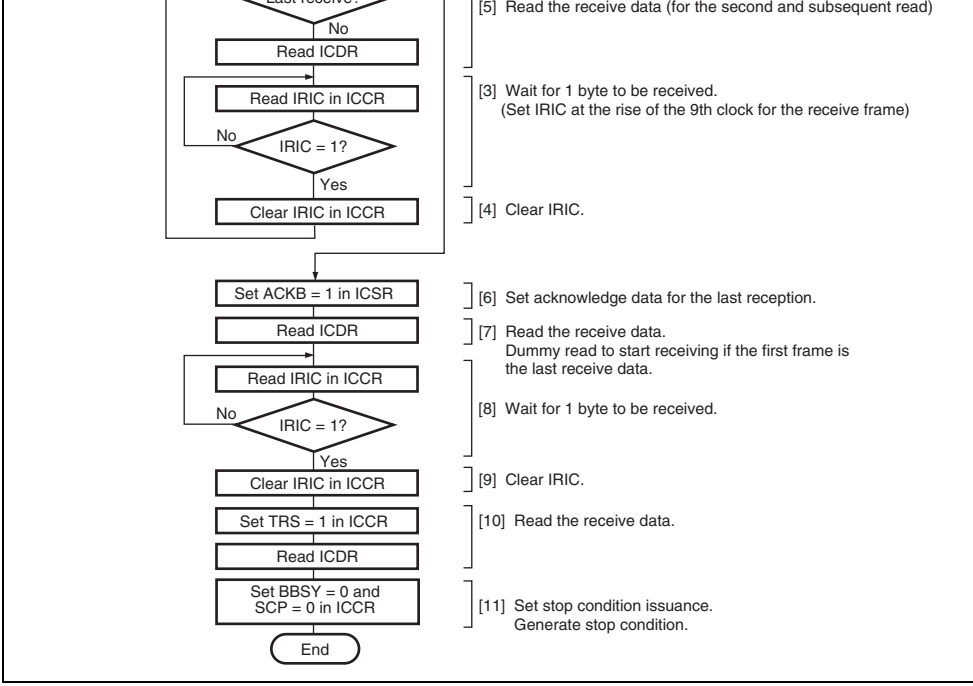


Figure 17.10 Sample Flowchart for Operations in Master Receive Mode (HNDS

sequentially transferred to ICDRS in synchronization with the rise of the receive clo

3. The master device drives SDA low to return the acknowledge data at the 9th receive pulse. The receive data is transferred to ICDRR from ICDRS at the rise of the 9th cl setting the ICDRF, IRIC, and IRTR flags to 1. If the IEIC bit has been set to 1, an i request is sent to the CPU.

   The master device drives SCL low from the fall of the 9th receive clock pulse to the reading.
4. Clear the IRIC flag to determine the next interrupt.

   Go to step [6] to halt reception operation if the next frame is the last receive data.
5. Read ICDR receive data. This clears the ICDRF flag to 0. The master device outputs receive clock continuously to receive the next data.

   Data can be received continuously by repeating steps [3] to [5].
6. Set the ACKB bit to 1 so as to return the acknowledge data for the last reception.
7. Read ICDR receive data. This clears the ICDRF flag to 0. The master device outputs receive clock to receive data.
8. When one frame of data has been received, the ICDRF, IRIC, and IRTR flags are se rise of the 9th receive clock pulse.
9. Clear the IRIC flag to 0.
10. Read ICDR receive data after setting the TRS bit. This clears the ICDRF flag to 0.
11. Clear the BBSY bit and SCP bit to 0 in ICCR. This changes SDA from low to high v is high, and generates the stop condition.

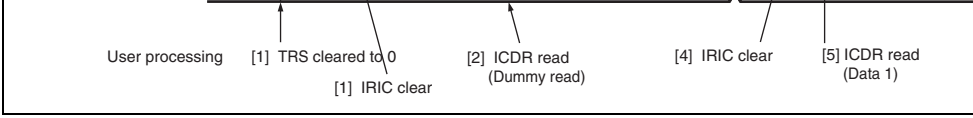


Figure 17.11 Master Receive Mode Operation Timing Example (MLS = WAIT = 0, HNDS = 1)

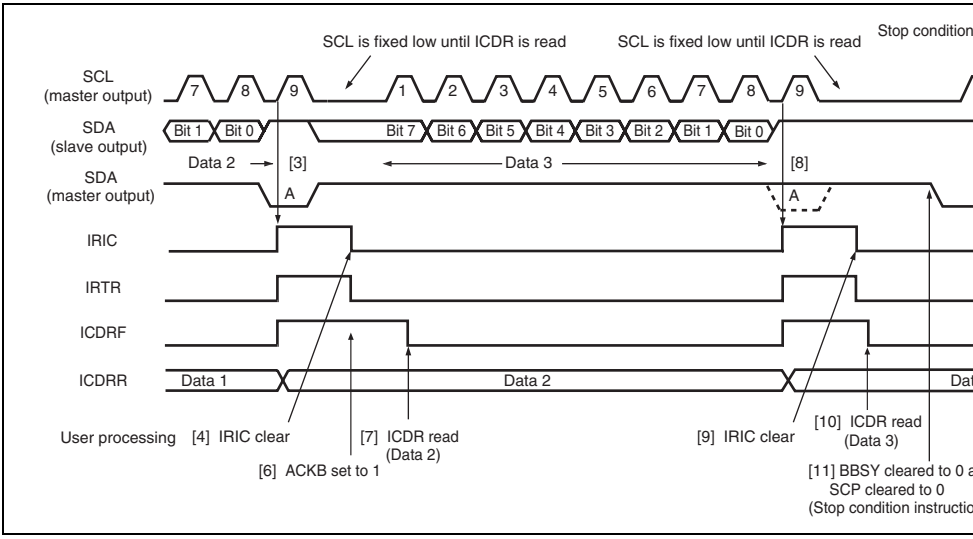


Figure 17.12 Stop Condition Issuance Timing Example in Master Receive Mo (MLS = WAIT = 0, HNDS = 1)

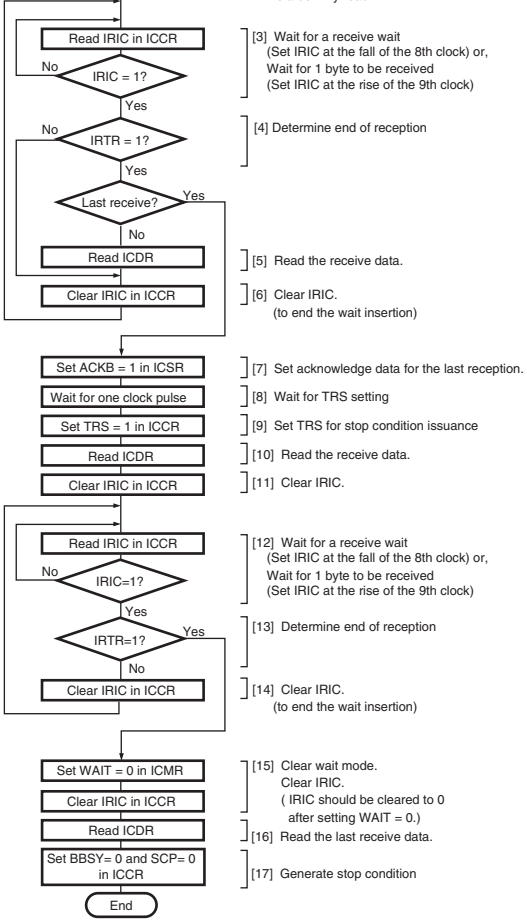


Figure 17.13 Sample Flowchart for Operations in Master Receive Mode (receiving multiple bytes) (WAIT = 1)

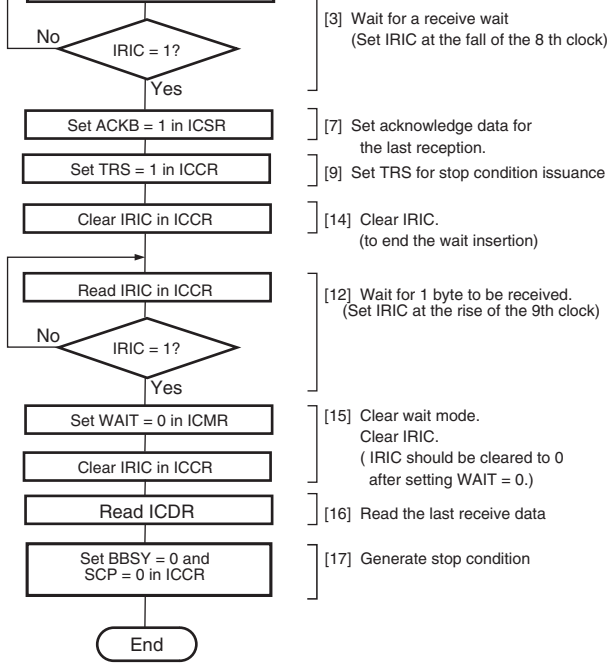


Figure 17.14 Sample Flowchart for Operations in Master Receive Mode (receiving a single byte) (WAIT = 1)

The reception procedure and operations using the wait function (WAIT bit), by which dat sequentially received in synchronization with ICDR (ICDRR) read operations, are descril below.

The following describes the multiple-byte reception procedure. In single-byte reception, s steps of the following procedure are omitted. At this time, follow the procedure shown in 17.14

flag clearing.

(2) At the rise of the 9th receive clock pulse for one frame

The IRTR and ICDRF flags are set to 1, indicating that one frame of data has received. The master device outputs the receive clock continuously to receive data.

4. Read the IRTR flag in ICSR.

   If the IRTR flag is 0, execute step [6] to clear the IRIC flag to 0 to release the wait st

   If the IRTR flag is 1 and the next data is the last receive data, execute step [7] to halt
5. If IRTR flag is 1, read ICDR receive data.
6. Clear the IRIC flag. When the flag is set as (1) in step [3], the master device outputs clock and drives SDA low at the 9th receive clock pulse to return an acknowledge si

   Data can be received continuously by repeating steps [3] to [6].

7. Set the ACKB bit in ICSR to 1 so as to return the acknowledge data for the last recep
8. After the IRIC flag is set to 1, wait for at least one clock pulse until the rise of the fir pulse for the next receive data.
9. Set the TRS bit in ICCR to 1 to switch from receive mode to transmit mode. The TR becomes valid when the rising edge of the next 9th clock pulse is input.
10. Read the ICDR receive data.
11. Clear the IRIC flag to 0.

condition.

14. If IRTR flag is 0, clear the IRIC flag to 0 to release the wait state.

    Execute step [12] to read the IRIC flag to detect the end of reception.

15. Clear the WAIT bit in ICMR to cancel the wait mode.

    Clearing of the IRIC flag should be done while WAIT = 0. (If the WAIT bit is cleared after clearing the IRIC flag and then an instruction to issue a stop condition is execute stop condition may not be issued correctly.)

16. Read the last ICDR receive data.

17. Clear the BBSY bit and SCP bit to 0 in ICCR. This changes SDA from low to high wi is high, and generates the stop condition.

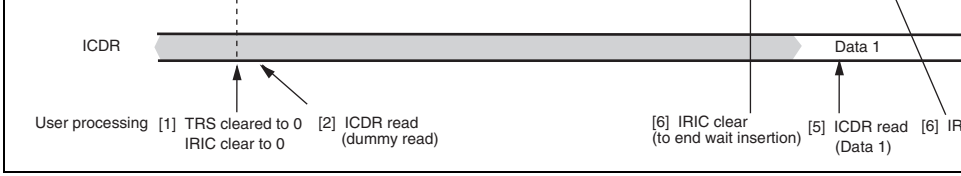


Figure 17.15 Master Receive Mode Operation Timing Example (MLS = ACKB = 0, WAIT = 1)

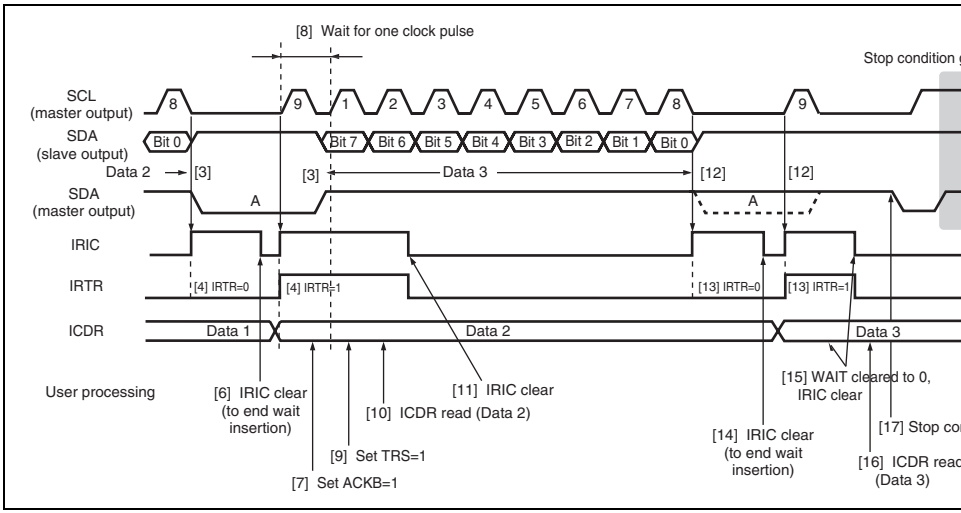


Figure 17.16 Stop Condition Issuance Timing Example in Master Receive M (MLS = ACKB = 0, WAIT = 1)

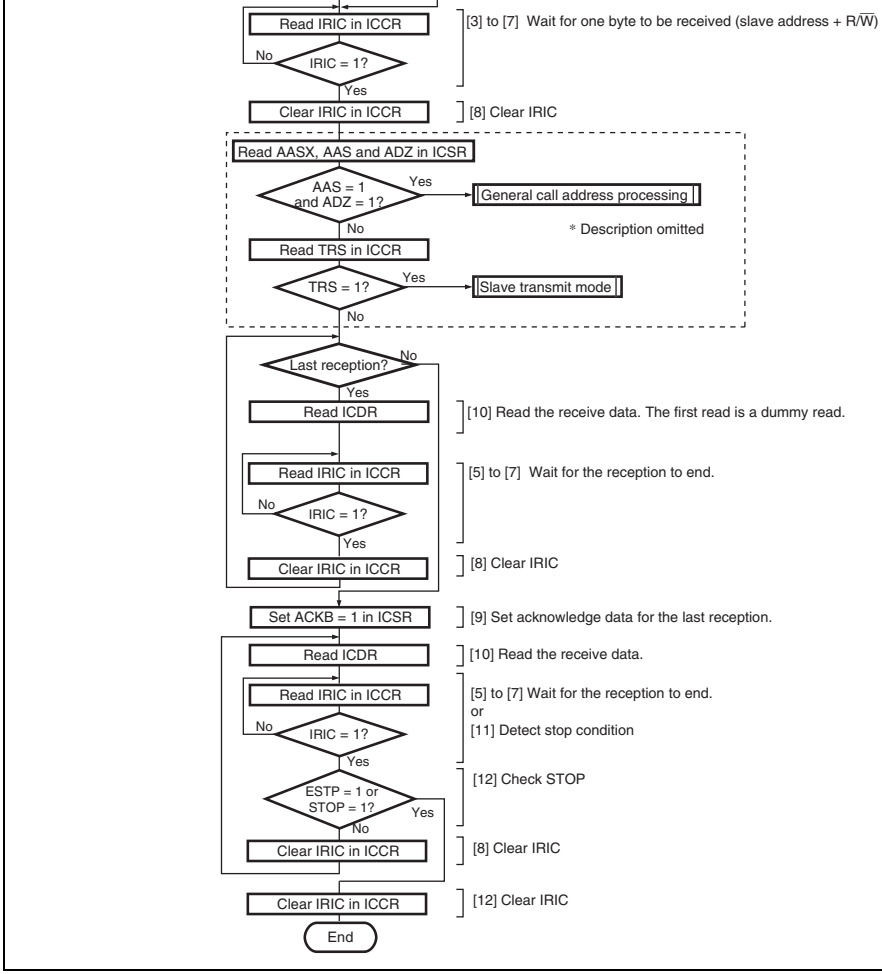


Figure 17.17 Sample Flowchart for Operations in Slave Receive Mode (HND

(R/$\overline{W}$), in synchronization with the transmit clock pulses.

4. When the slave address matches in the first frame following the start condition, the de operates as the slave device specified by the master device. If the 8th data bit (R/$\overline{W}$) is TRS bit remains cleared to 0, and slave receive operation is performed. If the 8th data (R/$\overline{W}$) is 1, the TRS bit is set to 1, and slave transmit operation is performed. When th address does not match, receive operation is halted until the next start condition is det
5. At the 9th clock pulse of the receive frame, the slave device returns the data in the AC as the acknowledge data.
6. At the rise of the 9th clock pulse, the IRIC flag is set to 1. If the IEIC bit has been set interrupt request is sent to the CPU.

   If the AASX bit has been set to 1, IRTR flag is also set to 1.
7. At the rise of the 9th clock pulse, the receive data is transferred from ICDRS to ICDR setting the ICDRF flag to 1. The slave device drives SCL low from the fall of the 9th clock pulse until data is read from ICDR.
8. Confirm that the STOP bit is cleared to 0, and clear the IRIC flag to 0.
9. If the next frame is the last receive frame, set the ACKB bit to 1.
10. If ICDR is read, the ICDRF flag is cleared to 0, releasing the SCL bus line. This enab master device to transfer the next data.

    Receive operations can be performed continuously by repeating steps [5] to [10].

11. When the stop condition is detected (SDA is changed from low to high when SCL is the BBSY flag is cleared to 0 and the STOP bit is set to 1. If the STOPIM bit has be cleared to 0, the IRIC flag is set to 1.
12. Confirm that the STOP bit is set to 1, and clear the IRIC flag to 0.

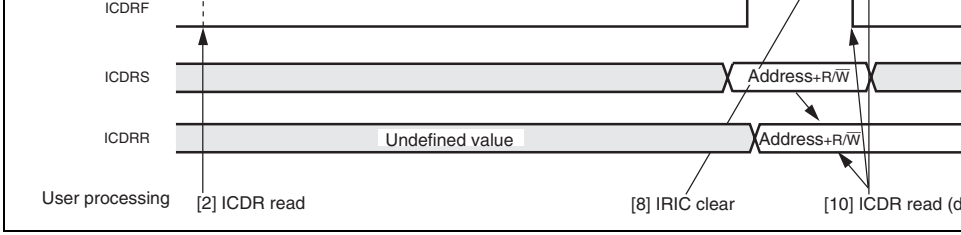


Figure 17.18 Slave Receive Mode Operation Timing Example (1) (MLS = 0, HN

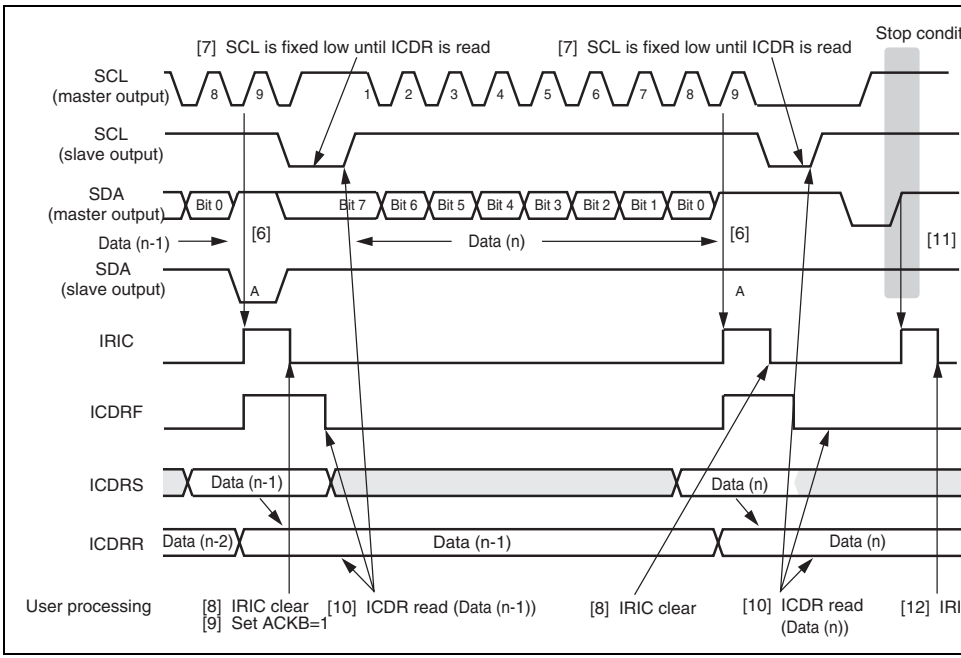


Figure 17.19 Slave Receive Mode Operation Timing Example (2) (MLS = 0, HN

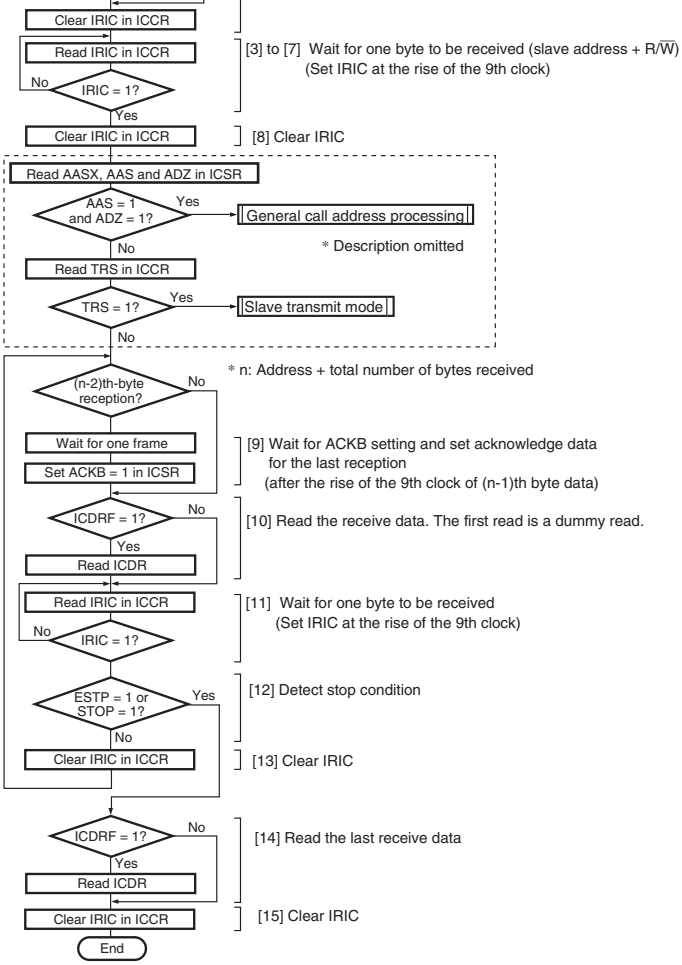


Figure 17.20 Sample Flowchart for Operations in Slave Receive Mode (HNDS

4. When the slave address matches in the first frame following the start condition, the d
operates as the slave device specified by the master device. If the 8th data bit (R/$\overline{W}$)
TRS bit remains cleared to 0, and slave receive operation is performed. If the 8th dat
(R/$\overline{W}$) is 1, the TRS bit is set to 1, and slave transmit operation is performed. When
address does not match, receive operation is halted until the next start condition is de
5. At the 9th clock pulse of the receive frame, the slave device returns the data in the A
as the acknowledge data.
6. At the rise of the 9th clock pulse, the IRIC flag is set to 1. If the IEIC bit has been se
interrupt request is sent to the CPU.

   If the AASX bit has been set to 1, the IRTR flag is also set to 1.
7. At the rise of the 9th clock pulse, the receive data is transferred from ICDRS to ICD
setting the ICDRF flag to 1.
8. Confirm that the STOP bit is cleared to 0 and clear the IRIC flag to 0.
9. If the next read data is the third last receive frame, wait for at least one frame time to
ACKB bit. Set the ACKB bit after the rise of the 9th clock pulse of the second last re
frame.
10. Confirm that the ICDRF flag is set to 1 and read ICDR. This clears the ICDRF flag t
11. At the rise of the 9th clock pulse or when the receive data is transferred from IRDRS
ICDRR due to ICDR read operation, The IRIC and ICDRF flags are set to 1.
12. When the stop condition is detected (SDA is changed from low to high when SCL is
BBSY flag is cleared to 0 and the STOP or ESTP flag is set to 1. If the STOPIM bit
cleared to 0, the IRIC flag is set to 1. In this case, execute step 14 to read the last rec
13. Clear the IRIC flag to 0.

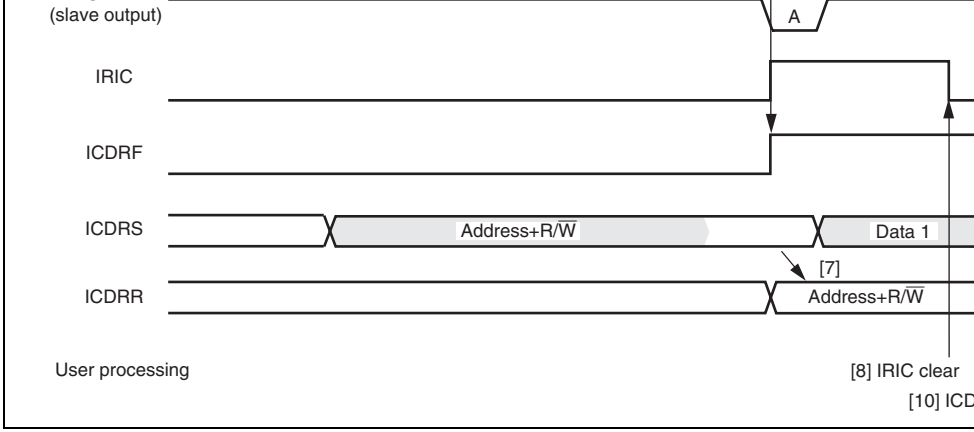


Figure 17.21 Slave Receive Mode Operation Timing Example (1)
(MLS = ACKB = 0, HNDS = 0)

Figure 17.22 Slave Receive Mode Operation Timing Example (2) (MLS = ACKB = 0, HNDS = 0)

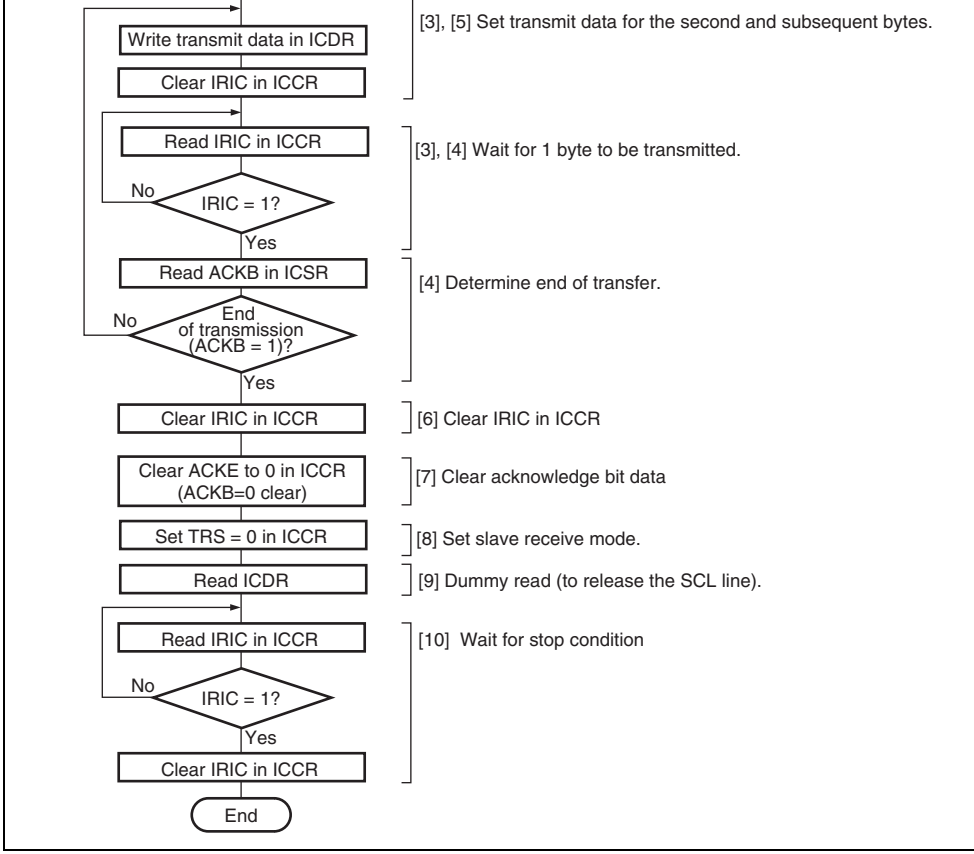


Figure 17.23 Sample Flowchart for Slave Transmit Mode

until ICDR data is written, to disable the master device to output the next transfer clo

3. After clearing the IRIC flag to 0, write data to ICDR. At this time, the ICDRE flag is 0. The written data is transferred to ICDRS, and the ICDRE and IRIC flags are set to The slave device sequentially sends the data written into ICDRS in accordance with output by the master device.

   The IRIC flag is cleared to 0 to detect the end of transmission. Processing from the I register writing to the IRIC flag clearing should be performed continuously. Prevent interrupt processing from being inserted.

4. The master device drives SDA low at the 9th clock pulse, and returns an acknowledg As this acknowledge signal is stored in the ACKB bit in ICSR, this bit can be used to determine whether the transfer operation was performed successfully. When one fram has been transmitted, the IRIC flag in ICCR is set to 1 at the rise of the 9th transmit pulse. When the ICDRE flag is 0, the data written into ICDR is transferred to ICDRS ICDRE and IRIC flags are set to 1 again. If the ICDRE flag has been set to 1, this sla drives SCL low from the fall of the 9th transmit clock until data is written to ICDR.

5. To continue transmission, write the next data to be transmitted into ICDR. The ICDR cleared to 0. The IRIC flag is cleared to 0 to detect the end of transmission. Processi the ICDR register writing to the IRIC flag clearing should be performed continuousl any other interrupt processing from being inserted.

   Transmit operations can be performed continuously by repeating steps 4 and 5.

6. Clear the IRIC flag to 0.
7. To end transmission, clear the ACKE bit in the ICCR register to 0, to clear the ackno bit stored in the ACKB bit to 0.
8. Clear the TRS bit to 0 for the next address reception, to set slave receive mode.
9. Dummy-read ICDR to release SCL on the slave side.

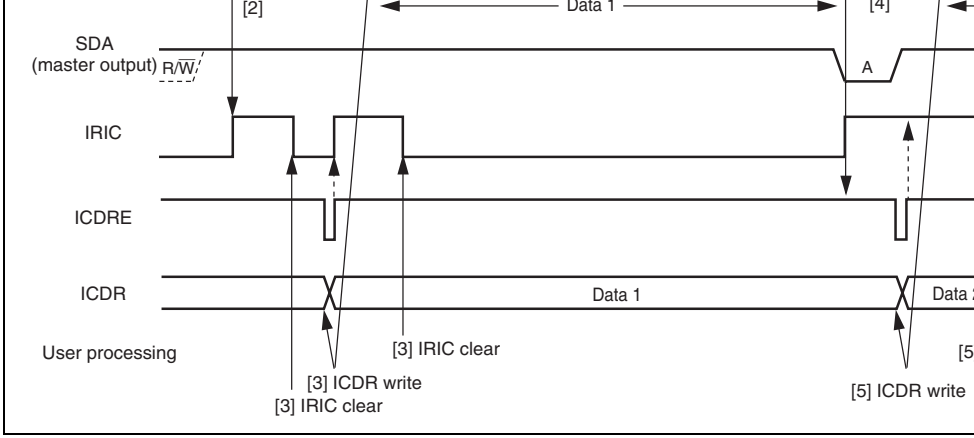
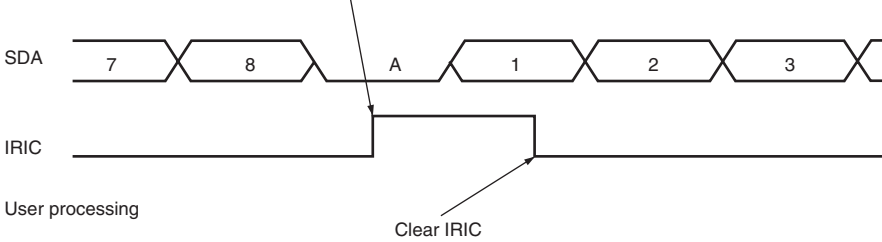
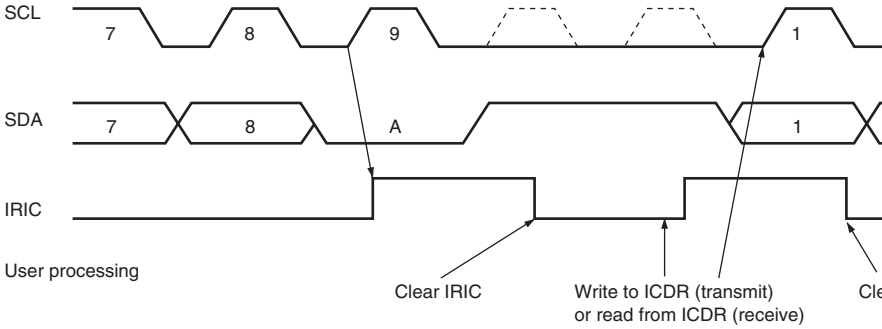


Figure 17.24 Slave Transmit Mode Operation Timing Example (MLS = 0)



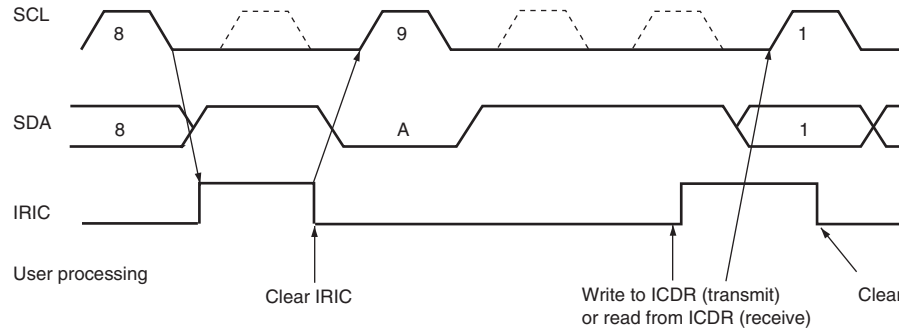
(a) Data transfer ends with ICDRE=0 at transmission, or ICDRF=0 at reception.



(b) Data transfer ends with ICDRE=1 at transmission, or ICDRF=1 at reception.

Figure 17.25 IRIC Setting Timing and SCL Control (1)

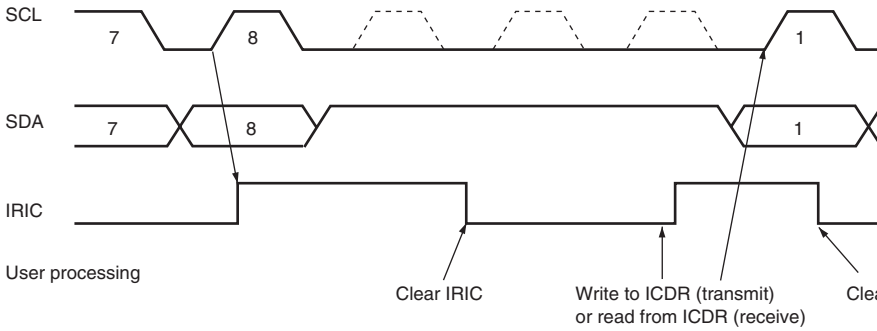
(a) Data transfer ends with ICDRE=0 at transmission, or ICDRF=0 at reception.



(b) Data transfer ends with ICDRE=1 at transmission, or ICDRF=1 at reception.

Figure 17.26 IRIC Setting Timing and SCL Control (2)

(a) Data transfer ends with ICDRE=0 at transmission, or ICDRF=0 at reception.



(b) Data transfer ends with ICDRE=1 at transmission, or ICDRF=1 at reception.

Figure 17.27 IRIC Setting Timing and SCL Control (3)

transmission is completed with the acknowledge bit value of 1 when the ACKE bit is 1, D
not initiated, thus allowing an interrupt to be generated if enabled.

The acknowledge bit may indicate specific events such as completion of receive data proc
for some receiving devices, and for other receiving devices, the acknowledge bit may be l
indicating no specific events.

The I^2C bus format provides for selection of the slave device and transfer direction by me
the slave address and the R/$\overline{W}$ bit, confirmation of reception with the acknowledge bit, in
of the last frame, and so on. Therefore, continuous data transfer using the DTC must be c
in conjunction with CPU processing by means of interrupts.

Table 17.9 shows some examples of processing using the DTC. These examples assume t
number of transfer data bytes is known in slave mode.

transmission/ reception	DTC (ICDRT write)	DTC (ICDRT read)	DTC (ICDRT write)	(ICDRT rea
Dummy data (H'FF) write	—	—	Processing by DTC (ICDR write)	—
Last frame processing	Not necessary	Reception by CPU (ICDR read)	Not necessary	Reception (ICDR rea
Transfer request processing after last frame processing	1st time: Clearing by CPU 2nd time: Stop condition issuance by CPU	Not necessary	Automatic clearing on detection of stop condition during transmission of dummy data (H'FF)	Not neces
Setting of number of DTC transfer data frames	Transmission: Actual data count + 1 (+1 equivalent to slave address + R/$\overline{W}$ bits)	Reception: Actual data count	Transmission: Actual data count + 1 (+1 equivalent to dummy data (H'FF))	Reception data coun

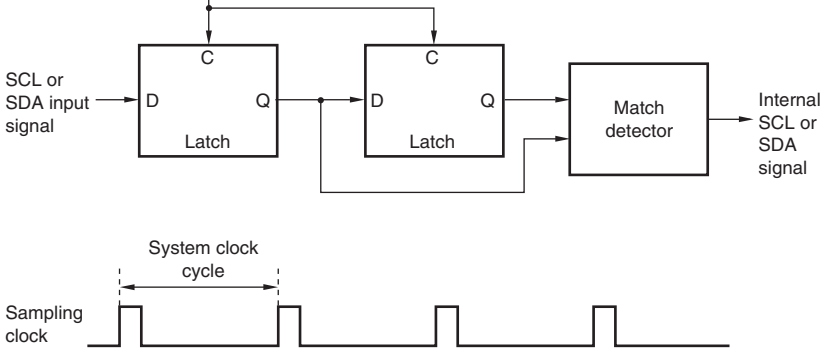


Figure 17.28 Block Diagram of Noise Canceler

17.4.10 Initialization of Internal State

The IIC has a function for forcible initialization of its internal state if a deadlock occurs d
communication.

Initialization is executed in accordance with clearing ICE bit.

Scope of Initialization: The initialization executed by this function covers the following

- ICDRE and ICDRF internal flags
- Transmit/receive sequencer and internal operating clock counter
- Internal latches for retaining the output state of the SCL and SDA pins (wait, clock, d
output, etc.)

- Interrupt flags and interrupt sources are not cleared, and so flag clearing measures m taken as necessary.
- Basically, other register flags are not cleared either, and so flag clearing measures m taken as necessary.
- If a flag clearing setting is made during transmission/reception, the IIC module will s transmitting/receiving at that point and the SCL and SDA pins will be released. Whe transmission/reception is started again, register initialization, etc., must be carried ou necessary to enable correct communication as a system.

The value of the BBSY bit cannot be modified directly by this module clear function, bu stop condition pin waveform is generated according to the state and release timing of the SDA pins, the BBSY bit may be cleared as a result. Similarly, state switching of other b flags may also have an effect.

To prevent problems caused by these factors, the following procedure should be used wh initializing the IIC state.

1. Execute initialization of the internal state according to the ICE bit clearing.
2. Execute a stop condition issuance instruction (write 0 to BBSY and SCP) to clear the bit to 0, and wait for two transfer rate clock cycles.
3. Re-execute initialization of the internal state according to the ICE bit clearing.
4. Initialize (re-set) the IIC registers.

			request		
3	IICI3	IEIC	I^2C bus interface interrupt request	IRIC	Possible
0	IICI0	IEIC	I^2C bus interface interrupt request	IRIC	Possible
1	IICI1	IEIC	I^2C bus interface interrupt request	IRIC	Possible
4	IICI4	IEIC	I^2C bus interface interrupt request	IRIC	Not possible
5	IICI5	IEIC	I^2C bus interface interrupt request	IRIC	Not possible

2. Either of the following two conditions will start the next transfer. Pay attention to the conditions when accessing to ICDR.
   — Write to ICDR when ICE = 1 and TRS = 1 (including automatic transfer from IC ICDRS)
   — Read from ICDR when ICE = 1 and TRS = 0 (including automatic transfer from ICDRR)
3. Table 17.11 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 17.11 I^2C Bus Timing (SCL and SDA Outputs)

Item	Symbol	Output Timing	Unit	No
SCL output cycle time	t_{SCLO}	28 t_{cyc} to 512 t_{cyc}	ns	Se
SCL output high pulse width	t_{SCLHO}	0.5 t_{SCLO}	ns	26 (re
SCL output low pulse width	t_{SCLLO}	0.5 t_{SCLO}	ns	
SDA output bus free time	t_{BUFO}	0.5 t_{SCLO} – 1 t_{cyc}	ns	
Start condition output hold time	t_{STAHO}	0.5 t_{SCLO} – 1 t_{cyc}	ns	
Retransmission start condition output setup time	t_{STASO}	1 t_{SCLO}	ns	
Stop condition output setup time	t_{STOSO}	0.5 t_{SCLO} + 2 t_{cyc}	ns	
Data output setup time (master)	t_{SDASO}	1 t_{SCLLO} – 3 t_{cyc}	ns	
Data output setup time (slave)		1 t_{SCLLO} – (6 t_{cyc} or 12 t_{cyc}*)		
Data output hold time	t_{SDAHO}	3 t_{cyc}	ns	

Note: * 6 t_{cyc} when IICXn is 0, 12 t_{cyc} when IICXn is 1 (n = 0 to 5).

Table 17.12 Permissible SCL Rise Time (t_{sr}) Values

				Time Indication [ns]			
TCSS	IICXn	t_{cyc} Indication		I^2C Bus Specification (Max.)	ϕ = 20 MHz	ϕ = 25 MHz	ϕ =
0	0	7.5 t_{cyc}	Standard mode	1000	375	300	22
			High-speed mode	300	300	300	22
	1	17.5 t_{cyc}	Standard mode	1000	875	700	51
1	0		High-speed mode	300	300	300	30
1	1	37.5 t_{cyc}	Standard mode	1000	1000	1000	10
			High-speed mode	300	300	300	30

Note: n = 0 to 5

t_{SCLLO} in high-speed mode and t_{STASO} in standard mode fail to satisfy the I²C bus interfa
specifications for worst-case calculations of t_{Sr}/t_{Sf}. Possible solutions that should be
investigated include (a) adjusting the rise and fall times by means of a pull-up resisto
capacitive load, (b) reducing the transfer rate to meet the specifications, or (c) selecti
whose input timing permits this output timing for use as slave devices connected to t
bus.

t_{SCLLO}	0.5 t_{SCLO} ($-t_{Sr}$)	Standard mode	–250	4700	4750	4230	4
		High-speed mode	–250	1300	950*[1]	870*[1]	9
t_{BUFO}	0.5 t_{SCLO} –1 t_{cyc} ($-t_{Sr}$)	Standard mode	–1000	4700	3950*[1]	3440*[1]	3
		High-speed mode	–300	1300	850*[1]	780*[1]	8
t_{STAHO}	0.5 t_{SCLO} –1 t_{cyc} ($-t_{Sr}$)	Standard mode	–250	4000	4700	4190	4
		High-speed mode	–250	600	900	830	8
t_{STASO}	1 t_{SCLO} ($-t_{Sr}$)	Standard mode	–1000	4700	9000	7960	8
		High-speed mode	–300	600	2100	1940	2
t_{STOSO}	0.5 t_{SCLO} + 2 t_{cyc} ($-t_{Sr}$)	Standard mode	–1000	4000	4100	3560	3
		High-speed mode	–300	600	1000	900	9
t_{SDASO} (master)	1 t_{SCLLO}*[3] –3 t_{cyc} ($-t_{Sr}$)	Standard mode	–1000	250	3600	3110	3
		High-speed mode	–300	100	500	450	5
t_{SDASO} (slave)	1 t_{SCLL}*[3] –12 t_{cyc}*[2] ($-t_{Sr}$)	Standard mode	–1000	250	3100	3220	3
		High-speed mode	–300	100	400	520	6
t_{SDAHO}	3 t_{cyc}	Standard mode	0	0	150	120	8
		High-speed mode	0	0	150	120	8

Notes: 1. Does not meet the I^2C bus interface specification. Remedial action such as the is necessary: (a) secure a start/stop condition issuance interval; (b) adjust the fall times by means of a pull-up resistor and capacitive load; (c) reduce the tra (d) select slave devices whose input timing permits this output timing.

The values in the above table will vary depending on the settings of the bits TC IICX3 to IICX0 and CKS2 to CKS0. Depending on the frequency it may not be to achieve the maximum transfer rate; therefore, whether or not the I^2C bus int specifications are met must be determined in accordance with the actual settin conditions.

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 confir 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 17.29 (after confirming that the BBS been cleared to 0 in the ICCR register).

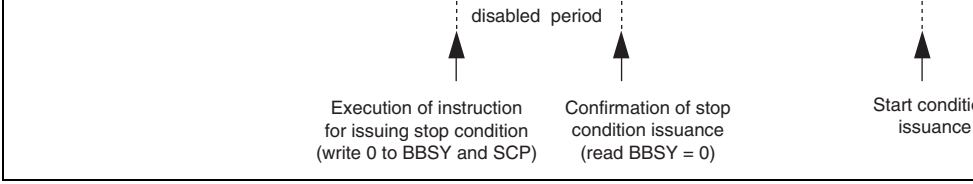


Figure 17.29 Notes on Reading Master Receive Data

Note: This restriction on usage can be canceled by setting the FNC1 and FNC0 bits to B
ICXR.

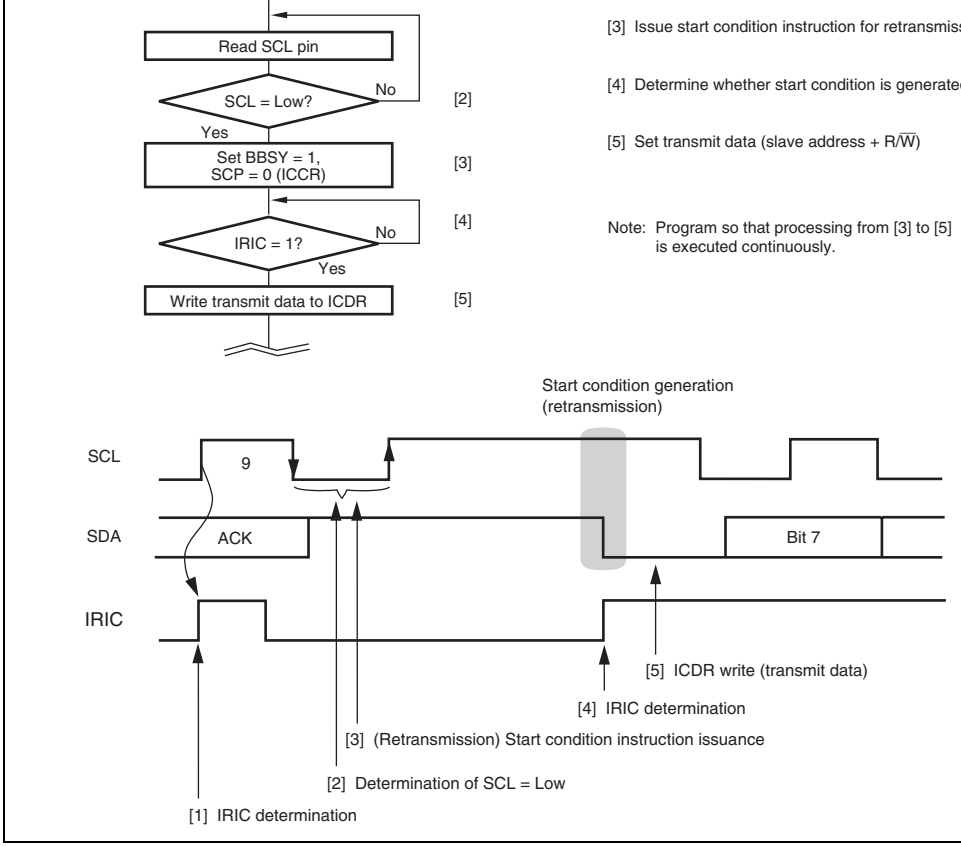


Figure 17.30 Flowchart for Start Condition Issuance Instruction for Retransmission and Timing

Note: This restriction on usage can be canceled by setting the FNC1 and FNC0 bits to ICXR.

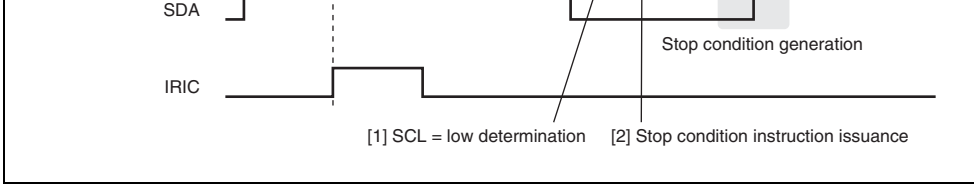


Figure 17.31 Stop Condition Issuance Timing

Note: This restriction on usage can be canceled by setting the FNC1 and FNC0 bits to E 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 17.32 IRIC Flag Clearing Timing When WAIT = 1

Note: This restriction on usage can be canceled by setting the FNC1 and FNC0 bits to ICXR.

11. Note on ICDR register read and ICCR register access in slave transmit mode

In I^2C bus interface slave transmit mode, do not read ICDR or do not read/write from during the time shaded in figure 17.33. However, such read and write operations sou problem in interrupt handling processing that is generated in synchronization with th edge of the 9th clock pulse because the shaded time has passed before making the tra interrupt handling.

To handle interrupts securely, be sure to keep either of the following conditions.

— Read ICDR data that has been received so far or read/write from/to ICCR before the receive operation of the next slave address.

— Monitor the BC2 to BC0 counter in ICMR; when the count is B'000 (8th or 9th 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.

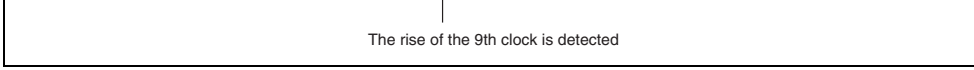


Figure 17.33 ICDR Register Read and ICCR Register Access Timing in Slave Transmit Mode

Note: This restriction on usage can be canceled by setting the FNC1 and FNC0 bits to B ICXR.

12. Note on TRS bit setting in slave mode

In I^2C bus interface slave mode, if the TRS bit value in ICCR is set after detecting the edge of the 9th clock pulse or the stop condition before detecting the next rising edge SCL pin (the time indicated as (a) in figure 17.34), the bit value becomes valid immed when it is set. However, if the TRS bit is set during the other time (the time indicated figure 17.34), the bit value is suspended and remains invalid until the rising edge of th clock pulse or the stop condition is detected. Therefore, when the address is received a restart condition is input without the stop condition, the effective TRS bit value remai (transmit mode) internally and thus the acknowledge bit is not transmitted after the ad been received at the 9th clock pulse.

To receive the address in slave mode, clear the TRS bit to 0 during the time indicated figure 17.34. To release the SCL low level that is held by means of the wait function i mode, clear the TRS bit to and then dummy-read ICDR.

Figure 17.34 TRS Bit Set Timing in Slave Mode

Note: This restriction on usage can be canceled by setting the FNC1 and FNC0 bits to 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 mo 0), the SCL pin may not be held low in some cases after transmit/receive operation h completed, thus inconveniently allowing clock pulses to be output on the SCL bus li ICDR is accessed correctly. To access ICDR correctly, read the ICDR after setting r mode or write to the ICDR after setting transmit mode.

of a series of transmit operation, clear the ACKE bit in ICCR once to initialize the bit to 0.

— Set receive mode (TRS = 0) before the next start condition is input in slave mode. Complete transmit operation by the procedure shown in figure 17.23, in order to s 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 when arbitration is lost in master mode and a transition to slave receive mode is automatical carried out.

When arbitration is lost not in the first frame but in the second frame or subsequent fr 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 SAR register, the I^2C bus interface erroneously recognizes that the address call has occurre figure 17.35.)

In multi-master mode, a bus conflict could happen. When the I^2C bus interface is oper master mode, check the state of the AL bit in the ICSR register every time after one fr data has been transmitted or received.

When arbitration is lost during transmitting the second frame or subsequent frame, tak avoidance measures.

Figure 17.35 Diagram of Erroneous Operation when Arbitration Lost

Though it is prohibited in the normal I^2C protocol, the same problem may occur whe bit is erroneously set to 1 and a transition to master mode is occurred during data tra or reception in slave mode.

When the MST bit is set to 1 during data transmission or reception in slave mode, th 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 recognize address.

In multi-master mode, pay attention to the setting of the MST bit when a bus conflic occur. In this case, the MST bit in the ICCR register should be set to 1 according to t below.

A. Make sure that the BBSY flag in the ICCR register is 0 and the bus is free before 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 be check that the BBSY flag in the ICCR register is 0 immediately after the MST bi set.

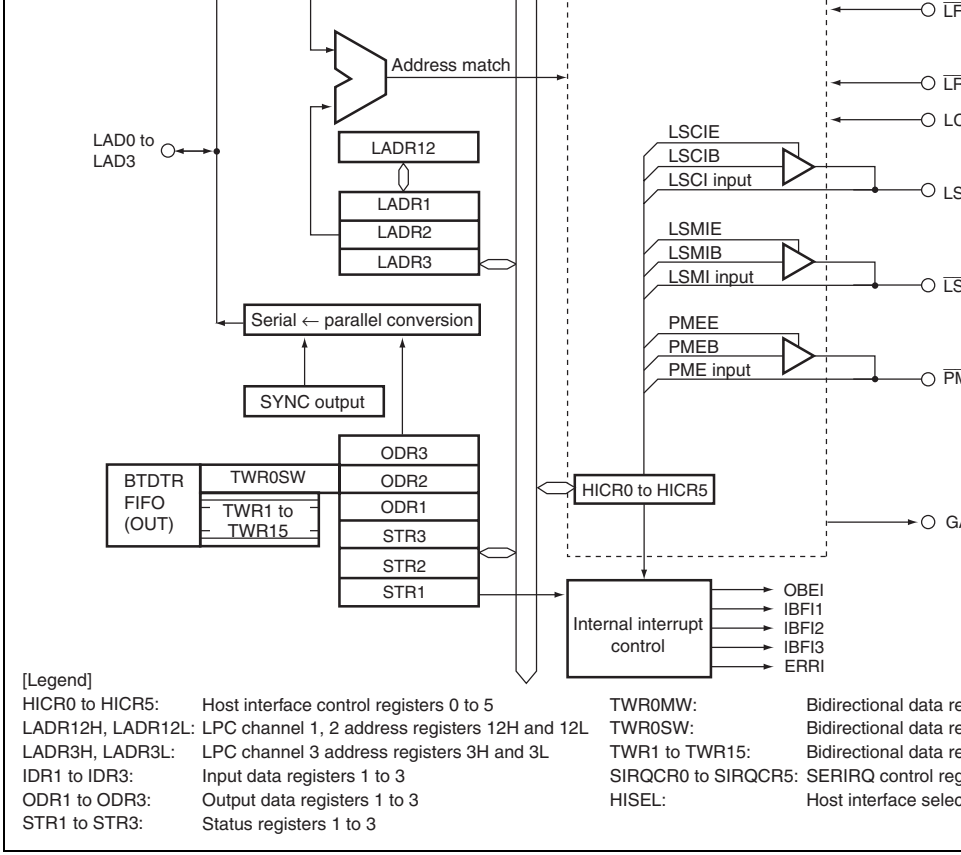
Note: Above restrictions can be released by setting the bits FNC1 and FNC2 in ICXR

18.1 Features

- Supports LPC interface I/O read and I/O write cycles
  - Uses four signal lines (LAD3 to LAD0) to transfer the cycle type, address, and d
  - Uses three control signals: clock (LCLK), reset ($\overline{\text{LRESET}}$), and frame ($\overline{\text{LFRAME}}$
- Three register sets comprising data and status registers
  - The basic register set comprises three bytes: an input register (IDR), output regis and status register (STR).
  - I/O addresses from H'0000 to H'FFFF are selected for channels 1 to 3.
  - A fast Gate A20 function is provided for channel 1.
  - For channel 3, sixteen bidirectional data register bytes can be manipulated in add the basic register set.
- Supports SCIF
  - The LPC interface is connected to the SCIF, allowing direct control of the SCIF LPC host.

Three pins, PME, ESMI, and ESCI, are provided for general input/output.

- Supports version 1.5 of the Intelligent Platform Management Interface (IPMI) specifi
  - Channel 3 supports the SMIC interface, KCS interface, and BT interface.



[Legend]

HICR0 to HICR5:	Host interface control registers 0 to 5	TWR0MW:	Bidirectional data re
LADR12H, LADR12L:	LPC channel 1, 2 address registers 12H and 12L	TWR0SW:	Bidirectional data re
LADR3H, LADR3L:	LPC channel 3 address registers 3H and 3L	TWR1 to TWR15:	Bidirectional data re
IDR1 to IDR3:	Input data registers 1 to 3	SIRQCR0 to SIRQCR5:	SERIRQ control reg
ODR1 to ODR3:	Output data registers 1 to 3	HISEL:	Host interface selec
STR1 to STR3:	Status registers 1 to 3		

Figure 18.1 Block Diagram of LPC

				termination signal
LPC reset	$\overline{\text{LRESET}}$	PE5	Input*[1]	LPC interface reset signal
LPC clock	LCLK	PE6	Input	33-MHz PCI clock signal
Serialized interrupt request	SERIRQ	PE7	I/O*[1]	Serialized host interrupt req signal (SMI, HIRQ1 to HIR synchronization with LCLK
LSCI general output	LSCI	PD0	Output*[1, 2]	General output
LSMI general output	$\overline{\text{LSMI}}$	PD1	Output*[1, 2]	General output
PME general output	$\overline{\text{PME}}$	PD2	Output*[1, 2]	General output
GATE A20	GA20	PD3	Output*[1, 2]	Gate A20 control signal ou
LPC clock run	$\overline{\text{CLKRUN}}$	PD4	I/O*[1, 2]	LCLK restart request signa serial host interrupt is requ
LPC power-down	$\overline{\text{LPCPD}}$	PD5	Input*[1]	LPC module shutdown sign

Notes: 1. Pin state monitoring input is possible in addition to the LPC interface control input/output function.

2. Only 0 can be output. If 1 is output, the pin is in the high-impedance state, so a external resistor is necessary to pull the signal up to VCC.

- Pin function control register (PINFNCR)
- LPC channel 1, 2 address register H, L (LADR12H, LADR12L)
- LPC channel 3 address register H, L (LADR3H, LADR3L)
- Input data register 1 (IDR1)
- Input data register 2 (IDR2)
- Input data register 3 (IDR3)
- Output data register 1 (ODR1)
- Output data register 2 (ODR2)
- Output data register 3 (ODR3)
- Status register 1 (STR1)
- Status register 2 (STR2)
- Status register 3 (STR3)
- Bidirectional data registers 0 to 15 (TWR0 to TWR15)
- SERIRQ control register 0 (SIRQCR0)
- SERIRQ control register 1 (SIRQCR1)
- SERIRQ control register 2 (SIRQCR2)
- SERIRQ control register 3 (SIRQCR3)
- SERIRQ control register 4 (SIRQCR4)
- SERIRQ control register 5 (SIRQCR5)
- Host interface select register (HISEL)
- SCIF address register H, L (SCIFADRH, SCIFADRL)

- BT status register 1 (BTSR1)
- BT control/status register 0 (BTCSR0)
- BT control/status register 1 (BTCSR1)
- BT control register (BTCR)
- BT data buffer (BTDTR)
- BT interrupt mask register (BTIMSR)
- FIFO valid size register 0 (BTFVSR0)
- FIFO valid size register 1 (BTFVSR1)

6	LPC2E	0	R/W	—	Enable or disable the LPC interface function. LPC interface is enabled (one of the three bit 1), processing for data transfer between the s LSI) and the host is performed using pins LA LAD0, $\overline{\text{LFRAME}}$, $\overline{\text{LRESET}}$, LCLK, SERIRQ, $\overline{\text{C}}$ and $\overline{\text{LPCPD}}$. • LPC3E 0: LPC channel 3 operation is disabled No address (LADR3) matches for IDR3, C STR3, TWR0 to TWR15, SMIC, KCS, or 1: LPC channel 3 operation is enabled • LPC2E 0: LPC channel 2 operation is disabled No address (LADR2) matches for IDR2, 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
5	LPC1E	0	R/W	—	

					1: Fast Gate A20 function enabled GA20 pin output is open-drain (external pu resistor (Vcc) required)
3	SDWNE	0	R/W	—	LPC Software Shutdown Enable Controls LPC interface shutdown. For details o LPC shutdown function, and the scope of initia by an LPC reset and an LPC shutdown, see s 18.4.6, LPC Interface Shutdown Function (LP 0: Normal state, LPC software shutdown sett enabled [Clearing conditions] • Writing 0 • LPC hardware reset or LPC software reset • LPC hardware shutdown release (rising ed $\overline{\text{LPCPD}}$ signal) 1: LPC hardware shutdown state setting enal Hardware shutdown state when $\overline{\text{LPCPD}}$ si low level [Setting condition] Writing 1 after reading SDWNE = 0

					1 0 : PME output enabled, PME output goes to 0 level 1 1 : PME output enabled, PME output is high-impedance
1	LSMIE	0	R/W	—	LSMI output Enable Controls LSMI output in combination with the bit in HICR1. LSMI pin output is open-drain, a external pull-up resistor (Vcc) is needed. The bit should be cleared to 0 when the LPC is us LSMIE LSMIB 0 X : LSMI output disabled; gen function of pin PD1 is enal 1 0 : LSMI output enabled, LSM output goes to 0 level 1 1 : LSMI output enabled, LSM output is Hi-Z

	1	0	: LSCI output enabled, LSCI output goes to 0 level
	1	1	: LSCI output enabled, LSCI output is high-impedance

[Legend]
X: Don't care

- Cycle type or address indeterminate durin cycle

[Clearing conditions]

- LPC hardware reset or LPC software rese
- LPC hardware shutdown or LPC software shutdown
- Forced termination (abort) of transfer cycl to processing
- Normal termination of transfer cycle subje processing

1: LPC interface is performing transfer cycle processing

[Setting condition]

Match of cycle type and address

					• SERIRQ is set to continuous mode • There are no further interrupts for transfer host in quiet mode 1: LCLK restart request issued [Setting condition] In quiet mode, SERIRQ interrupt output becon necessary while LCLK is stopped
5	IRQBSY	0	R	—	SERIRQ Busy Indicates that the LPC interface's SERIRQ is e in transfer processing. 0: SERIRQ transfer frame wait state [Clearing conditions] • LPC hardware reset or LPC software reset • LPC hardware shutdown or LPC software shutdown • End of SERIRQ transfer frame 1: SERIRQ transfer processing in progress [Setting condition] Start of SERIRQ transfer frame

					1: LPC software reset state [Setting condition] Writing 1 after reading LRSTB = 0
3	SDWNB	0	R/W	—	LPC Software Shutdown Bit Controls LPC interface shutdown. For details LPC shutdown function, and the scope of init by an LPC reset and an LPC shutdown, see 18.4.6, LPC Interface Shutdown Function (LP 0: Normal state [Clearing conditions] • Writing 0 • LPC hardware reset or LPC software rese • LPC hardware shutdown (falling edge of $\overline{\text{LPCPD}}$ signal when SDW • LPC hardware shutdown release (rising edge of $\overline{\text{LPCPD}}$ signal when SDW 1: LPC software shutdown state [Setting condition] Writing 1 after reading SDWNB = 0
2	PMEB	0	R/W	—	PME Output Bit Controls PME output in combination with the bit. For details, refer to description on the PM HICR0.

Bit	Bit Name	Value	Slave	Host	Description
7	GA20	Undefined	R	—	GA20 Pin Monitor
6	LRST	0	R/(W)*	—	LPC Reset Interrupt Flag This bit is a flag that generates an ERRI inte when an LPC hardware reset occurs. 0: [Clearing condition] Writing 0 after reading LRST = 1 1: [Setting condition] $\overline{\text{LRESET}}$ pin falling edge detection
5	SDWN	0	R/(W)*	—	LPC Shutdown Interrupt Flag This bit is a flag that generates an ERRI inte when an LPC hardware shutdown request is generated. 0: [Clearing conditions] • Writing 0 after reading SDWN = 1 • LPC hardware reset ($\overline{\text{LRESET}}$ pin falling edge detection) • LPC software reset (LRSTB = 1) 1: [Setting condition] $\overline{\text{LPCPD}}$ pin falling edge detection

					• LPC software reset (LRSTB = 1) • LPC hardware shutdown (SDWNE = 1 and $\overline{\text{LPCPD}}$ pin falling edge detection) • LPC software shutdown (SDWNB = 1) 1: [Setting condition] $\overline{\text{LFRAME}}$ pin falling edge detection during LP transfer cycle
3	IBFIE3	0	R/W	—	IDR3 and TWR Receive Complete interrupt E Enables or disables IBFI3 interrupt to the sla LSI). 0: Input data register (IDR3) and TWR recei complete interrupt requests and SMIC/BT interrupt requests disabled 1: [When TWRIE = 0 in LADR3] Input data register (IDR3) receive com interrupt requests and SMIC/BT mode interrupt requests enabled [When TWRIE = 1 in LADR3] Input data register (IDR3) and TWR re complete interrupt requests and SMIC mode interrupt requests enabled

					Enables or disables IBFI1 interrupt to the sl LSI). 0: Input data register (IDR1) receive comple interrupt requests disabled 1: Input data register (IDR1) receive comple interrupt requests enabled
0	ERRIE	0	R/W	—	Error Interrupt Enable Enables or disables ERRI interrupt to the sl LSI). 0: Error interrupt requests disabled 1: Error interrupt requests enabled

Note: * Only 0 can be written to bits 6 to 4, to clear the flag.

4	LRESET	Undefined	R	—	0: $\overline{\text{LRESET}}$ Pin state is low level 1: $\overline{\text{LRESET}}$ Pin state is high level
3	LPCPD	Undefined	R	—	0: $\overline{\text{LPCPD}}$ Pin state is low level 1: $\overline{\text{LPCPD}}$ Pin state is high level
2	PME	Undefined	R	—	0: $\overline{\text{PME}}$ Pin state is low level 1: $\overline{\text{PME}}$ Pin state is high level
1	LSMI	Undefined	R	—	0: $\overline{\text{LSMI}}$ Pin state is low level 1: $\overline{\text{LSMI}}$ Pin state is high level
0	LSCI	Undefined	R	—	0: LSCI Pin state is low level 1: LSCI Pin state is high level

18.3.3 Host Interface Control Register 4 (HICR4)

HICR4 controls the operation of the KCS, SMIC, and BT interface functions on channel

			R/W		
Bit	**Bit Name**	**Initial Value**	**Slave**	**Host**	**Description**
7	LADR12SEL	0	R/W	—	Switches the channel accessed via LADR LADR12L. 0: LADR1 is selected 1: LADR2 is selected
6 to 4	—	All 0	R/W	—	Reserved The initial value should not be changed.

					0: KCS interface operation is disabled No address (LADR3) matches for IDR 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

					0: Disables access to the SCIF from the LP
					1: Enables access to the SCIF from the LP
0	—	0	R/W	—	Reserved The initial value should not be changed.

18.3.5 Pin Function Control Register (PINFNCR)

PINFNCR selects whether the pins of the associated port are used for the LPC function o
I/O.

Bit	Bit Name	Initial Value	R/W		Description
			Slave	Host	
7 to 3	—	All 0	R/W	—	Reserved The initial value bit should not be changed.
2	SERIRQOFF	0	R/W	—	0: SERIRQ pin 1: General I/O port
1	LPCPDOFF	0	R/W	—	0: LPCPD pin 1: General I/O port
0	CLKRUNOFF	0	R/W	—	0: CLKRUN pin 1: General I/O port

2 is operating (while LPC2E is set to 1).

Table 18.2 shows the initial value of each register. Table 18.3 shows the host register se
address match determination. Table 18.4 shows the slave selection internal registers in s
LSI) access.

Table 18.2 LADR1, LADR2 Initial Values

Register Name	Initial Value	Description
LADR1	H'0060	I/O address of channel 1
LADR2	H'0062	I/O address of channel 2

LADR2 (bits 15 to 3)	0	LADR2 (bit 1)	LADR2 (bit 0)	I/O write	IDR2 write (data) C/$\overline{D}$2 ← 0
LADR2 (bits 15 to 3)	1	LADR2 (bit 1)	LADR2 (bit 0)	I/O write	IDR2 write (comm C/$\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

Table 18.4 Slave Selection Internal Registers

Slave (R/W)	Bus Width (B/W)	LADR12SEL	LADR12		Internal Reg	
R/W	B	0	LADR12H		LADR1H	
R/W	B	1	LADR12H		LADR2H	
R/W	B	0		LADR12L		LAD
R/W	B	1		LADR12L		LAD
R/W	W	0	LADR12H	LADR12L	LADR1H	LAD
R/W	W	1	LADR12H	LADR12L	LADR2H	LAD

6	Bit 14				
5	Bit 13				
4	Bit 12				
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 se
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 se
0	TWRE	0	R/W	—	Bidirectional data Register Enable
					Enables or disables bidirectional data re operation.
					Clear this bit to 0 in KCS mode.
					0: TWR operation is disabled
					TWR-related address (LADR3) match occur.
					1: TWR operation is enabled

I/O Address						Transfer	Host Register
Bits 15 to5	**Bit 4**	**Bit 3**	**Bit 2**	**Bit 1**	**Bit 0**	**Cycle**	**Selection**
Bits 15 to5	Bit 4	Bit 3	0	Bit 1	0	I/O write	IDR3 write, C
Bits 15 to5	Bit 4	Bit 3	1	Bit 1	0	I/O write	IDR3 write, C
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 wr
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 rea
Bits 15 to5	$\overline{\text{Bit 4}}$	0	0	0	1	I/O read	TWR1 to TW read
		•	•	•	•		
		•	•	•	•		
		•	•	•	•		
		1	1	1	1		

I/O Address						Transfer	
Bits 15 to5	**Bit 4**	**Bit 3**	**Bit 2**	**Bit 1**	**Bit 0**	**Cycle**	**Host Register Sele**
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 Sele**
Bits 15 to5	Bit 4	1	0	0	1	I/O write	SMICDTR write
Bits 15 to5	Bit 4	1	0	1	0	I/O write	SMICCSR write
Bits 15 to5	Bit 4	1	0	1	1	I/O write	SMICFLG write
Bits 15 to5	Bit 4	1	0	0	1	I/O read	SMICDTR read
Bits 15 to5	Bit 4	1	0	1	0	I/O read	SMICCSR read
Bits 15 to5	Bit 4	1	0	1	1	I/O read	SMICFLG read

The initial values of the IDR registers are undefined.

18.3.9 Output Data Registers 0 to 3 (ODR1 to ODR3)

The ODR registers are 8-bit readable/writable registers to the slave processor (this LSI), a
read-only registers to the host processor. The registers selected from the host according to
address are described in the following sections: for information on ODR1 and ODR2 sele
section 18.3.6, LPC Channel 1, 2 Address Register H, L (LADR12H, LADR12L), and fo
information on ODR3 selection, see section 18.3.7, LPC Channel 3 Address Register H, L
(LADR3H, LADR3L). In an LPC I/O read cycle, the data in the selected register is transf
the host.

The initial values of the ODR registers are undefined.

the I/O address, see section 18.3.7, LPC Channel 3 Address Register H, L (LADR3H, L

Data transferred in an LPC I/O write cycle is written to the selected register; in an LPC I
cycle, the data in the selected register is transferred to the host.

The initial values of TWR0 to TWR15 are undefined.

Address Register H, L (LADR12H, LADR12L), and information on STR3 selection, see 18.3.7, LPC Channel 3 Address Register H, L (LADR3H, LADR3L). In an LPC I/O read 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 (when CH1OFFSEL bit 0 of the I/O address (when CH1OFFSEL written to this bit to indicate whether IDR1 c data or a command.
					0: Content of input data register (IDR1) is da
					1: Content of input data register (IDR1) is a command
2	DBU12	0	R/W	R	Defined by User
					The user can use this bit as necessary.

					When the slave processor reads IDR 1: There is receive data in IDR1 [Setting condition] When the host processor writes to IDR usi write cycle
0	OBF1	0	R/(W)*	R	Output Data Register Full Indicates whether or not there is transmit d ODR1. 0: There is not transmit data in ODR1 [Clearing condition] When the host processor reads ODR1 usi read cycle, or the slave processor writes 0 OBF1 bit 1: There is transmit data in ODR1 [Setting condition] When the slave processor writes to ODR1

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

					address (when CH2OFFSEL1 = 1) is written bit to indicate whether IDR2 contains data o command. 0: Content of input data register (IDR2) is a 1: Content of input data register (IDR2) is a command
2	DBU22	0	R/W	R	Defined by User The user can use this bit as necessary.
1	IBF2	0	R	R	Input Data Register Full Indicates whether or not there is receive dat IDR2.This bit is an internal interrupt source f slave (this LSI). 0: There is not receive data in IDR2 [Clearing condition] When the slave reads IDR2 1: There is receive data in IDR2 [Setting condition] When the host writes to IDR2 in an I/O write

[Setting condition]

- When the slave writes to ODR2

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

					When the host writes to TWR15 in I/O write
6	OBF3B	0	R/(W)*	R	Bidirectional Data Register Output Buffer Fu 0: [Clearing conditions] • When the host reads TWR15 in I/O read • When the slave writes 0 to the OBF3B b 1: [Setting condition] When the slave writes to TWR15
5	MWMF	0	R	R	Master Write Mode Flag 0: [Clearing condition] When the slave reads TWR15 1: [Setting condition] When the host writes to TWR0 in I/O write c while SWMF = 0
4	SWMF	0	R/(W)*	R	Slave Write Mode Flag In the event of simultaneous writes by the m and the slave, the master write has priority. 0: [Clearing conditions] • When the host reads TWR15 in I/O read • When the slave writes 0 to the SWMF bi 1: [Setting condition] When the slave writes to TWR0 while MWM

					The user can use this bit as necessary.
1	IBF3A	0	R	R	Input Data Register Full Indicates whether or not there is receive d IDR3. This is an internal interrupt source t slave (this LSI). 0: There is not receive data in IDR3 [Clearing condition] When the slave reads IDR3 1: There is receive data in IDR3 [Setting condition] When the host writes to IDR3 in an I/O wr
0	OBF3A	0	R/(W)*	R	Output Data Register Full Indicates whether or not there is transmit ODR3. 0: There is not transmit data in ODR3 [Clearing conditions] • When the host reads ODR3 in an I/O r • When the slave writes 0 to bit OBF3A 1: There is transmit data in ODR3 [Setting condition] • When the slave writes to ODR3

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

					address is written into this bit to indicate wh IDR3 contains data or a command. 0: Content of input data register (IDR3) is a 1: Content of input data register (IDR3) is a command
2	DBU32	0	R/W	R	Defined by User The user can use this bit as necessary.
1	IBF3A	0	R	R	Input Data Register Full Indicates whether or not there is receive dat IDR3. This bit is an internal interrupt source slave (this LSI). 0: There is not receive data in IDR3 [Clearing condition] When the slave reads IDR3 1: There is receive data in IDR3 [Setting condition] When the host writes to IDR3 in an I/O write

[Setting condition]

- When the slave writes to ODR3

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

					0: Continuous mode [Clearing conditions] • LPC hardware reset, LPC software reset • Specification by SERIRQ transfer cycle stop frame 1: Quiet mode [Setting condition] Specification by SERIRQ transfer cycle stop
6	SELREQ	0	R/W	—	Start Frame Initiation Request Select Selects the condition of a start frame initiation request when a host interrupt request is cleared in quiet mode. 0: Start frame initiation is requested when all interrupt requests are cleared 1: Start frame initiation is requested when one or more interrupt requests are cleared
5	IEDIR2	0	R/W	—	Interrupt Enable Direct Mode Specifies whether LPC channel 2 and channel SERIRQ interrupt source (SMI, IRQ6, IRQ9 to IRQ11) generation is conditional upon OBF, or controlled only by the host interrupt enable bit. 0: Host interrupt is requested when host interrupt enable and corresponding OBF bits are both 1 1: Host interrupt is requested when host interrupt enable bit is set to 1

					• Clearing OBF3B to 0 (when IEDIR3 = 0 1: [When IEDIR3 = 0] Host SMI interrupt request by setting O is enabled [When IEDIR3 = 1] Host SMI interrupt is requested [Setting condition] Writing 1 after reading SMIE3B = 0
3	SMIE3A	0	R/W	—	Host SMI Interrupt Enable 3A Enables or disables an SMI interrupt reque OBF3A is set by an ODR3 write. 0: Host SMI interrupt request by OBF3A a 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 is [When IEDIR3 = 1] Host SMI interrupt is requested [Setting condition] Writing 1 after reading SMIE3A = 0

					• Clearing OBF2 to 0 (when IEDIR2 = 0) 1: [When IEDIR2 = 0] Host SMI interrupt request by setting OB is enabled [When IEDIR2 = 1] Host SMI interrupt is requested [Setting condition] Writing 1 after reading SMIE2 = 0
1	IRQ12E1	0	R/W	—	Host IRQ12 Interrupt Enable 1 Enables or disables an HIRQ12 interrupt rec when OBF1 is set by an ODR1 write. 0: HIRQ12 interrupt request by OBF1 and IB 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

- Clearing OBF1 to 0

1: HIRQ1 interrupt request by setting OBF
enabled

[Setting condition]

Writing 1 after reading IRQ1E1 = 0

					IRQE11E3 is disabled [Clearing conditions] • Writing 0 to IRQ11E3 • LPC hardware reset, LPC software reset • Clearing OBF3A to 0 (when IEDIR3 = 0) 1: [When IEDIR3 = 0] 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 an HIRQ10 interrupt req when OBF3A is set by an ODR3 write. 0: HIRQ10 interrupt request by OBF3A and IRQE10E3 is disabled [Clearing conditions] • Writing 0 to IRQ10E3 • LPC hardware reset, LPC software reset • Clearing OBF3A 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

					• Clearing OBF3A to 0 (when IEDIR3 = 0 1: [When IEDIR3 = 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 an HIRQ6 interrupt req when OBF3A is set by an ODR3 write. 0: HIRQ6 interrupt request by OBF3A and is disabled [Clearing conditions] • Writing 0 to IRQ6E3 • LPC hardware reset, LPC software 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

					• Clearing OBF2 to 0 (when IEDIR2 = 0) 1: [When IEDIR2 = 0] HIRQ11 interrupt request by setting OBF enabled [When IEDIR2 = 1] HIRQ11 interrupt is requested [Setting condition] Writing 1 after reading IRQ11E2 = 0
2	IRQ10E2	0	R/W	—	Host IRQ10 Interrupt Enable 2 Enables or disables an HIRQ10 interrupt req when OBF2 is set by an ODR2 write. 0: HIRQ10 interrupt request by OBF2 and IRQE10E2 is disabled [Clearing conditions] • Writing 0 to IRQ10E2 • LPC hardware reset, LPC software reset • Clearing OBF2 to 0 (when IEDIR2 = 0) 1: [When IEDIR2 = 0] HIRQ10 interrupt request by setting OBF enabled [When IEDIR2 = 1] HIRQ10 interrupt is requested [Setting condition] Writing 1 after reading IRQ10E2 = 0

					• Clearing OBF2 to 0 (when IEDIR2 = 0) 1: [When IEDIR2 = 0] HIRQ9 interrupt request by setting OBF enabled [When IEDIR2 = 1] HIRQ9 interrupt is requested [Setting condition] Writing 1 after reading IRQ9E2 = 0
0	IRQ6E2	0	R/W	—	Host IRQ6 Interrupt Enable 2 Enables or disables an HIRQ6 interrupt req when OBF2 is set by an oDR2 write. 0: HIRQ6 interrupt request by OBF2 and I is disabled [Clearing conditions] • Writing 0 to IRQ6E2 • LPC hardware reset, LPC software res • Clearing OBF2 to 0 (when IEDIR2 = 0) 1: [When IEDIR2 = 0] HIRQ6 interrupt request by setting OBF enabled [When IEDIR2 = 1] HIRQ6 interrupt is requested [Setting condition] Writing 1 after reading IRQ6E2 = 0

					enable bit. 0: A host interrupt is generated when both enable bit and the corresponding OBF set 1: A host interrupt is generated when the is set
6 to 0	—	All 0	R/W	—	Reserved The initial value should not be changed.

1	SCSIRQ1	0	R/W	—	host.
0	SCSIRQ0	0	R/W	—	0000: No interrupt request to the host 0001: HIRQ1 0010: SMI 0011: HIRQ3 0100: HIRQ4 0101: HIRQ5 0110: HIRQ6 0111: HIRQ7 1000: HIRQ8 1001: HIRQ9 1010: HIRQ10 1011: HIRQ11 1100: HIRQ12 1101: HIRQ13 1110: HIRQ14 1111: HIRQ15

					0: Disables HIRQ14 interrupt request by IRQ
					1: Enables HIRQ14 interrupt request
5	IRQ13E	0	R/W	—	Host IRQ13 Interrupt Enable
					0: Disables HIRQ13 interrupt request by IRQ
					1: Enables HIRQ13 interrupt request
4	IRQ8	0	R/W	—	Host IRQ8 Interrupt Enable
					0: Disables HIRQ8 interrupt request by IRQ8
					1: Enables HIRQ8 interrupt request
3	IRQ7	0	R/W	—	Host IRQ7 Interrupt Enable
					0: Disables HIRQ7 interrupt request by IRQ7
					1: Enables HIRQ7 interrupt request
2	IRQ5	0	R/W	—	Host IRQ5 Interrupt Enable
					0: Disables HIRQ5 interrupt request by IRQ5
					1: Enables HIRQ5 interrupt request
1	IRQ4	1	R/W	—	Host IRQ4 Interrupt Enable
					0: Disables HIRQ4 interrupt request by IRQ4
					1: Enables HIRQ4 interrupt request
0	IRQ3	1	R/W	—	Host IRQ3 Interrupt Enable
					0: Disables HIRQ3 interrupt request by IRQ3
					1: Enables HIRQ3 interrupt request

3	SELIRQ7	0	R/W	—	0: [When host interrupt request is cleared] SERIRQ pin output is in the Hi-Z state [When host interrupt request is set] SERIRQ pin output is low 1: [When host interrupt request is cleared] SERIRQ pin output is low [When host interrupt request is set] SERIRQ pin output is in the Hi-Z state.
2	SELIRQ5	0	R/W	—	
1	SELIRQ4	0	R/W	—	
0	SELIRQ3	0	R/W	—	

					details of STR3, see section 18.3.11, Status Registers 1 to 3 (STR1 to STR3). 0: Bits 7 to 4 in STR3 indicate processing st the LPC interface. 1: [When TWRE = 1] Bits 7 to 4 in STR3 indicate processing st the LPC interface. [When TWRE = 0] Bits 7 to 4 in STR3 are readable/writable which user can use as necessary
6	SELIRQ11	0	R/W	—	Host IRQ Interrupt Select These bits select the state of the output on t SERIRQ pin. 0: [When host interrupt request is cleared] SERIRQ pin output is in the Hi-Z state [When host interrupt request is set] SERIRQ pin output is low 1: [When host interrupt request is cleared] SERIRQ pin output is low [When host interrupt request is set] SERIRQ pin output is in the Hi-Z state.
5	SELIRQ10	0	R/W	—	
4	SELIRQ9	0	R/W	—	
3	SELIRQ6	0	R/W	—	
2	SELSMI	0	R/W	—	
1	SELIRQ12	1	R/W	—	
0	SELIRQ1	1	R/W	—	

6	—	0	R/W	—	These bits set the host address for the SCI
5	—	0	R/W	—	
4	—	0	R/W	—	
3	—	0	R/W	—	
2	—	0	R/W	—	
1	—	1	R/W	—	
0	—	1	R/W	—	

- SCIFADRL

Bit	Bit Name	Initial Value	R/W		Description
			Slave	Host	
7	—	1	R/W	—	SCIF Address 15 to 8
6	—	1	R/W	—	These bits set the host address for the SCI
5	—	1	R/W	—	
4	—	1	R/W	—	
3	—	1	R/W	—	
2	—	0	R/W	—	
1	—	0	R/W	—	
0	—	0	R/W	—	

Note: When the SCIF is in use, SCIFADR must be set to an address that is different fro
for LPC channels 1, 2, and 3.

					0: Slave waits for ready status 1: Slave is ready for the host read tran
6	TX_DATA_RDY	0	R/W	R	Write Transfer Ready Indicates whether or not the slave is r the host next write transfer. 0: The slave waits for ready status 1: The slave is ready for the host write transfer.
5	—	0	R/W	R	Reserved The initial value should not be change
4	SMI	0	R/W	R	SMI Flag This bit indicates that the SMI is asser 0: Indicates waiting for SMI assertion 1: Indicates SMI assertion
3	SEVT_ATN	0	R/W	R	Event Flag When the slave detects an event for th this bit is set. 0: Indicates waiting for event detection 1: Indicates event detection
2	SMS_ATN	0	R/W	R	SMS Flag When there is a message to be transm from the slave to the host, this bit is se 0: There is not a message 1: There is a message

	0: Transfer cycle wait state
	[Clearing conditions]
	After the slave reads BUSY = 1, writes 0 to
	1: Transfer cycle in progress
	[Setting condition]
	When the host writes 1 to this bit.

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

18.3.21 SMIC Control Status Register (SMICCSR)

SMICCSR is one of the registers used to implement SMIC mode. This is an 8-bit readable/writable register that stores a control code issued from the host and a status cod returned from the slave.

The control code is written to this register accompanied by the transfer between the host The status code is returned to this register to indicate that the slave has recognized the co code, and a specified transfer cycle has been completed.

18.3.22 SMIC Data Register (SMICDTR)

SMICDTR is one of the registers used to implement SMIC mode. This is an 8-bit registe accessible (readable/writable) from both the slave processor (this LSI) and host processo used for data transfer between the host and slave.

				This is a status flag that indicates that the h finished transmitting the transfer data to SM When the IBFIE3 bit and HDTWIE bit are se the IBFI3 interrupt is requested to the slave. 0: Transfer data transmission wait state [Clearing condition] After the slave reads HDTWI = 1, writes 0 to 1: Transfer data transmission end [Setting condition] The transfer cycle is write transfer and the h 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 h finished receiving the transfer data from SM When the IBFIE3 bit and HDTRIE bit are se the IBFI3 interrupt is requested to the slave. 0: Transfer data receive wait state [Clearing condition] After the slave reads HDTRI = 1, writes 0 to 1: Transfer data receive end [Setting condition] The transfer cycle is read transfer and the h reads the transfer data from SMICDTR.

				1: Status code receive end [Setting condition] When the host reads the status code of SM
1	CTLWI	0	R/(W)* —	Control Code Transmission End Interrupt This is a status flag that indicates that the I finished transmitting the control code to SM When the IBFIE3 bit and CTLWIE bit are s IBFI3 interrupt is requested to the slave. 0: Control code transmission wait state [Clearing condition] After the slave reads CTLWI = 1, writes 0 t 1: Control code transmission end [Setting condition] 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 I transferring. When the IBFIE3 bit and BUS are set to 1, the IBFI3 interrupt is requeste slave. 0: Transfer start wait state [Clearing condition] After the slave reads BUSYI = 1, writes 0 t 1: Transfer start [Setting condition] When the rising edge of the BUSY bit in SI detected.

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

4	HDTWIE	0	R/W	—	Transfer Data Transmission End Interrupt E Enables or disables HDTWI interrupt that is interrupt source to the slave. 0: Disables transfer data transmission end i 1: Enables transfer data transmission end in
3	HDTRIE	0	R/W	—	Transfer Data Receive End Interrupt Enable Enables or disables HDTRI interrupt that is interrupt source to the slave. 0: Disables transfer data receive end interru 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 interrup 1: Enables status code receive end interrup
1	CTLWIE	0	R/W	—	Control Code Transmission End Interrupt En Enables or disables CTLWI interrupt that is interrupt source to the slave. 0: Disables control code transmission end in 1: Enables control code transmission end in
0	BUSYIE	0	R/W	—	Transfer Start Interrupt Enable Enables or disables BUSYI interrupt that is interrupt source to the slave. 0: Disables transfer start interrupt 1: Enables transfer start interrupt

				This status flag indicates that host writes th BTDTR buffer with FIFO full state at the ho transfer. When the IBFIE3 bit and FRDIE b to 1, IBFI3 interrupt is requested to the slav slave must clear the flag after creating an u area by reading the data in FIFO. 0: FIFO read is not requested [Clearing condition] After the slave reads FRDI = 1, writes 0 to 1: FIFO read is requested [Setting condition] After the host processor transfers data, the writes the data with FIFO Full state.
3	HRDI	0	R/(W)*	BT Host Read Interrupt This status flag indicates that the host read from BTDTR buffer. When the IBFIE3 bit a bit are set to 1, IBFI3 interrupt is requested slave. 0: Host BTDTR read wait state [Clearing condition] After the slave reads HRDI = 1, writes 0 to 1: The host reads from BTDTR [Setting condition] The host reads one byte from BTDTR.

					1: The host writes to BTDTR [Setting condition] The host writes one byte to BTDTR.
1	HBTWI	0	R/(W)*	—	BTDTR Host Write Start Interrupt This status flag indicates that the host writes byte of valid data to BTDTR buffer. When th bit and HBTWIE bit are set to 1, IBFI3 interr requested to the slave. 0: BTDTR host write start wait state [Clearing condition] After the slave reads HBTWI = 1 and writes bit. 1: BTDTR host write start [Setting condition] The host starts writing valid data to BTDTR.

		bit. 1: BTDTR host read end [Setting condition] When the host finished reading the valid da BTDTR.

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

					This status flag indicates that the BMC_HW in BTIMSR is set to 1 by the host. When the 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.

					1: [Setting condition] When the slave detects the falling edge 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] When the slave detects the falling edge B2H_ATN.
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.

					1: [Setting condition] When the slave detects the rising edge o CLR_RD_PTR.
0	CRWPI	0	R/(W)*	—	Write Pointer Clear Interrupt This status flag indicates that the CLR_WR in BTCR is set to 1 by the host. When the IE and CRWPIE bit are set to 1, the IBFI3 inter requested to the slave. 0: [Clearing condition] After the slave reads CRWPI = 1, writes bit. 1: [Setting condition] When the slave detects the rising edge o CLR_WR_PTR.

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

6 5	FSEL1 FSEL0	0 0	R/W R/W	— —	These bits select either FIFO during BT tran FSEL1 FSEL0 0 X :FIFO disabled 1 X :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.

0: BTDTR host read end interrupt is disabled.

1: BTDTR host read end interrupt is enabled.

Note: X Don't care.

18.3.28 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.

					0: BEVT_ATN clear interrupt is disabled. 1: BEVT_ATN clear interrupt is enabled.
3	B2HIE	0	R/W	—	Read End Interrupt Enable Enables or disables the B2HI interrupt whi IBFI3 interrupt source to the slave. 0: Read end interrupt is disabled. 1: Read end interrupt is enabled.
2	H2BIE	0	R/W	—	Write End Interrupt Enable Enables or disables the H2BI interrupt 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 v IBFI3 interrupt source to the slave. 0: Write pointer clear interrupt is disabled. 1: Write pointer clear interrupt is enabled.

					0: Indicates waiting for BT write transfer 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

					1: ATN event interrupt request is availabl [Setting condition] When the slave writes a 1 after a 0 has 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 finished writing all data to the BTDTR bu Setting the B2H_IRQ_EN bit in the BTIM register enables the B2H_ATN bit to be an interrupt source to the host. 0: Host has completed reading the BTD [Clearing condition] When the host writes a 1 1: Slave has completed writing to the BT buffer [Setting condition] When the slave writes a 1 after a 0 has 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 h finished writing all data to the BTDTR bu 0: Slave has completed reading the BTD [Clearing condition] When the slave writes a 0 after a 1 has from H2B_ATN. 1: Host has completed writing to the BT buffer [Setting condition] When the host writes a 1

				1: Read pointer clear [Setting condition] When the host writes a 1.
0	CLR_WR_PTR	0	R/(W)*2 (W)*1	Write Pointer Clear This bit is used by the host to clear the 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.

					the slave is stored in FIFO (64 bytes) for transfer and read out by the host in order writing.

18.3.31 BT Interrupt Mask Register (BTIMSR)

BTIMSR is one of the registers used to implement BT mode. The BTIMSR register con bits used to control the interrupts to the host.

			R/W		
Bit	**Bit Name**	**Initial Value**	**Slave**	**Host**	**Description**
7	BMC_ HWRST	0	R/(W)*[2]	R/(W)*[1]	Slave Reset Performs a reset from the host to the s host can only write a 1. Writing a 0 to t invalid. The host will always return a 0 out. Setting the RSTRENBL bit enable read from the host. 0: The reset is cancelled [Clearing condition] When the slave writes a 0, after a 1 ha read from BMC_HWRST. 1: The reset is in progress. [Setting condition] When the host writes a 1.
6	—	0	R/W	R/W	Reserved
5	—	0	R/W	R/W	

					read from OEM, or when the host wr
1	B2H_IRQ	0	R/(W)*[1]	R/(W)*[3]	BMC to HOST interrupt Informs the host that an interrupt has b requested when the BEVT_ATN or B2H bit has been set. The SERIRQ is not is generate the SERIRQ, it should be issu the program. 0: B2H_IRQ interrupt is not requested [Clearing condition] When the host writes a 1. 1: B2H_IRQ interrupt is requested [Setting condition] When the slave writes a 1, after a 0 ha read from B2H_IRQ
0	B2H_IRQ_EN	0	R	R/W	BMC to HOST interrupt enable Enables or disables the B2H_IRQ inter which is an interrupt source from the sl the 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.

					number of bytes that have been written to. when data is read from the slave, the value decremented by only the number of bytes been read.

18.3.33 BT FIFO Valid Size Register 1 (BTFVSR1)

BTFVSR1 is one of the registers used to implement BT mode. BTFVSR1 indicates a va
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 byt FIFO (the number of bytes which the host for host read transfer. When data is written slave, the value in BTFVSR1 is incremente number of bytes that have been written to. when data is read from the host, the value decremented by only the number of bytes been read.

Use the following procedure to activate the LPC interface after a reset release.

1. Read the signal line status and confirm that the LPC module can be connected. Also c the LPC module is initialized internally.
2. When using channels 1 and 2, set LADR1 and LADR2 to determine the I/O address.
3. When using channel 3, set LADR3 to determine the I/O address and whether bidirecti registers are to be used.
4. When using the SCIF module, set SCIFAR to determine the I/O address.
5. Set the enable bit (LPC3E to LPC1E) for the channel to be used. Also set SCIFE if th to be used.
6. Set the enable bits (FGA20E, PMEE, LSMIE, and LSCIE) for the additional function used.
7. Set the selection bits for other functions (SDWNE, IEDIR).
8. As a precaution, clear the interrupt flags (LRST, SDWN, ABRT, OBF, and OBEI). R or TWR15 to clear IBF.
9. Set receive complete interrupt enable bits (IBFIE3 to IBFIE1, and ERRIE) as necessa

18.4.2 LPC I/O Cycles

There are 12 types of LPC transfer cycle: LPC memory read, LPC memory write, I/O rea write, DMA read, DMA write, bus master memory read, bus master memory write, bus m read, bus master I/O write, FW memory read, and FW memory write. Of these, the LPC c LSI supports I/O read and I/O write.

changes are made at this timing, so in the event of a transfer cycle forced termination (at
registers and flags are not changed.

The timing of the $\overline{\text{LFRAME}}$, LCLK, and LAD signals is shown in figures 18.2 and 19.3

Table 18.5 LPC I/O Cycle

	I/O Read Cycle			I/O Write Cycle		
State Count	**Contents**	**Drive Source**	**Value (3 to 0)**	**Contents**	**Drive Source**	**V (**
1	Start	Host	0000	Start	Host	0
2	Cycle type/direction	Host	0000	Cycle type/direction	Host	0
3	Address 1	Host	Bits 15 to 12	Address 1	Host	B
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	1111	Data 1	Host	B
8	Turnaround	None	ZZZZ	Data 2	Host	B
9	Synchronization	Slave	0000	Turnaround (recovery)	Host	1
10	Data 1	Slave	Bits 3 to 0	Turnaround	None	Z
11	Data 2	Slave	Bits 7 to 4	Synchronization	Slave	0
12	Turnaround (recovery)	Slave	1111	Turnaround (recovery)	Slave	1
13	Turnaround	None	ZZZZ	Turnaround	None	Z

Figure 18.2 Typical LFRAME Timing

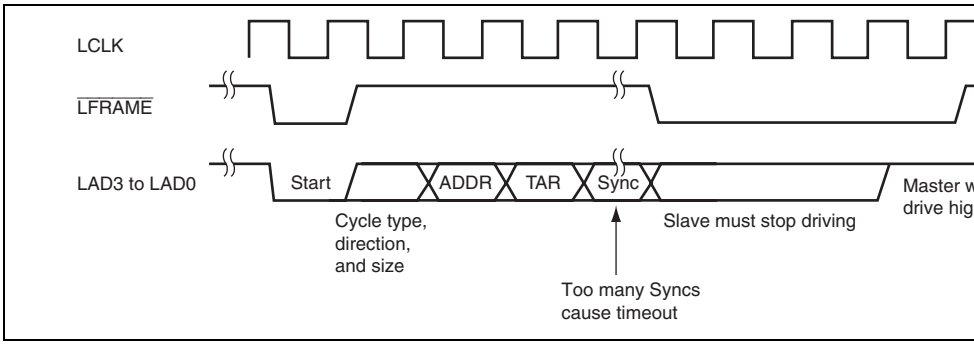


Figure 18.3 Abort Mechanism

18.4.3 SMIC Mode Transfer Flow

Figure 18.4 shows the write transfer flow and figure 18.5 shows the read transfer flow in mode.

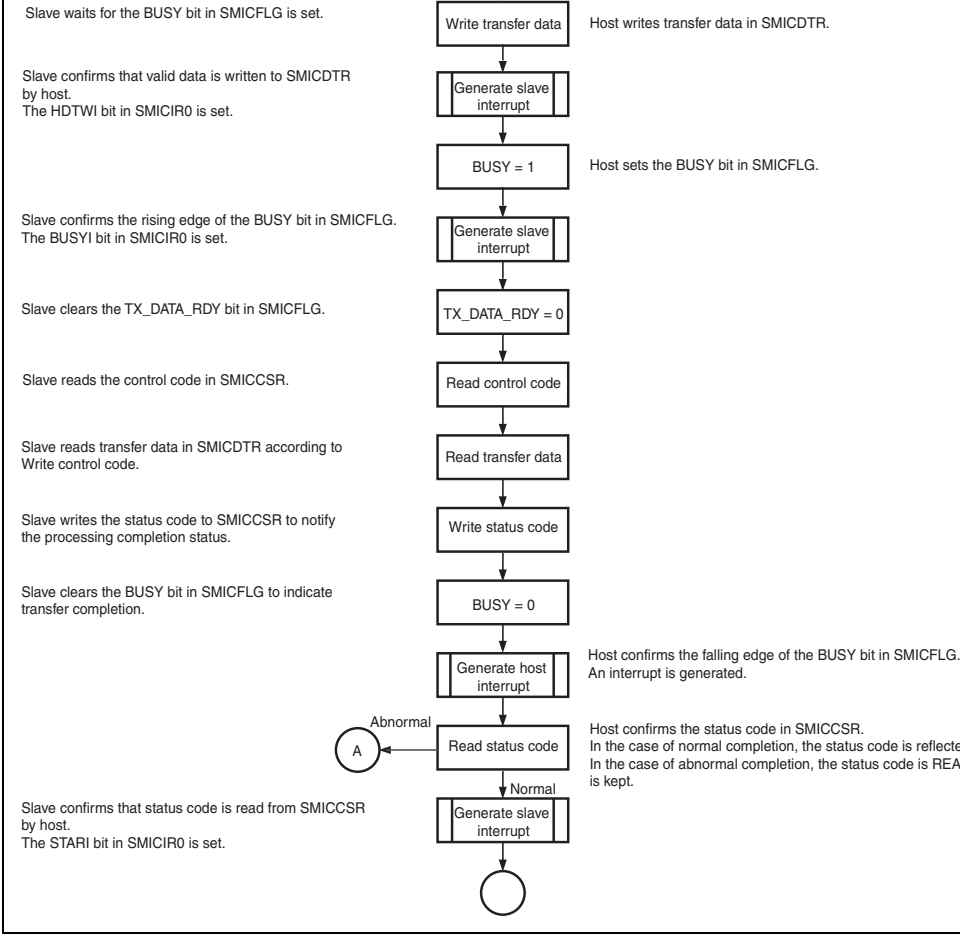


Figure 18.4 SMIC Write Transfer Flow

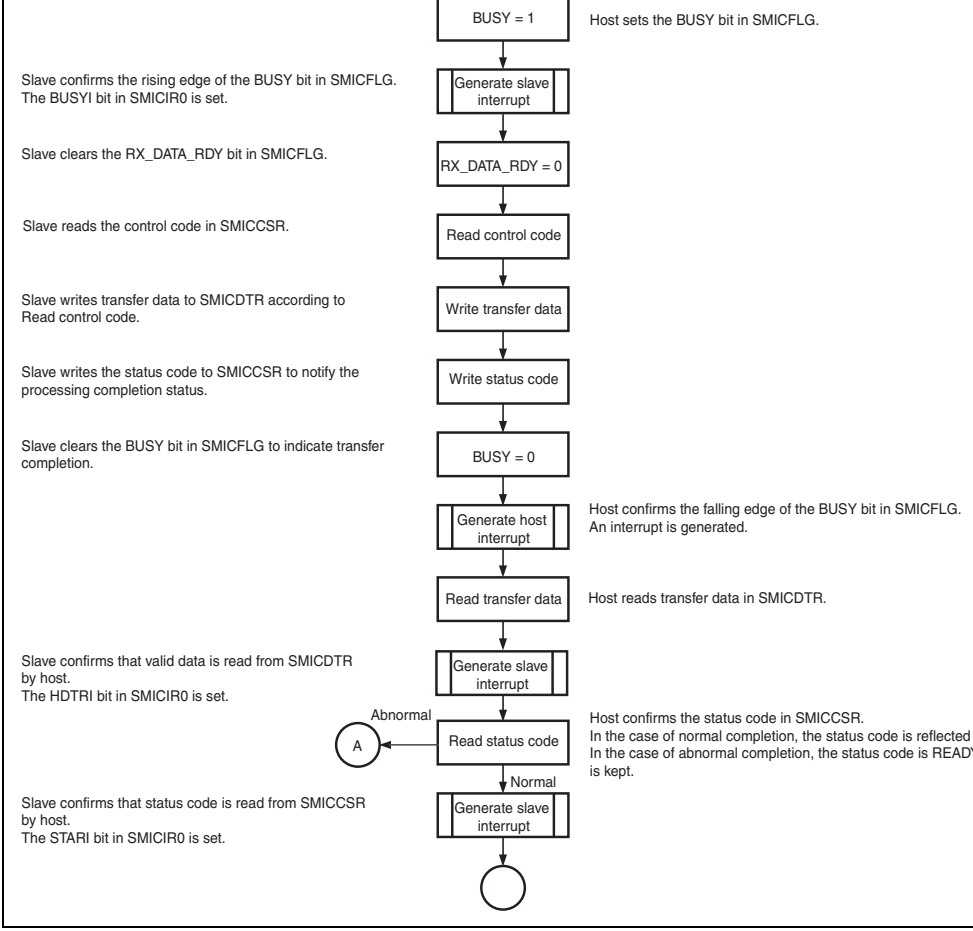


Figure 18.5 SMIC Read Transfer Flow

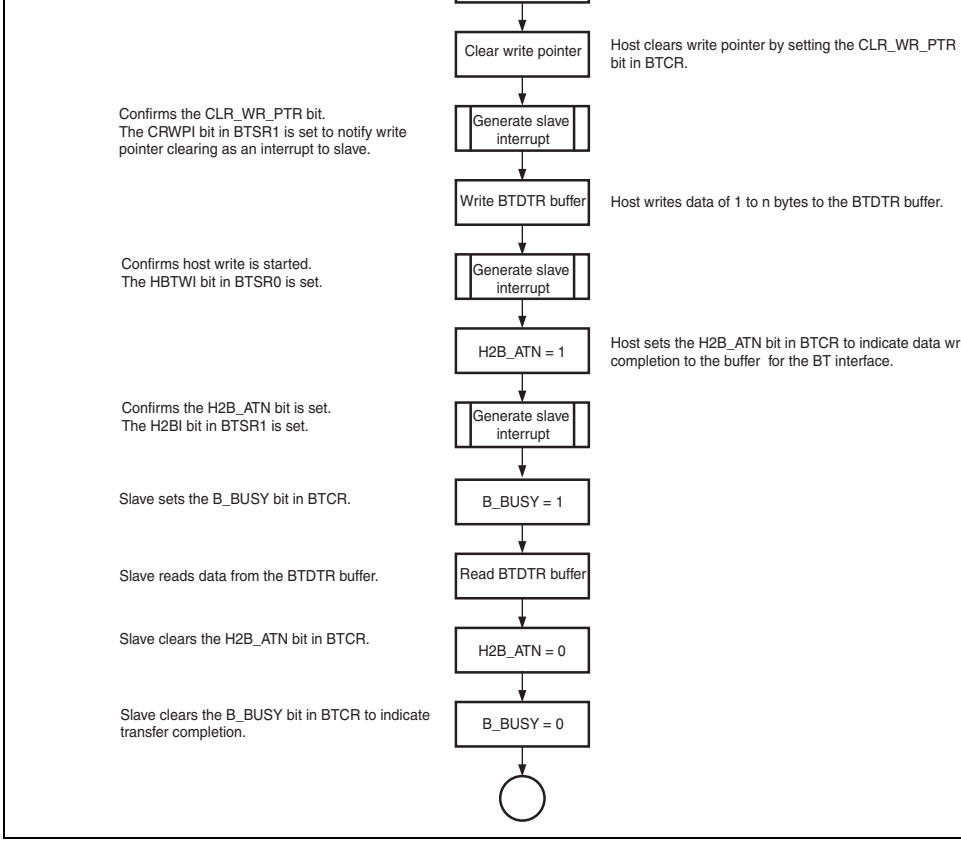


Figure 18.6 BT Write Transfer Flow

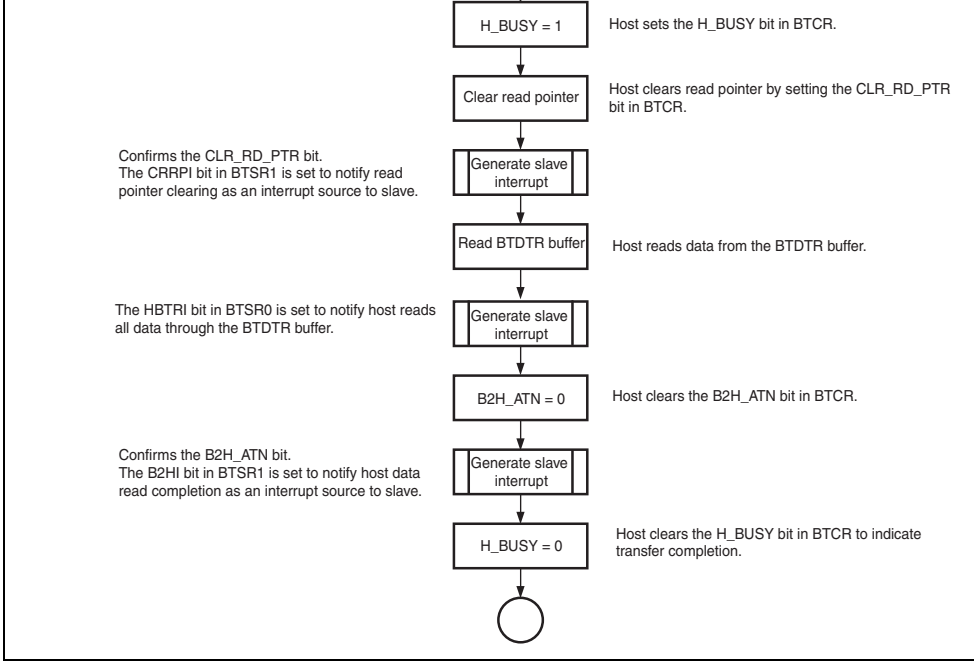


Figure 18.7 BT Read Transfer Flow

Output of the Gate A20 signal can be controlled by an H'D1 command and data. When t
(this LSI) receives data, it normally reads IDR1 in the interrupt handling routine activate
IBFI1 interrupt. At this time, firmware copies bit 1 of data following an H'D1 command
outputs it on pin GA20.

(2) Fast Gate A20 Operation

The internal state of pin GA20 is initialized to 1 since the initial value of the FGA20E b
When the FGA20E bit is set to 1, pin P81/GA20 functions as the output of the fast GA2
The state of pin GA20 can be monitored by reading bit GA20 in HICR2.

The initial output from this pin is 1, which is the initial value. Afterward, the host can m
the output from this pin by sending commands and data. This function is only available
IDR1. The LPC decodes commands input from the host. When an H'D1 host command
detected, bit 1 of the data following the host command is output from pin GA20. This op
does not depend on firmware or interrupts, and is faster than the regular processing usin
interrupts. Table 18.6 shows the conditions that set and clear pin GA20. Figure 18.8 sho
GA20 output flow. Table 18.7 indicates the GA20 output signal values.

Table 18.6 GA20 Setting/Clearing Timing

Pin Name	Setting Condition	Clearing Condition
GA20	When bit 1 of the data that follows an H'D1 host command is 1	When bit 1 of the data that fol H'D1 host command is 0

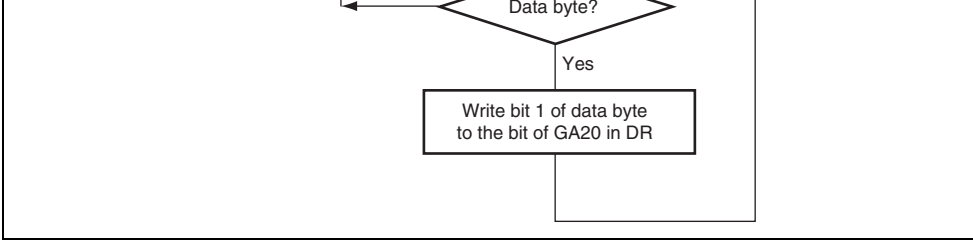


Figure 18.8 GA20 Output

1	H'FF command	0	Q (0)	
1	H'D1 command	0	Q	Turn-on seque
0	1 data*[1]	0	1	(abbreviated fo
1/0	Command other than H'FF and H'D1	1	Q (1)	
1	H'D1 command	0	Q	Turn-off seque
0	0 data*[2]	0	0	(abbreviated fo
1/0	Command other than H'FF and H'D1	1	Q (0)	
1	H'D1 command	0	Q	Cancelled seq
1	Command other than H'D1	1	Q	
1	H'D1 command	0	Q	Retriggered se
1	H'D1 command	0	Q	
1	H'D1 command	0	Q	Consecutively
0	Any data	0	1/0	sequences
1	H'D1 command	0	Q (1/0)	

Notes: 1. Any data with bit 1 set to 1.
2. Any data with bit 1 cleared to 0.

for exiting software standby mode before clearing the shutdown state with the $\overline{\text{LPCPD}}$ sig

If the SDWNE bit has been set to 1 beforehand, the LPC hardware shutdown state is ente same time as the $\overline{\text{LPCPD}}$ signal falls, and prior preparation is not possible. If the LPC sof shutdown state is set by means of the SDWNB bit, on the other hand, the LPC software s state cannot be cleared at the same time as the rising edge of the $\overline{\text{LPCPD}}$ signal. Taking th points into consideration, the following operating procedure uses a combination of LPC s shutdown and LPC hardware shutdown.

1. Clear the SDWNE bit to 0.
2. Set the ERRIE bit to 1 and wait for an interrupt by the SDWN flag.
3. When an ERRI interrupt is generated by the SDWN flag, check the LPC interface inte status flags and perform any necessary processing.
4. Set the SDWNB bit to 1 to set LPC software standby mode.
5. Set the SDWNE bit to 1 and make a transition to LPC hardware standby mode. The S 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 risen steps 3 to 5. If the signal has risen, clear SDWNE to 0 to return to the state in step 1.
7. If software standby mode has been set, exit software standby mode by some means independent of the LPC.
8. When a rising edge is detected in the $\overline{\text{LPCPD}}$ signal, the SDWNE bit is automatically to 0. If the slave has been placed in sleep mode, the mode is exited by means of $\overline{\text{LRES}}$ signal input, on completion of the LPC transfer cycle, or by some other means.

LSCI	PD0	Δ	I/O	Hi-Z, only when LSCIE = 1
$\overline{\text{LSMI}}$	PD1	Δ	I/O	Hi-Z, only when LSMIE = 1
$\overline{\text{PME}}$	PD2	Δ	I/O	Hi-Z, only when PMEE = 1
GA20	PD3	Δ	I/O	Hi-Z, only when FGA20E = 1
$\overline{\text{CLKRUN}}$	PD4	O	Input	Hi-Z
$\overline{\text{LPCPD}}$	PD5	X	Input	Needed to clear shutdown stat

[Legend]

O: Pin that is shutdown by the shutdown function

Δ: Pin that is shutdown only when the LPC function is selected by register setting

X: Pin that is not shutdown

In the LPC shutdown state, the LPC's internal state and some register bits are initialized. of priority of LPC shutdown and reset states is as follows.

1. System reset (reset by $\overline{\text{RES}}$ pin input, or WDT0 overflow)
   All register bits, including bits LPC4E to LPC1E, are initialized.
2. LPC hardware reset (reset by $\overline{\text{LRESET}}$ pin input)
   LRSTB, SDWNE, and SDWNB bits are cleared to 0.
3. LPC software reset (reset by LRSTB)
   SDWNE and SDWNB bits are cleared to 0.
4. LPC hardware shutdown
   SDWNB bit is cleared to 0.
5. LPC software shutdown

The scope of the initialization in each mode is shown in table 18.9.

BTCSR1), SMIE2 (internal state)			
Host interrupt enable bits (IRQ1E1, IRQ12E1, SMIE2, IRQ6E2, IRQ9E2 to IRQ11E2, SMIE3B, SMIE3A, IRQ6E3, IRQ9E3 to IRQ11E3, SELREQ, IEDIR2 to IEDIR3), Q/$\overline{C}$ flag	Initialized	Initialized	Re
LRST flag	Initialized (0)	Can be set/cleared	Ca se
SDWN flag	Initialized (0)	Initialized (0)	Ca se
LRSTB bit	Initialized (0)	HR: 0 SR: 1	0 (se
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, LADR1 to LADR3, IBFIE1 to IBFIE3, PMEE, PMEB, LSMIE, LSMIB, LSCIE, LSCIB, TWRE, SELSTR3, SELIRQ1, SELSMI, SELIRQ3 to SELIRQ15, HICR4, HICR5, SCIFAR, HISEL, BTCSR0, BTCSR1)	Initialized	Retained	Re
$\overline{\text{LRESET}}$ signal	Input (port function	Input	In
$\overline{\text{LPCPD}}$ signal		Input	In
LAD3 to LAD0, $\overline{\text{LFRAME}}$, LCLK, SERIRQ, $\overline{\text{CLKRUN}}$ signals		Input	Hi
$\overline{\text{PME}}$, $\overline{\text{LSMI}}$, LSCI, GA20 signals (when function is selected)		Output	Hi
$\overline{\text{PME}}$, $\overline{\text{LSMI}}$, LSCI, GA20 signals (when function is not selected)		Port function	Po

Note: System reset: Reset by STBY input, RES input, or WDT overflow
LPC reset: Reset by LPC hardware reset (HR) or LPC software reset (SR)
LPC shutdown: Reset by LPC hardware shutdown (HS) or LPC software shutdown

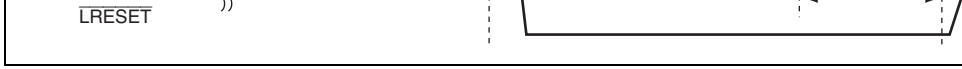


Figure 18.9 Power-Down State Termination Timing

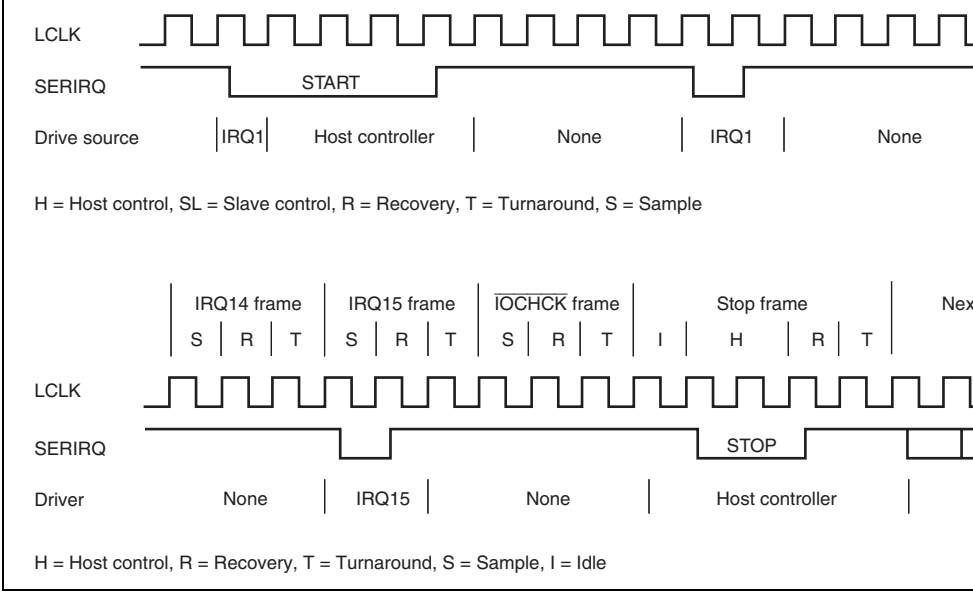


Figure 18.10 SERIRQ Timing

		Host		first state, then next 3 states 0-drive
1	IRQ0	Slave	3	Drive impossible
2	IRQ1	Slave	3	Drive possible in LPC channel 1 and
3	SMI	Slave	3	Drive possible in LPC channels 2, 3,
4	IRQ3	Slave	3	Drive possible in SCIF or by IRQ3E
5	IRQ4	Slave	3	Drive possible in SCIF or by IRQ4E
6	IRQ5	Slave	3	Drive possible in SCIF or by IRQ5E
7	IRQ6	Slave	3	Drive possible in LPC channels 2, 3,
8	IRQ7	Slave	3	Drive possible in SCIF or by IRQ7E
9	IRQ8	Slave	3	Drive possible in SCIF or by IRQ8E
10	IRQ9	Slave	3	Drive possible in LPC channels 2, 3,
11	IRQ10	Slave	3	Drive possible in LPC channels 2, 3,
12	IRQ11	Slave	3	Drive possible in LPC channels 2, 3,
13	IRQ12	Slave	3	Drive possible in LPC channel 1 and
14	IRQ13	Slave	3	Drive possible in SCIF or by IRQ13E
15	IRQ14	Slave	3	Drive possible in SCIF or by IRQ14E
16	IRQ15	Slave	3	Drive possible in SCIF or by IRQ15E
17	IOCHCK	Slave	3	Drive impossible
18	Stop	Host	Undefined	First, 1 or more idle states, then 2 or 0-driven by host 2 states: Quiet mode next 3 states: Continuous mode next

18.4.8 LPC Interface Clock Start Request

A request to restart the clock (LCLK) can be sent to the host by means of the $\overline{\text{CLKRUN}}$ p
LPC data transfer and SERIRQ in continuous mode, a clock restart is never requested sin
transfer cycles are initiated by the host. With SERIRQ in quiet mode, when a host interru
request is generated the $\overline{\text{CLKRUN}}$ signal is driven and a clock (LCLK) restart request is s
the host. The timing for this operation is shown in figure 18.11.

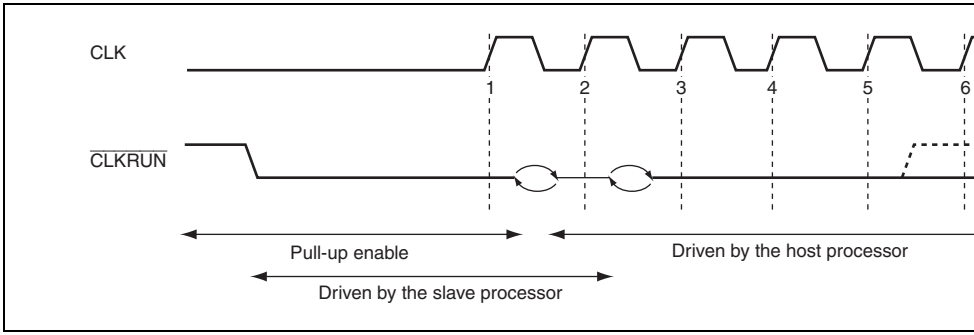


Figure 18.11 Clock Start Request Timing

Cases other than SERIRQ in quiet mode when clock restart is required must be handled v
different protocol, using the $\overline{\text{PME}}$ signal, etc.

18.4.9 SCIF Control from LPC Interface

Setting the SCIFE bit in HICR5 to 1 allows the LPC host to communicate with the SCIF.
the LPC interface can access the registers of the module SCIF other than SCIFCR. For de
transmission and reception, see section 15, Serial Communication Interface with FIFO (S

Interrupt	Description
IBFI1	When IBFIE1 is set to 1 and IDR1 reception is completed
IBFI2	When IBFIE2 is set to 1 and IDR2 reception is completed
IBFI3	When IBFIE3 is set to 1 and IDR3 reception is completed, or when TW IBFIE3 are set to 1 and reception is completed up to TWR15
ERRI	When ERRIE is set to 1 and one of LRST, SDWN and ABRT is set to 1

When the IEDIR bit in SIRQCR0is cleared to 0, host interrupt sources and LPC channels linked to the host interrupt request enable bits. When the OBF flag is cleared to 0 by a rea ODR or TWR15 by the host in the corresponding LPC channel, the corresponding host in enable bit is automatically cleared to 0, and the host interrupt request is cleared.

When the IEDIR bit is set to 1 in SIRQCR, a host interrupt is only requested by the host i enable bits. The host interrupt enable bit is not cleared when OBF is cleared. Therefore, S SMIE3A, SMIE3B, SMIE4 and IRQ6En, IRQ9En, IRQ10En, IRQ11En lose their respect functional differences (n = 2, 3). In order to clear a host interrupt request, it is necessary t the host interrupt enable bit. As for HIRQ3 to HIRQ5, HIRQ7, HIRQ8, and HIRQ13 to H setting the enable bit in SIRQCR4 to 1 requests the corresponding host interrupt, and clea enable bit to 0 clears the corresponding host interrupt request.

When the SCIF channels are used, a host interrupt request is cleared when the relevant SC interrupt is cleared.

Table 18.12 summarizes the methods of setting and clearing these bits when the LPC cha used, and table 18.13 summarizes the methods of setting and clearing these bits when the channels are used. Figure 18.12 shows the processing flowchart.

	SMIE3A and writes 1 • writes to TWR15, then reads 0 from bit SMIE3B and writes 1	reads ODR3 • writes 0 to bit SMIE3B reads TWR15
SMI (IEDIR2 = 1 or IEDIR3 = 1)	Internal CPU • reads 0 from bit SMIE2, then writes 1 • reads 0 from bit SMIE3A, then writes 1 • reads 0 from bit SMIE3B, then writes 1	Internal CPU • writes 0 to bit SMIE2 • writes 0 to bit SMIE3A • writes 0 to bit SMIE3B
HIRQi (i = 6, 9, 10, 11) (IEDIR2 = 0 or IEDIR3 = 0)	Internal CPU • writes to ODR2, then reads 0 from bit IRQiE2 and writes 1 • writes to ODR3, then reads 0 from bit IRQiE3 and writes 1	Internal CPU • writes 0 to bit IRQiE2, reads ODR2 • writes 0 to bit IRQiE3, reads ODR3
HIRQi (i = 6, 9, 10, 11) (IEDIR2 = 1 or IEDIR3 = 1)	Internal CPU • reads 0 from bit IRQiE2, then writes 1 • reads 0 from bit IRQiE3, then writes 1	Internal CPU • writes 0 to bit IRQiE2 • writes 0 to bit IRQiE3

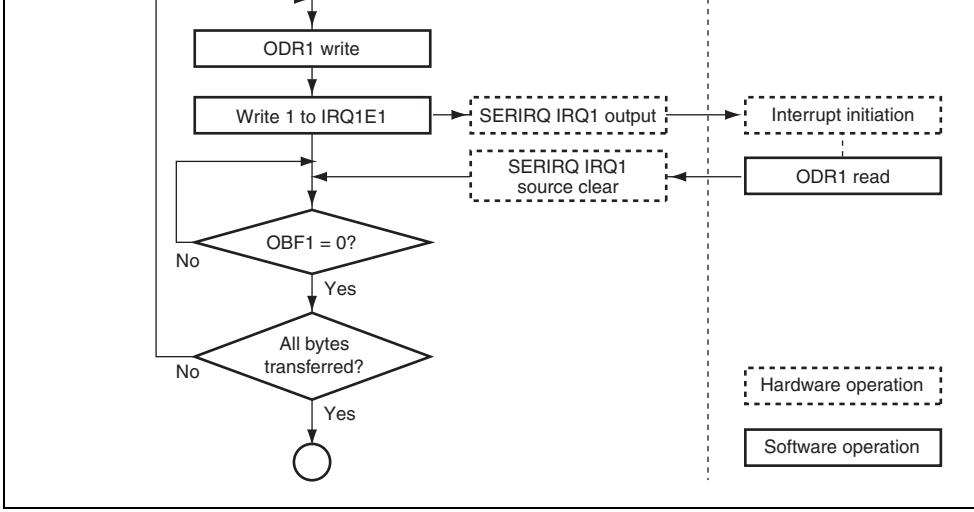


Figure 18.12 HIRQ Flowchart (Example of Channel 1)

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 write authority for TWI
TWR15 has been obtained.

Table 18.14 shows host address examples for LADR3 and registers, IDR3, ODR3, STR
TWR0MW, TWR0SW, and TWR1 to TWR15.

TWR3	H'A253	H'3FC3
TWR4	H'A254	H'3FC4
TWR5	H'A255	H'3FC5
TWR6	H'A256	H'3FC6
TWR7	H'A257	H'3FC7
TWR8	H'A258	H'3FC8
TWR9	H'A259	H'3FC9
TWR10	H'A25A	H'3FCA
TWR11	H'A25B	H'3FCB
TWR12	H'A25C	H'3FCC
TWR13	H'A25D	H'3FCD
TWR14	H'A25E	H'3FCE
TWR15	H'A25F	H'3FCF

- Eight input channels
- Conversion time: 4.7 μs per channel (at 34-MHz operation)
- Two operating modes

  Single mode: Single-channel A/D conversion

  Scan mode: Continuous A/D conversion on 1 to 4 channels or continuous A/D conv
  1 to 8 channels
- Eight data registers

  Conversion results are held in a 16-bit data register for each channel
- Sample and hold function
- Three ways of starting A/D conversion
  - Software
  - Trigger from TMR_0
  - External trigger signal
- Interrupt request

  A/D conversion end interrupt (ADI) request can be generated
- Module stop mode can be set

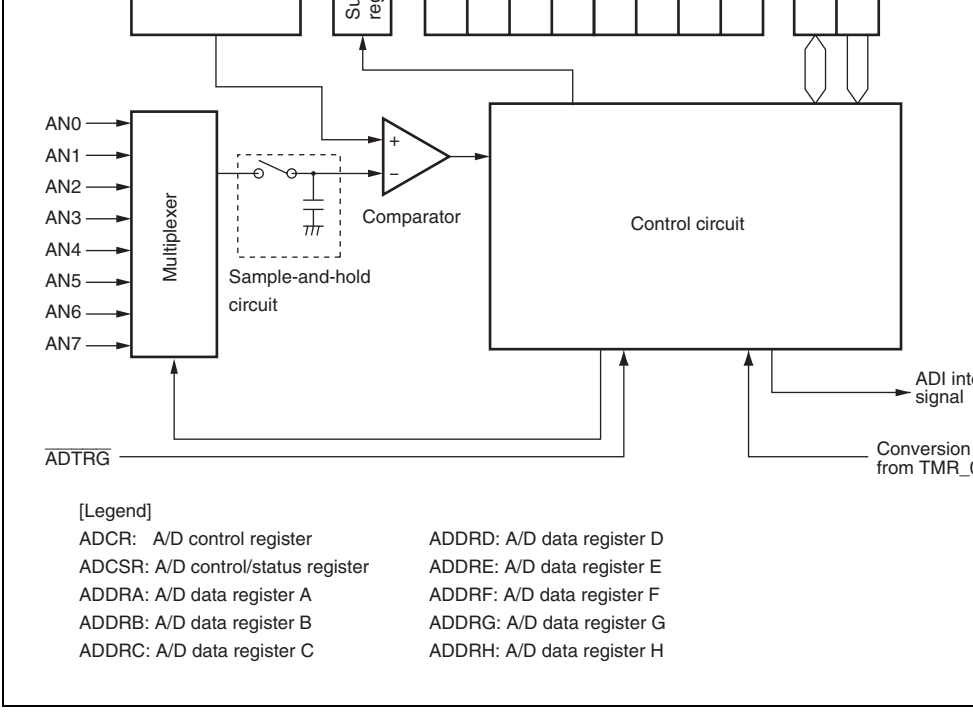


Figure 19.1 Block Diagram of the A/D Converter

Analog input pin 3	AN3	Input	
Analog input pin 4	AN4	Input	
Analog input pin 5	AN5	Input	
Analog input pin 6	AN6	Input	
Analog input pin 7	AN7	Input	
A/D external trigger input pin	$\overline{\text{ADTRG}}$	Input	External trigger input for starting A/D co
Analog power supply pin	AVcc	Input	Analog block power supply
Analog ground pin	AVss	Input	Analog block ground
Reference power supply pin	AVref	Input	Reference voltage for A/D converter

- A/D data register G (ADDRG)
- A/D data register H (ADDRH)
- A/D control/status register (ADCSR)
- A/D control register (ADCR)

19.3.1 A/D Data Registers A to H (ADDRA to ADDRH)

The ADDR are eight 16-bit read-only registers, ADDRA to ADDRH, which store the res
A/D conversion. The ADDR registers, which store a conversion result for each channel, a
in table 19.2.

The converted 10-bit data is stored to bits 15 to 6. The lower 6-bit data is always read as

The data bus between the CPU and the A/D converter is 16-bit width and can be read dire
from the CPU. The ADDR must always be accessed in 16-bit unit. They cannot be access
bit unit.

The results of A/D conversion are stored in each registers, when the ADF flag is set to 1.

AN6	ADDRG
AN7	ADDRH

19.3.2 A/D Control/Status Register (ADCSR)

The ADCSR controls the operation of the A/D conversion.

Bit	Bit Name	Initial Value	R/W	Description
7	ADF	0	R/(W)*	A/D End Flag A status flag that indicates the end of A/D conve This flag indicates that the results of A/D conver stored in the A/D data registers. [Setting conditions] • When A/D conversion ends in single mode • When A/D conversion ends on all channels s in scan mode [Clearing conditions] • When 0 is written after reading ADF = 1 • When DTC starts by an ADI interrupt and AD read
6	ADIE	0	R/W	A/D Interrupt Enable Enables ADI interrupt by ADF when this bit is se

				This is a read-only bit and cannot be modified.
3	—	0	R/W	Reserved The initial value should not be changed.
2 1 0	CH2 CH1 CH0	All 0	R/W	Channel Select 2 to 0 Select analog input channels together with the SCA and the SCANS bit of ADCR.

When SCANE = 0, and SCANS = X	When SCANE = 1 and SCANS = 0	When SC and SCA
000: AN0	000: AN0	000: AN0
001: AN1	001: AN0 and AN1	001: AN1
010: AN2	010: AN0 to AN2	010: AN0
011: AN3	011: AN0 to AN3	011: AN0
100: AN4	100: AN4	100: AN0
101: AN5	101: AN4 and AN5	101: AN0
110: AN6	110: AN4 to AN6	110: AN0
111: AN7	111: AN4 to AN7	111: AN0

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

[Legend] X: Don't care

				00: Disables starting by trigger signals. 10 0: Enables starting by a trigger from TMR_0. 10 1: Enables starting by the $\overline{\text{ADTRG}}$ pin input. Other than above: Setting prohibited
5	SCANE	0	R/W	Scan Mode
4	SCANS	0	R/W	Select the operation mode of A/D conversion 0X: Single mode 10: Scan mode (consecutive A/D conversion of channels 1 t 11: Scan mode (consecutive A/D conversion of channels 1 t
3	CKS1	0	R/W	Clock Select 1 and 0
2	CKS0	0	R/W	Set the A/D conversion time. Setting should be while the conversion is stopped (ADST = 0). 00: Setting prohibited 01: Conversion time = 80 states (max) (20 MHz 10: Conversion time = 160 states (max) 11: Conversion time = 320 states (max)
1	ADSTCLR	0	R/W	A/D Start Clear Sets automatic clearing of the ADST bit in scan 0: Disables automatic clearing of ADST in scan 1: ADST is automatically cleared when A/D conv all the selected channels has been completed mode.

[Legend]

X: Don't care

Operations are as follows.

1. A/D conversion on the specified channel is started when the ADST bit in ADCSR is s by software or by the input of trigger signal.
2. When A/D conversion is completed, the result is transferred to the A/D data register corresponding to the channel.
3. On completion of A/D conversion, the ADF bit in ADCSR is set to 1. If the ADIE bit 1 at this time, an ADI interrupt request is generated.
4. The ADST bit remains set to 1 during A/D conversion. When conversion ends, the AD is automatically cleared to 0, and the A/D converter enters the idle state. If the ADST cleared during A/D conversion, the A/D converter stops conversion and enters the idl

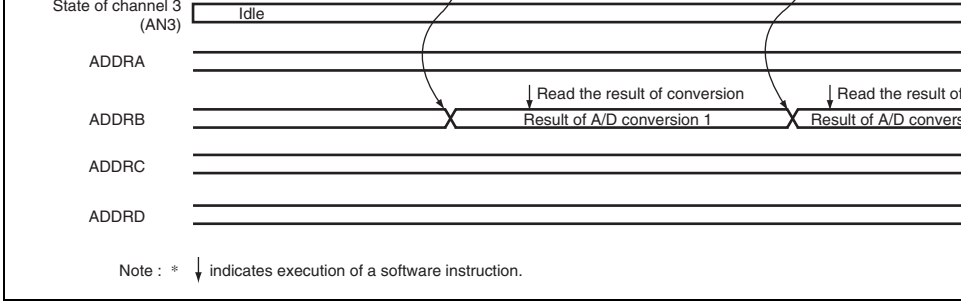
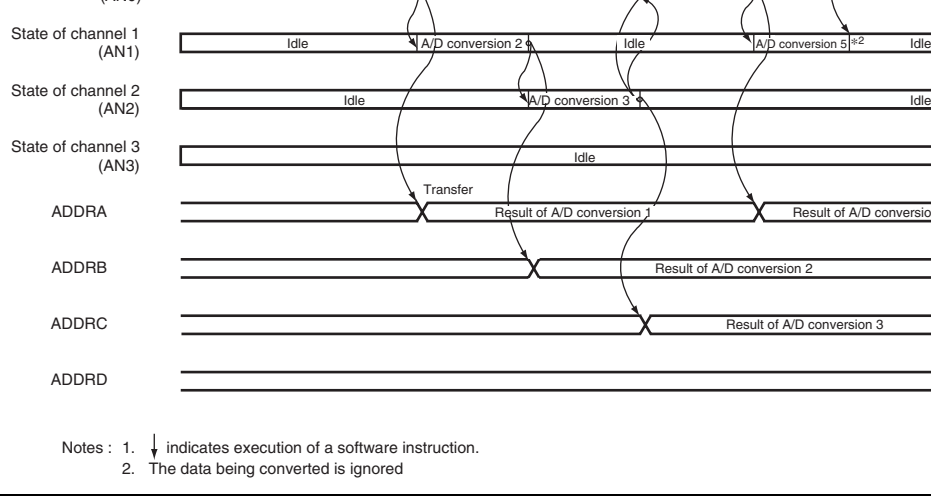


Figure 19.2 Example of A/D Converter Operation (When Channel 1 is Selected in Single Mode)

19.4.2 Scan Mode

In scan mode, A/D conversion is performed sequentially on the specified channels (four or eight channel maximum). Operations are as follows.

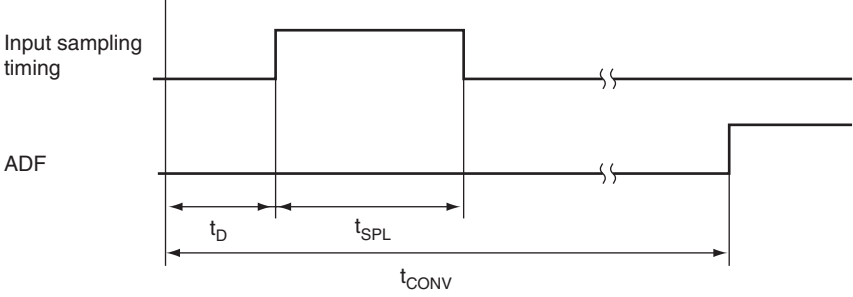
1. When the ADST bit in ADCSR is set to 1 by software or by the input of trigger signa conversion starts from the first channel of the selected channel. Consecutive A/D co either four channels maximum (SCANE and SCANS = B'10) or eight channels maxi (SCANE and SCANS = B'11) can be selected. In the case of consecutive A/D conve four channels, the operation starts from AN0 when CH2 = B'0, and starts from AN4 = B'1. In the case of consecutive A/D conversion on eight channels, the operation sta AN0.
2. When A/D conversion for each channel is completed, the result is sequentially transf the A/D data register corresponding to each channel.
3. When conversion of all the selected channels is completed, the ADF bit in ADCSR i If the ADIE bit is set to 1 at this time, an ADI interrupt is requested after A/D conve Conversion of the first channel in the group starts again.



Notes : 1. ↓ indicates execution of a software instruction.
2. The data being converted is ignored

Figure 19.3 Example of A/D Converter Operation (When Channels AN0 to AN3 are Selected in Scan Mode)

In scan mode, the values given in table 19.3 apply to the first conversion time. The valu
table 19.4 apply to the second and subsequent conversions. In either case, bits CKS1 an
ADCR should be set so that the conversion time is within the ranges indicated by the A/
conversion characteristics.



[Legend]
(1): ADCSR write cycle
(2): ADCSR address
t_D: A/D conversion start delay
t_{SPL}: Input sampling time
t_{CONV}: A/D conversion time

Figure 19.4 A/D Conversion Timing

Note: Values in the table are the number of states.

Table 19.4 A/D Conversion Characteristics (Scan Mode)

CKS1	CKS0	Conversion Time (Number of States)
0	0	Setting prohibited
	1	80 (Fixed)
1	0	160 (Fixed)
	1	320 (Fixed)

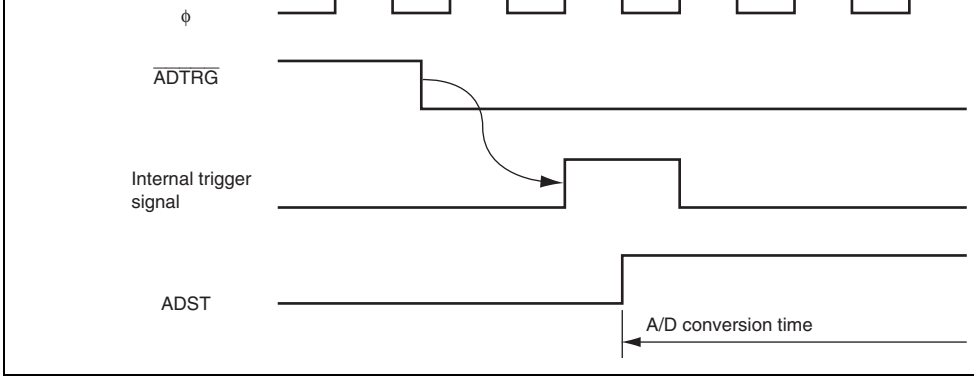


Figure 19.5 Timing of External Trigger Input

ADI	A/D conversion end	ADF	Possible

19.6 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 figure19.6).
- Offset error

  The deviation of the analog input voltage value from the ideal A/D conversion chara
  when the digital output changes from the minimum voltage value B'00 0000 0000 (H
  B'00 0000 0001 (H'001) (see figure 19.7).
- Full-scale error

  The deviation of the analog input voltage value from the ideal A/D conversion chara
  when the digital output changes from B'11 1111 1110 (H'3FE) to B'11 1111 1111 (H
  figure 19.7).
- Nonlinearity error

  The error with respect to the ideal A/D conversion characteristics between the zero v
  the full-scale voltage. Does not include the offset error, full-scale error, or quantizati
  (see figure 19.7).
- Absolute accuracy

  The deviation between the digital value and the analog input value. Includes the offs
  full-scale error, quantization error, and nonlinearity error.

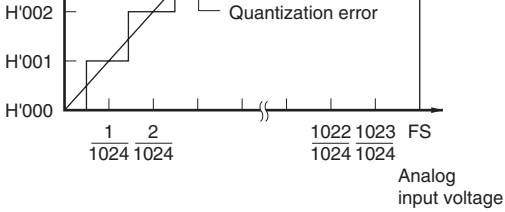


Figure 19.6 A/D Conversion Accuracy Definitions

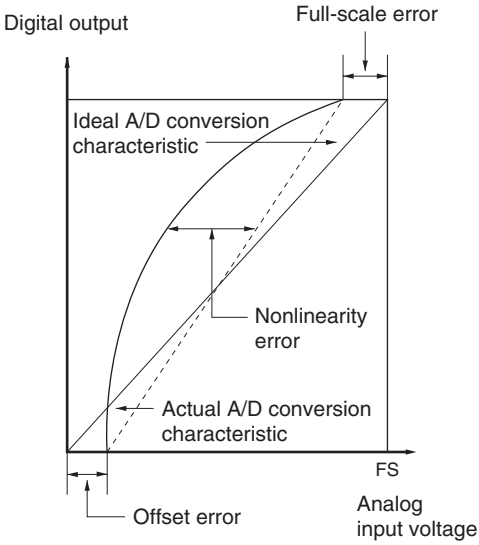


Figure 19.7 A/D Conversion Accuracy Definitions

This LSI's analog input is designed so that the conversion accuracy is guaranteed for an signal for which the signal source impedance is 5 kΩ or less. This specification is provid enable the A/D converter's sample-and-hold circuit input capacitance to be charged with sampling time; if the sensor output impedance exceeds 5 kΩ, charging may be insufficie may not be possible to guarantee the A/D conversion accuracy. However, if a large capa provided externally in single mode, the input load will essentially comprise only the inte resistance of 10 kΩ, and the signal source impedance is ignored. However, since a low-p effect is obtained in this case, it may not be possible to follow an analog signal with a la differential coefficient (voltage fluctuation ratio of 5 mV/μs or greater for example) (see 19.8). When converting a high-speed analog signal or converting in scan mode, a low-in buffer should be inserted.

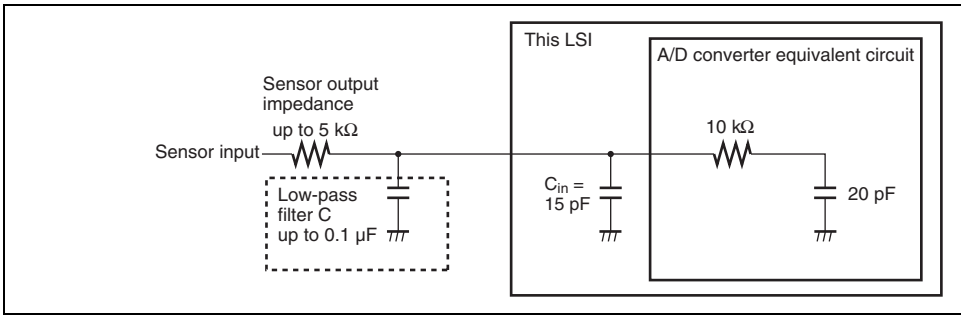


Figure 19.8 Example of Analog Input Circuit

If conditions shown below are not met, the reliability of this LSI may be adversely affecte

- Analog input voltage range

  The voltage applied to analog input pin ANn during A/D conversion should be in the
  $AVss \leq V_{AN} \leq AVref$.
- Relation between AVcc, AVss and Vcc, Vss

  The relationship between AVcc, AVss and Vcc, Vss should be Avcc = Vcc ± 0.3V an
  Vss. When the A/D converter is not used, set AVcc = Vcc and Avss = Vss.
- AVref pin reference voltage specification range

  The reference voltage of the AVref pin should be in the range AVref ≤ AVcc.

19.7.5 Notes on Board Design

In board design, digital circuitry and analog circuitry should be as mutually isolated as po
and layout in which digital circuit signal lines and analog circuit signal lines cross or are
proximity should be avoided as far as possible. Failure to do so may result in incorrect op
of the analog circuitry due to inductance, adversely affecting A/D conversion values. Also
circuitry must be isolated from the analog input signals (AN0 to AN7), the analog referen
voltage (AVref) and analog power supply (AVcc) by the analog ground (AVss). Also, the
ground (AVss) should be connected at one point to a stable digital ground (Vss) on the bo

the analog input pin voltage. Therefore, careful consideration is required upon deciding constants.

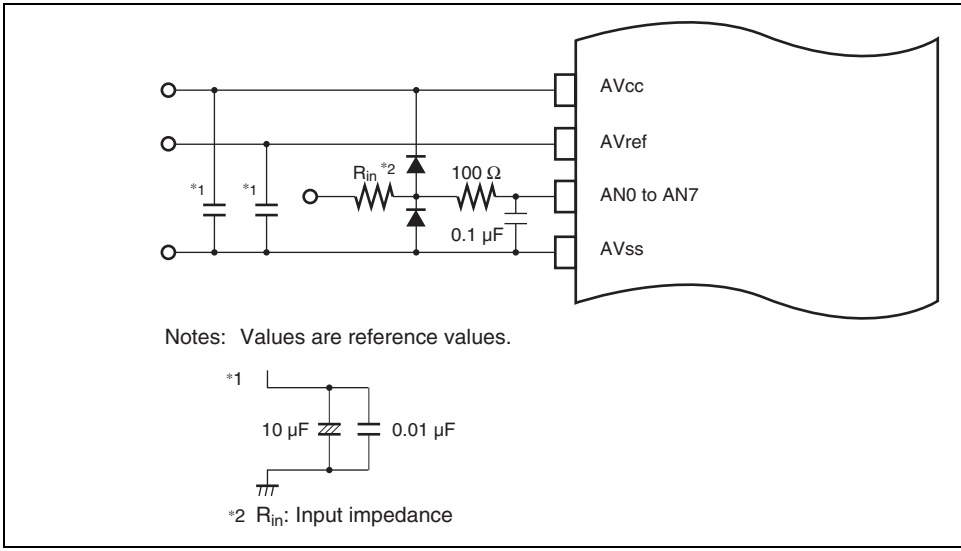


Figure 19.9 Example of Analog Input Protection Circuit

Note: Values are reference values.

Figure 19.10 Analog Input Pin Equivalent Circuit

19.7.7 Note on the Usage in Software Standby Mode

If this LSI enters software standby mode with the A/D conversion enabled, the content of converter is retained and about the same amount of analog supply current may flow as tha when A/D conversion in progress. If the analog supply current must be reduced in softwa standby mode, clear the ADST bit to disable the A/D conversion.

This LSI has a dedicated programming/erasing program. After downloading this pro
the on-chip RAM, programming/erasing can be performed by setting the argument p

- Programming/erasing time

  The flash memory programming time is 1 ms (typ) in 128-byte simultaneous prograı
  approximately 7.8 µs per byte. The erasing time is 600 ms (typ) per 64-Kbyte block.

- Number of programming

  The number of flash memory programming can be up to 100 times at the minimum.
  ranged from 1 to 100 is guaranteed.)

- 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.

  — User boot mode

    The user boot program of the optional interface can be made and the user MAT c
    programmed.

- Programming/erasing protection

  Sets protection against flash memory programming/erasing via hardware, software, c
  protection.

- Programmer mode

  This mode uses the PROM programmer. The user MAT and user boot MAT can be
  programmed.

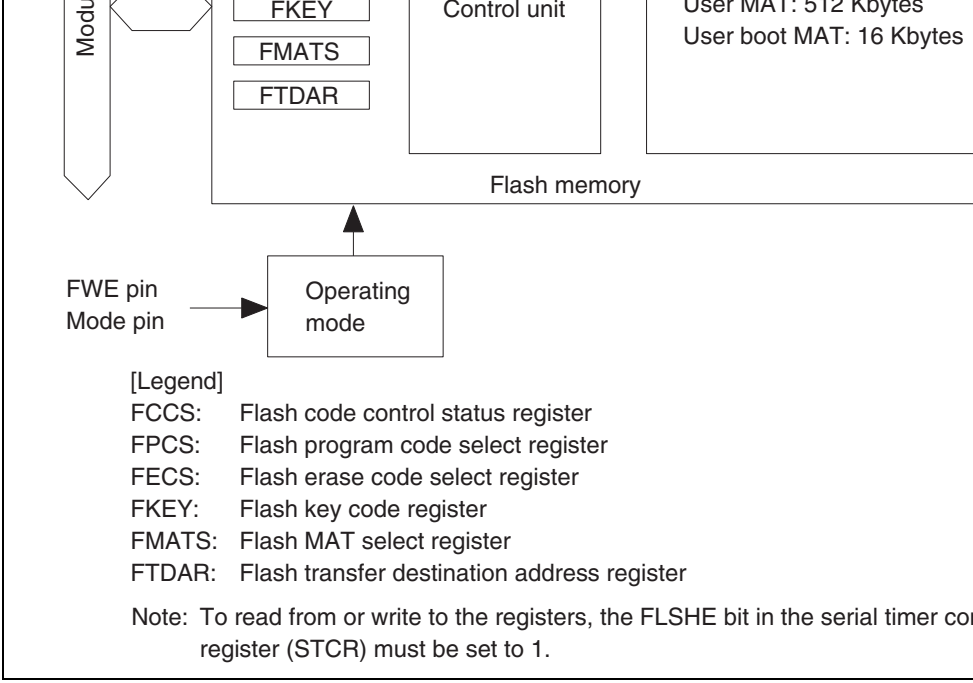


Figure 21.1 Block Diagram of Flash Memory

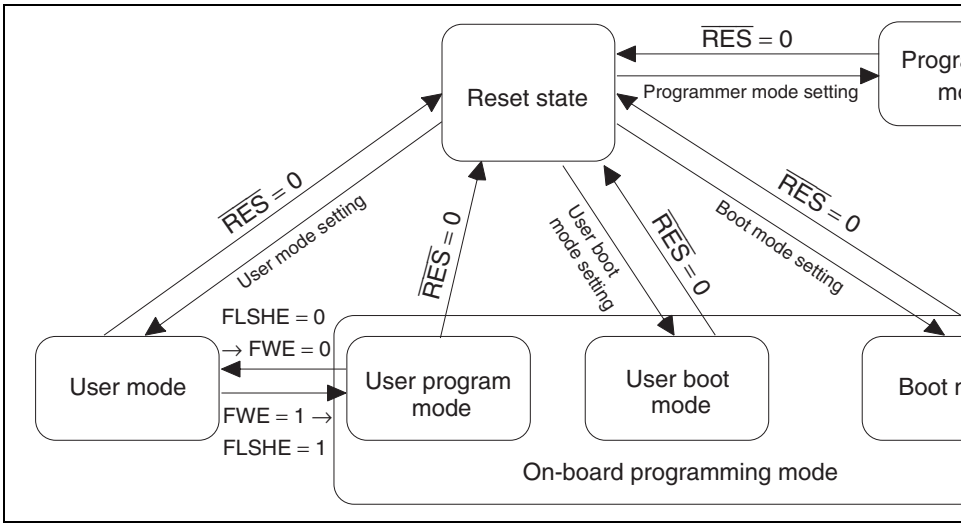


Figure 21.2 Mode Transition of Flash Memory

Programming/ erasing enable MAT	User MAT User boot MAT	User MAT	User MAT	User MAT User boot
All erasure	○ (Automatic)	○	○	○ (Autom
Block division erasure	○ *1	○	○	×
Program data transfer	From host via SCI	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 boot MAT can be programmed by means of the command method. However, the cont the 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 settin different from user program mode in user boot mode.

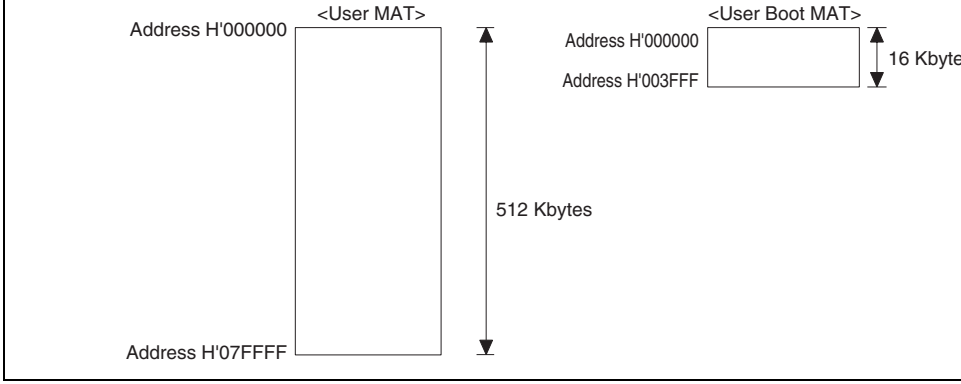


Figure 21.3 Flash Memory Configuration

The size of the user MAT is different from that of the user boot MAT. An address which
the size of the 16-Kbyte user boot MAT should not be accessed. If the attempt is made,
read as undefined value.

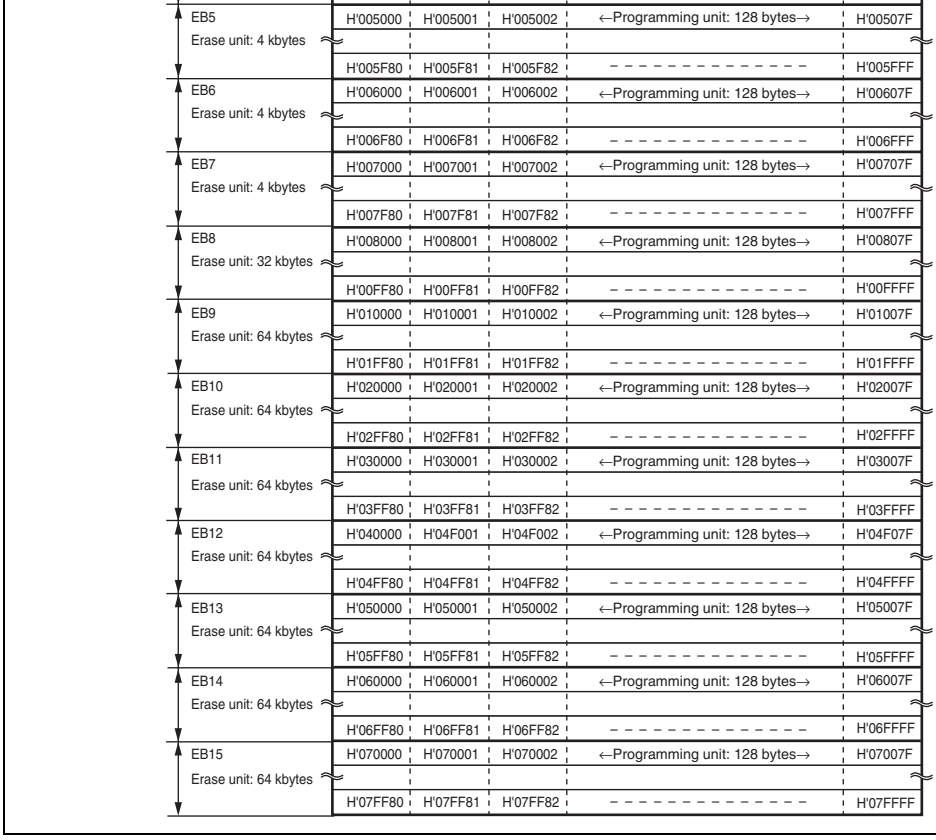


Figure 21.4 Block Division of User MAT

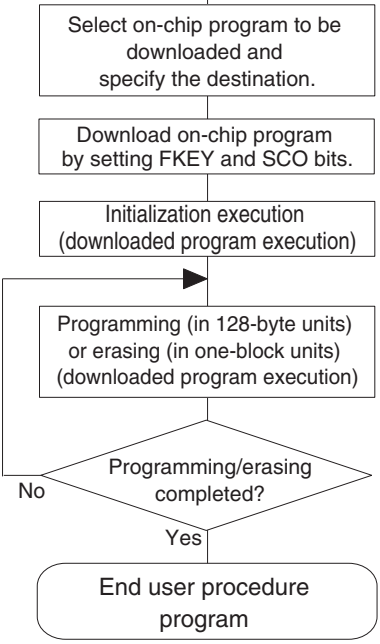


Figure 21.5 Overview of User Procedure Program

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 i programming/erasing interface register. The address of the programming destination i specified by the flash transfer destination address register (FTDAR).

3. Initialization of programming/erasing

   The operating frequency is set before execution of programming/erasing. This 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.

MD0	Input	Sets operating mode of this LSI
TxD1	Output	Serial transmit data output (used in boot mode)
RxD1	Input	Serial receive data input (used in boot mode)

21.3 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.

	FKEY	○	—	○	○	—
	FMATS	—	—	○ *1	○ *1	○
	FTDAR	○	—	—	—	—
Programming/ Erasing Interface Parameter	DPFR	○	—	—	—	—
	FPFR	—	○	○	○	—
	FPEFEQ	—	○	—	—	—
	FMPAR	—	—	○	—	—
	FMPDR	—	—	○	—	—
	FEBS	—	—	—	○	—

Notes: 1. The setting is required when programming or erasing user MAT in user boot m
2. The setting may be required according to the combination of initiation mode a target MAT.

7	FWE	1/0	R	Flash Program Enable Monitors the signal level input to the FWE pin a enables or disables programming/erasing flash 0: Programming/erasing disabled 1: Programming/erasing enabled
6, 5	—	All 0	R/W	Reserved The initial value should not be changed.

Programming/erasing protection for flash me
(error protection) is invalid.

[Clearing condition]

- At a reset or in hardware standby mode

1: An error occurs during programming/erasing
memory.
Programming/erasing protection for flash me
(error protection) is valid.

[Setting conditions]

- When an interrupt, such as NMI, occurs d
programming/erasing flash memory.
- When the flash memory is read during
programming/erasing flash memory (includ
vector read or an instruction fetch).
- When the SLEEP instruction is executed d
programming/erasing flash memory (includ
software-standby mode)
- When a bus master other than the CPU, s
DTC, gets bus mastership during
programming/erasing flash memory.

				The interrupt exception handling on and after ve number 32 should not be used because the cor vector is not read, resulting in the CPU runaway 0: The space for the interrupt vector table is not When interrupt vector data is not read succes the operation for the interrupt exception hand cannot be guaranteed. An occurrence of any should be masked. 1: The space for the interrupt vector table is mo Even when interrupt vector data is not read successfully, the interrupt exception handling vector number 31 is enabled.
2, 1	—	All 0	R/W	Reserved The initial value should not be changed.

		after setting this bit to 1. Since this bit is cleared to 0 when download is completed, this bit cannot be read as 1. All interrupts must be disabled. This should be the user system. 0: Download of the on-chip programming/eras program to the on-chip RAM is not executed. [Clearing condition] When download is completed 1: Request that the on-chip programming/eras program is downloaded to the on-chip RAM occurred. [Setting conditions] When all of the following conditions are satisfi set to this bit • H'A5 is written to FKEY • During execution in the on-chip RAM

Note: * This bit is a write only bit. This bit is always read as 0.

				[Clearing condition] When transfer is completed
				1: On-chip programming program is selected.

- Flash Erase Code Select Register (FECS)

FECS selects download of the on-chip erasing program.

Bit	Bit Name	Initial Value	R/W	Description
7 to 1	—	All 0	R/W	Reserved
				The initial value should not be changed.
0	EPVB	0	R/W	Erase Pulse Verify Block
				Selects the erasing program.
				0: On-chip erasing program is not selected.
				[Clearing condition] When transfer is completed
				1: On-chip erasing program is selected.

4	K4	0	R/W	
3	K3	0	R/W	
2	K2	0	R/W	
1	K1	0	R/W	
0	K0	0	R/W	

cannot be set to the SCO bit. Therefore downl
the on-chip RAM cannot be executed.

Only when H'5A is written, programming/erasi
executed. Even if the on-chip programming/er
program is executed, the flash memory cannot
programmed or erased when the value other t
is written to FKEY.

H'A5: Writing to the SCO bit is enabled. (The S
cannot be set by the value other than H'

H'5A: Programming/erasing is enabled. (The v
than H'A5 is in software protection state

H'00: Initial value

2	MS2	0	R/W	Switching between User MAT and User Boot MA user boot MAT cannot be programmed in user p mode if user boot MAT is selected by FMATS. T boot MAT must be programmed in boot mode o programmer mode.) H'AA: The user boot MAT is selected (in user-M selection state when the value of these bi other than H'AA) Initial value when these bits are initiated i boot mode. H'00: Initial value when these bits are initiated i except for user boot mode (in user-MAT s state) [Programmable condition] These bits are in the execution state in the on-c
1	MS1	0/1*	R/W	
0	MS0	0	R/W	

Note: * Set to 1 when in user boot mode, otherwise set to 0.

				of H'00 to H'03 is determined when an on-chip downloaded by setting the SCO bit in FCCS to sure that this bit is cleared to 0 before setting t to 1 and the value specified by TDA6 to TDA0 the range of H'00 to H'03. 0: The value specified by bits TDA6 to TDA0 is range. 1: The value specified by is TDA6 to TDA0 is o range (H'04 to H'FF) and the download is st
6	TDA6	0	R/W	Transfer Destination Address
5	TDA5	0	R/W	Specifies the start address to download an on- program. H'00 to H'03 can be specified as the address in the on-chip RAM space. H'00: H'FFE080 is specified as a start address download an on-chip program. H'01: H'FF0800 is specified as a start address download an on-chip program. H'02: H'FF1800 is specified as a start address download an on-chip program. H'03: H'FF8800 is specified as a start address download an on-chip program. H'04 to H'FF: Setting prohibited. Specifying th sets the TDER bit to 1 and stop download.
4	TDA4	0	R/W	
3	TDA3	0	R/W	
2	TDA2	0	R/W	
1	TDA1	0	R/W	
0	TDA0	0	R/W	

(A maximum size of a stack area to be used is 128 bytes.)

The programming/erasing interface parameter is used in the following four items.

1. Download control
2. Initialization before programming or erasing
3. Programming
4. Erasing

These items use different parameters. The correspondence table is shown in table 21.4. Th
meaning of the bits in FPFR varies in each processing program: initialization, programmi
erasure. For details, see descriptions of FPFR for each process.

Flash multipurpose address area	FMPAR	—	—	○	—	R/W	Undefined
Flash multipurpose data destination area	FMPDR	—	—	○	—	R/W	Undefined
Flash erase block select	FEBS	—	—	—	○	R/W	Undefined

Note: * A single byte of the start address to download an on-chip program, which is s
FTDAR

be used to determine if downloading is executed or not. Since the confirmation whether t
bit is set to 1 is difficult, the certain determination must be performed by writing the sing
the start address specified by FTDAR to the value other than the return value of downloa
example, H'FF) before the download start (before setting the SCO bit to 1).

Bit	Bit Name	Initial Value	R/W	Description
7 to 3	—	—	—	Unused Return 0
2	SS	—	R/W	Source Select Error Detect Only one type for the on-chip program which ca downloaded can be specified. When more than types of the program are selected, the program selected, or the program is selected without ma error is occurred. 0: Download program can be selected normally 1: Download error is occurred (multi-selection o program which is not mapped is selected)
1	FK	—	R/W	Flash Key Register Error Detect Returns the check result whether the value of F set to H'A5. 0: KEY setting is normal (FKEY = H'A5) 1: Setting value of FKEY becomes error (FKEY other than H'A5)

CPU)

This parameter sets the operating frequency of the CPU. The settable range of the operati
frequency in this LSI is 20 to 34 MHz.

Bit	Bit Name	Initial Value	R/W	Description
31 to 16	—	—	—	Unused This bit should be cleared to 0.
15 to 0	F15 to F0	—	R/W	Frequency Set Set the operating frequency of the CPU. With th multiplication function, set the frequency multipl setting value must be calculated as the followin methods. 1. The operating frequency which is shown in M must be rounded in a number to three decim and be shown in a number of two decimal pl 2. The value multiplied by 100 is converted to t digit and is written to the FPEFEQ paramete (general register ER0). For example, when the operating frequency of t is 33.000 MHz, the value is as follows. 1. The number to three decimal places of 34.0 rounded and the value is thus 34.00. 2. The formula that 34.00 × 100 = 3400 is conv the binary digit and B'0000,1101,0100,1000 is set to ER0.

				operating frequency. 0: Setting of operating frequency is normal 1: Setting of operating frequency is abnormal
0	SF	—	R/W	Success/Fail Indicates whether initialization is completed no 0: Initialization is ended normally (no error) 1: Initialization is ended abnormally (error occu

(3) Programming Execution

When flash memory is programmed, the programming destination address on the user M be passed to the programming program in which the program data is downloaded.

1. The start address of the programming destination on the user MAT must be stored in register ER1. This parameter is called as flash multipurpose address area parameter (
   Since the program data is always in units of 128 bytes, the lower eight bits (A7 to A0 H'00 or H'80 as the boundary of the programming start address on the user MAT.
2. The program data for the user MAT must be prepared in the consecutive area. The p data must be in the consecutive space which can be accessed by using the MOV.B in of the CPU and in other than the flash memory space.
   When data to be programmed does not satisfy 128 bytes, the 128-byte program data prepared by filling with the dummy code H'FF.
   The start address of the area in which the prepared program data is stored must be st general register ER0. This parameter is called as flash multipurpose data destination parameter (FMPDR).

For details on the program processing procedure, see section 21.4.2, User Program Mod

31 to 0	MOA31 to MOA0	—	R/W	Store the start address of the programming des on the user MAT. The consecutive 128-byte programming is executed starting from the spec address of the user MAT. Therefore, the specifi programming start address becomes a 128-byte boundary and MOA6 to MOA0 are always 0.

(b) Flash multipurpose data destination parameter (FMPDR: general register ER0 CPU):

This parameter stores the start address in the area which stores the data to be programmed user MAT. When the storage destination of the program data is in flash memory, an error The error occurrence is indicated by the WD bit in FPFR.

Bit	Bit Name	Initial Value	R/W	Description
31 to 0	MOD31 to MOD0	—	R/W	Store the start address of the area which stores program data for the user MAT. The consecutiv byte data is programmed to the user MAT starti the specified start address.

(c) Flash pass/fail parameter (FPFR: general register R0L of CPU)

This parameter indicates the return value of the program processing result.

Bit	Bit Name	Initial Value	R/W	Description
7	—	—	—	Unused Return 0.

				1: Programming cannot be performed (FWE FLER = 1)
5	EE	—	R/W	Programming Execution Error Detect 1 is returned to this bit when the specified data be written because the user MAT was not eras bit is set to 1, there is a high possibility that the is partially rewritten. In this case, after removin factor, erase the user MAT. If FMATS is set to H'AA and the user boot MA selected, an error occurs when programming is performed. In this case, both the user MAT an MAT are not rewritten. Programming of the use MAT should be performed in boot mode or pro mode. 0: Programming has ended normally 1: Programming has ended abnormally (prog result is not guaranteed)
4	FK	—	R/W	Flash Key Register Error Detect Returns the check result of the value of FKEY start of the programming processing. 0: FKEY setting is normal (FKEY = H'5A) 1: FKEY setting is error (FKEY = value other th
3	—	—	—	Unused Returns 0.

				• When the programming destination address area other than flash memory is specified • When the specified address is not in a 128- boundary. (The lower eight bits of the addre other than H'00 and H'80.) 0: Setting of programming destination address i 1: Setting of programming destination address i abnormal
0	SF	—	R/W	Success/Fail Indicates whether the program processing is en normally or not. 0: Programming is ended normally (no error) 1: Programming is ended abnormally (error occ

Bit	Bit Name	Initial Value	R/W	Description
31 to 16	—	—	—	Unused These bits should be cleared to H′0.
15	EB15	—	R/W	Erase Block Set the erase-block number in the range from 0 to 15. 0 corresponds to the EB0 block, and 15 corresponds to the EB15 block.
14	EB14	—	R/W	
13	EB13	—	R/W	
12	EB12	—	R/W	
11	EB11	—	R/W	
10	EB10	—	R/W	
9	EB9	—	R/W	
8	EB8	—	R/W	
7	EB7	—	R/W	
6	EB6	—	R/W	
5	EB5	—	R/W	
4	EB4	—	R/W	
3	EB3	—	R/W	
2	EB2	—	R/W	
1	EB1	—	R/W	
0	EB0	—	R/W	

				entered. When the low-level signal is input to the pin or the error protection state is entered, 1 is w this bit. The state can be confirmed with the FW FLER bits in FCCS. For conditions to enter the protection state, see section 21.5.3, Error Prote 0: FWE and FLER settings are normal (FWE = FLER = 0) 1: Programming cannot be performed (FWE = FLER = 1)
5	EE	—	R/W	Erasure Execution Error Detect 1 is returned to this bit when the user MAT coul erased or when flash-memory related register s are partially changed. If this bit is set to 1, there possibility that the user MAT is partially erased. case, after removing the error factor, erase the MAT. If FMATS is set to H'AA and the user boo selected, an error occurs when erasure is perfo this case, both the user MAT and user boot MA erased. Erasing of the user boot MAT should be performed in boot mode or programmer mode. 0: Erasure has ended normally 1: Erasure has ended abnormally (erasure res guaranteed)
4	FK	—	R/W	Flash Key Register Error Detect Returns the check result of FKEY value before s the erasing processing. 0: FKEY setting is normal (FKEY = H'5A) 1: FKEY setting is error (FKEY = value other tha

	Indicates whether the erasing processing is en normally or not. 0: Erasure is ended normally (no error) 1: Erasure is ended abnormally (error occurs)

21.4 On-Board Programming Mode

When the pin is set in on-board programming mode and the reset start is executed, the o programming state that can program/erase the on-chip flash memory is entered. On-boar programming mode has three operating modes: boot mode, user program mode, and use mode.

For details of the pin setting for entering each mode, see table 21.5. For details of the sta transition of each mode for flash memory, see figure 21.2.

Table 21.5 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	0/1
User boot mode	1	0	0	0	0

Note: * Before downloading the programming/erasing programs, the FLSHE bit must 1 to transition to user program mode.

setting in boot mode, see table 21.5. The NMI and other interrupts are ignored in boot mo
However, the NMI and other interrupts should be disabled in the user system.

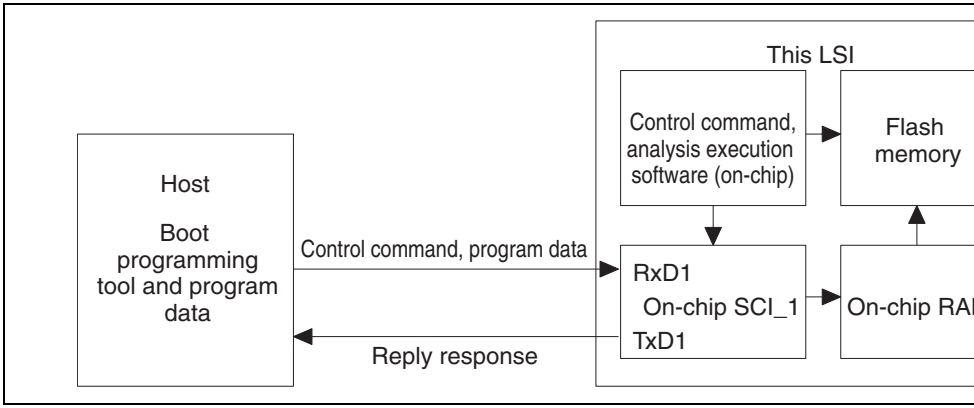


Figure 21.6 System Configuration in Boot Mode

set to 9,600 bps or 19,200 bps.

The system clock frequency, which can automatically adjust the transfer bit rate of the h
the bit rate of this LSI, is shown in table 21.6. Boot mode must be initiated in the range
system clock.

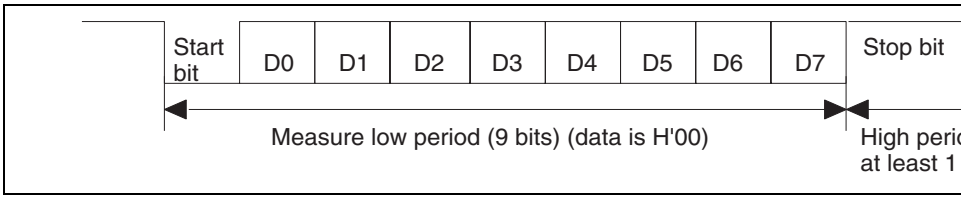


Figure 21.7 Automatic-Bit-Rate Adjustment Operation of SCI

Table 21.6 System Clock Frequency for Automatic-Bit-Rate Adjustment by This

Bit Rate of Host	System Clock Frequency
9,600 bps	20 to 34 MHz
19,200 bps	

After inquiries have finished, all user MAT and user boot MAT are automatically era

4. Waiting for programming/erasing command
   - When the program preparation notice is received, the state for waiting program da entered. The programming start address and program data must be transmitted foll the programming command. When programming is finished, the programming sta must be set to H'FFFFFFFF and transmitted. Then the state for waiting program d returned to the state of programming/erasing command wait.
   - When the erasure preparation notice is received, the state for waiting erase-block d entered. The erase-block number must be transmitted following the erasing comm When the erasure is finished, the erase-block number must be set to H'FF and tran Then the state for waiting erase-block data is returned to the state for waiting programming/erasing command. The erasure must be used when the specified blo programmed without a reset start after programming is executed in boot mode. Wh programming can be executed by only one operation, all blocks are erased before for waiting programming/erasing/other command is entered. The erasing operation required.
   - There are many commands other than programming/erasing. Examples are sum ch blank check (erasure check), and memory read of the user MAT/user boot MAT a acquisition of current status information.

Note that memory read of the user MAT/user boot MAT can only read the programmed d all user MAT/user boot MAT has automatically been erased.

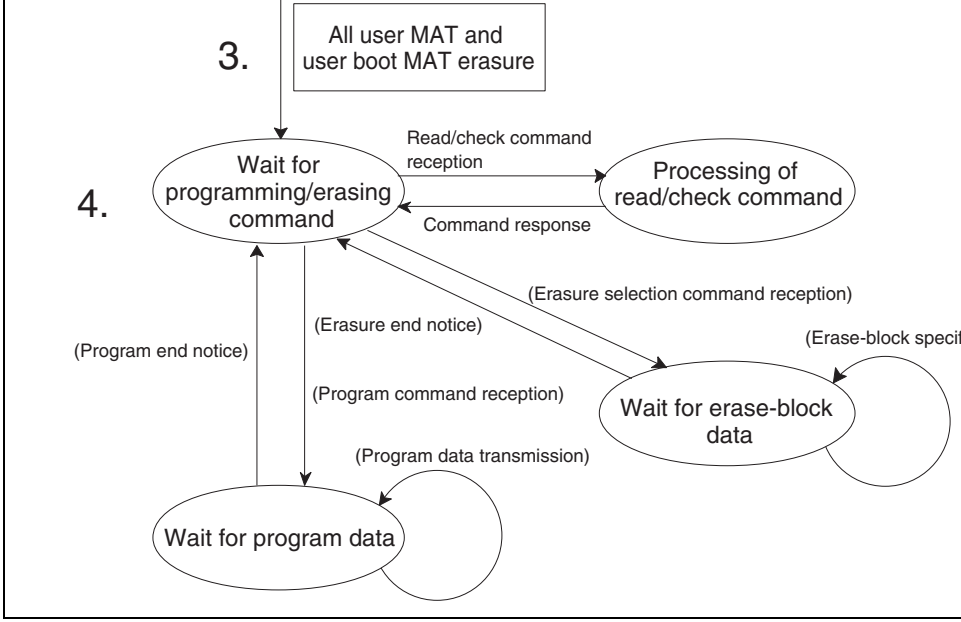


Figure 21.8 Overview of Boot Mode State Transition Diagram

period of 100 µs which is longer than normal.

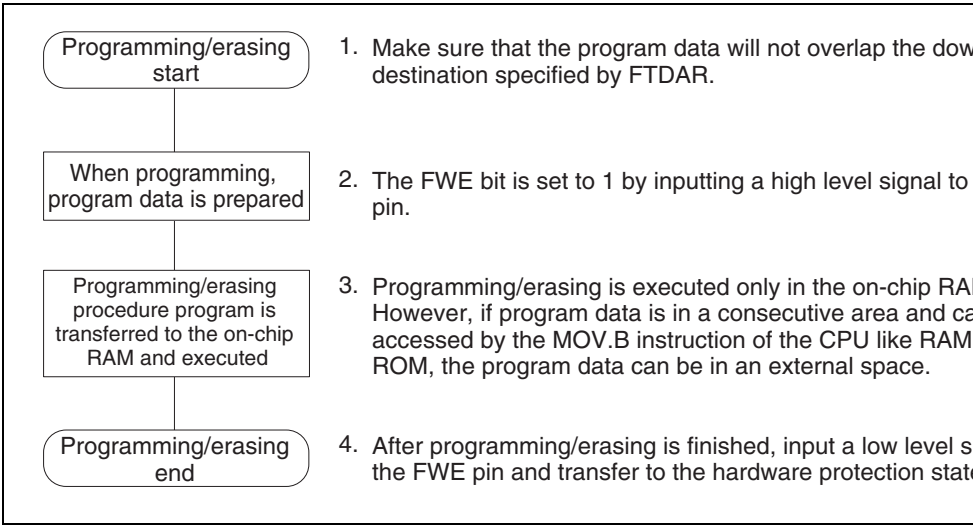


Figure 21.9 Programming/Erasing Overview Flow

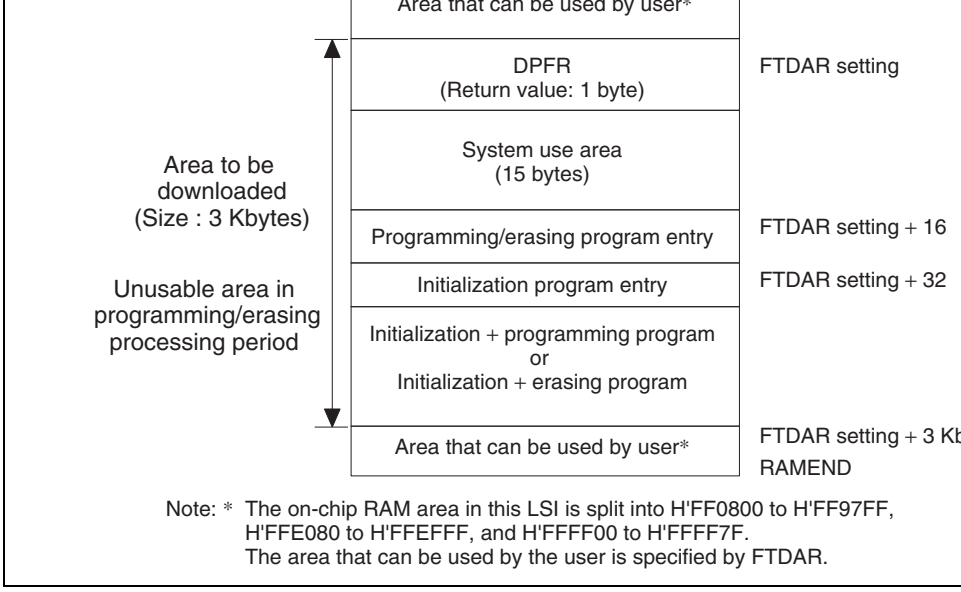


Figure 21.10 RAM Map When Programming/Erasing is Executed

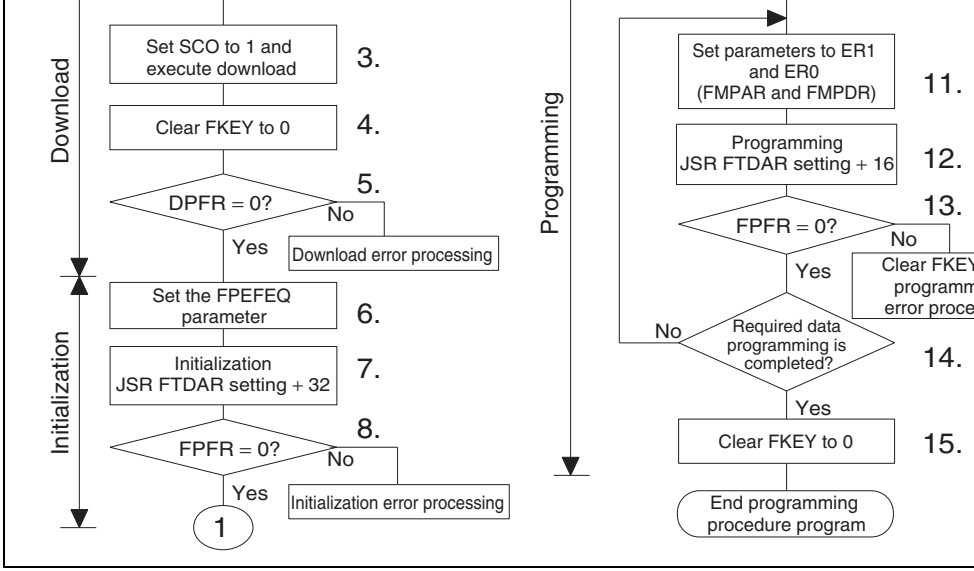


Figure 21.11 Programming Procedure

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 21.4.4, 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.

download is not performed and a download error is returned to the SS bit in DPFR. T
address of a download destination is specified by FTDAR.

2. Program H'A5 in FKEY

   If H'A5 is not written to FKEY for protection, 1 cannot be set to the SCO bit for dow
   request.

3. 1 is set to the SCO bit of FCCS and then download is executed.

   To set 1 to the SCO bit, the following conditions must be satisfied.

   — H'A5 is written to FKEY.

   — The SCO bit writing is executed in the on-chip RAM.

   When the SCO bit is set to 1, download is started automatically. When the SCO bit i
   to the user procedure program, the SCO is cleared to 0. Therefore, the SCO bit canno
   confirmed to be 1 in the user procedure program.

   The download result can be confirmed only by the return value of DPFR. Before the
   is set to 1, incorrect determination must be prevented by setting the one byte of the s
   address (to be used as DPFR) specified by FTDAR to a value other than the return v
   H'FF).

   When download is executed, particular interrupt processing, which is accompanied b
   switch as described below, is performed as an internal microcomputer processing. Fo
   instructions are executed immediately after the instructions that set the SCO bit to 1.

   — The user-MAT space is switched to the on-chip program storage area.

   — After the selection condition of the download program and the FTDAR setting ar
   the transfer processing to the on-chip RAM specified by FTDAR is executed.

   — The SCO bit in FCCS is cleared to 0.

   — The return value is set to the DPFR parameter.

— Since a stack area of 128 bytes at the maximum is used, the area must be allocated setting the SCO bit to 1.

— If a flash memory access by the DTC signal is requested during downloading, the cannot be guaranteed. Therefore, an access request by the DTC signal must not be generated.

4. FKEY is cleared to H'00 for protection.

5. The value of the DPFR parameter must be checked and the download result must be confirmed.

— Check the value of the DPFR parameter (one byte of start address of the downloac destination specified by FTDAR). If the value is H'00, download has been 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 is set in the FPEFEQ parameter for initialization.

— The current frequency of the CPU clock is set to the FPEFEQ parameter value (ge register ER0).

The settable range of the FPEFEQ parameter is 21 to 34 MHz. When the frequenc out of this range, an error is returned to the FPFR parameter of the initialization pr and initialization is not performed. For details on the frequency setting, see the de in 21.3.2 (2) (a), Flash programming/erasing frequency parameter (FPEFEQ: gene register ER0 of CPU).

— R0L is a return value of the FPFR parameter.
— Since the stack area is used in the initialization program, 128-byte stack area at th
maximum must be allocated in RAM.
— Interrupts can be accepted during the execution of the initialization program. The
storage area and stack area in the on-chip RAM and register values must not be d

8. The return value in the initialization program, FPFR (general register R0L) is determ

9. All interrupts and the use of a bus master other than the CPU are prohibited.
The specified voltage is applied for the specified time when programming or erasing
interrupts occur or the bus mastership is moved to other than the CPU during this tim
voltage for more than the specified time will be applied and flash memory may be da
Therefore, interrupts and bus mastership to other than the CPU, such as to the DTC,
prohibited.

To disable interrupts, bit 7 (I) in the condition code register (CCR) of the CPU shoul
B'1 in interrupt control mode 0 or bits 7 and 6 (I and UI) should be set to B'11 in inte
control mode 1. Interrupts other than NMI are held and not executed.

The NMI interrupts must be masked within the user system.

The interrupts that are held must be executed after all program processing.

When the bus mastership is moved to other than the CPU, such as to the DTC, the en
protection state is entered. Therefore, taking bus mastership by the DTC is prohibited

10. FKEY must be set to H′5A and the user MAT must be prepared for programming.

— Example of the FMPDR setting

When the storage destination of the program data is flash memory, even if the pro execution routine is executed, programming is not executed and an error is returne FPFR parameter. In this case, the program data must be transferred to the on-chip and then programming must be executed.

12. Programming

There is an entry point of the programming program in the area from the start address by FTDAR + 16 bytes of the on-chip RAM. The subroutine is called and programmin executed by using the following steps.

```
MOV.L   #DLTOP+16,ER2          ; Set entry address to ER2
JSR     @ER2                   ; Call programming routine
NOP
```

— The general registers other than R0L are held in the programming program.

— R0L is a return value of the FPFR parameter.

— Since the stack area is used in the programming program, a stack area of 128 byte maximum must be allocated in RAM.

13. The return value in the programming program, FPFR (general register R0L) is determ

14. Determine whether programming of the necessary data has finished.

If more than 128 bytes of data are to be programmed, specify FMPAR and FMPDR in byte units, and repeat steps 12 to 14. Increment the programming destination address bytes and update the programming data pointer correctly. If an address which has alre programmed is written to again, not only will a programming error occur, but also fla memory will be damaged.

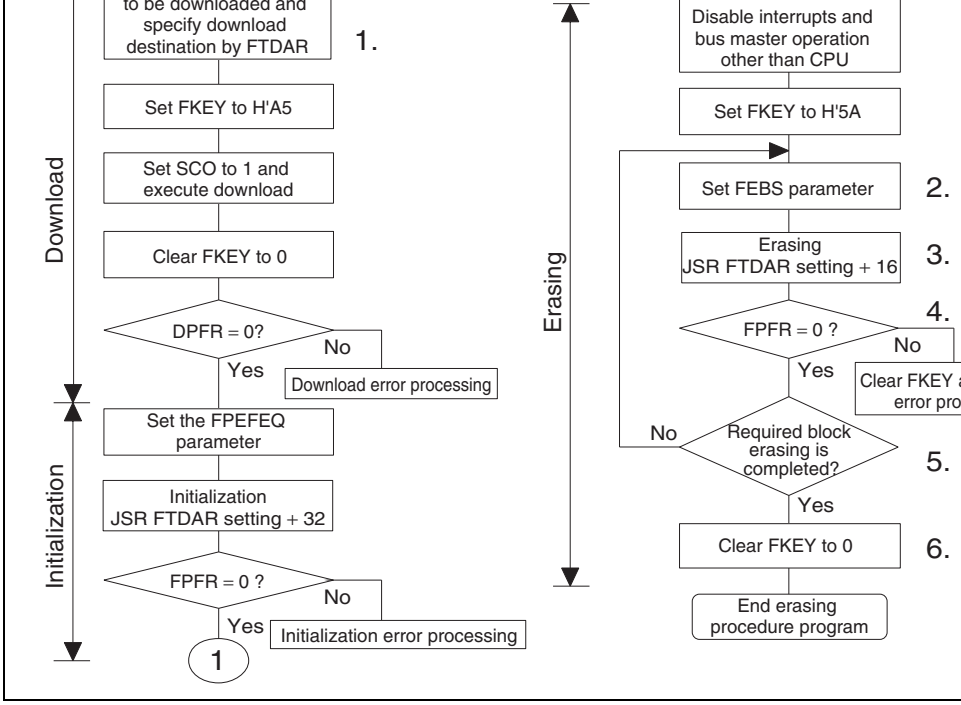


Figure 21.12 Erasing Procedure

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 21.4.4, Procedure Program and Storable A
Programming Data.

parameter.

Specify the start address of a download destination by FTDAR.

The procedures to be carried out after setting FKEY, e.g. download and initialization, same as those in the programming procedure. For details, refer to section 21.4.2 (2), Programming Procedure in User Program Mode.

The procedures after setting parameters for erasing programs are as follows:

2. Set the FEBS parameter necessary for erasure

   Set the erase block number of the user MAT in the flash erase block select parameter (general register ER0). If a value other than an erase block number of the user MAT i block is erased even though the erasing program is executed, and an error is returned return value parameter FPFR.

3. Erasure

   Similar to as in programming, there is an entry point of the erasing program in the are the start address of a download destination specified by FTDAR + 16 bytes of on-chip The subroutine is called and erasing is executed by using the following steps.

```
MOV.L   #DLTOP+16,ER2          ; Set entry address to ER2
JSR     @ER2                   ; Call erasing routine
NOP
```

   - The general registers other than R0L are held in the erasing program.
   - R0L is a return value of the FPFR parameter.
   - Since the stack area is used in the erasing program, a stack area of 128 bytes at the maximum must be allocated in RAM.

4. The return value in the erasing program, FPFR (general register R0L) is determined.

Figure 21.13 shows a repeating procedure of erasing and programming.

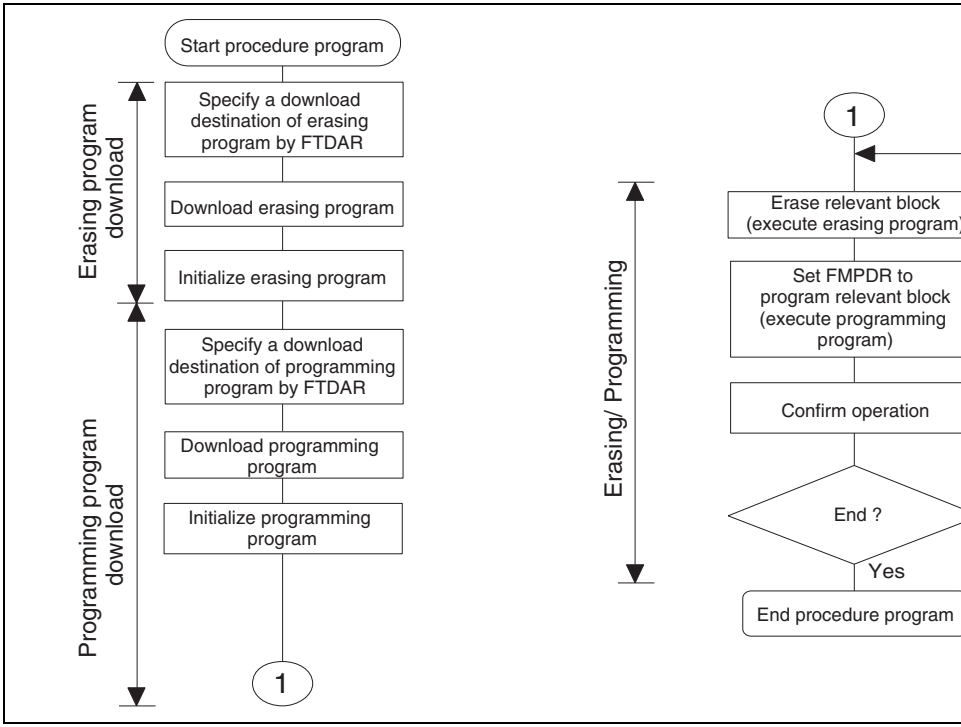


Figure 21.13 Repeating Procedure of Erasing and Programming

In the above procedure, download and initialization are performed only once at the begi

In this kind of operation, note the following:

For the mode pin settings to start up user boot mode, see table 21.5.

When the reset start is executed in user boot mode, the built-in check routine runs. The u
and user boot MAT states are checked by this check routine.

While the check routine is running, NMI and all other interrupts cannot be accepted.

Next, processing starts from the execution start address of the reset vector in the user bo
At this point, H′AA is set to FMATS because the execution MAT is the user boot MAT.

(2) User MAT Programming in User Boot Mode

For programming the user MAT in user boot mode, additional processing made by settin
are required: switching from user-boot-MAT selection state to user-MAT selection state
switching back to user-boot-MAT selection state after programming completes.

Figure 21.14 shows the procedure for programming the user MAT in user boot mode.

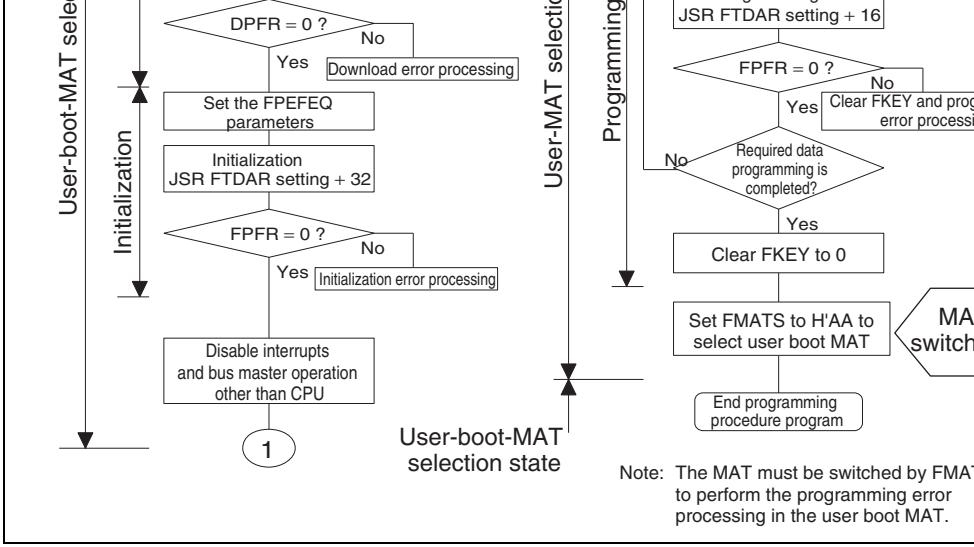


Figure 21.14 Procedure for Programming User MAT in User Boot Mode

The difference between the programming procedures in user program mode and user boo whether the MAT is switched or not as shown in figure 21.14.

In user boot mode, the user boot MAT can be seen in the flash memory space with the us hidden in the background. The user MAT and user boot MAT are switched only while the MAT is being programmed. Because the user boot MAT is hidden while the user MAT is programmed, the procedure program must be located in an area other than flash memory. programming completes, switch the MATs again to return to the first state.

MAT switching is enabled by writing a specific value to FMATS. However note that whi MATs are being switched, the LSI is in an unstable state, e.g. access to a MAT is not allo MAT switching is completed, and if an interrupt occurs, from which MAT the interrupt v

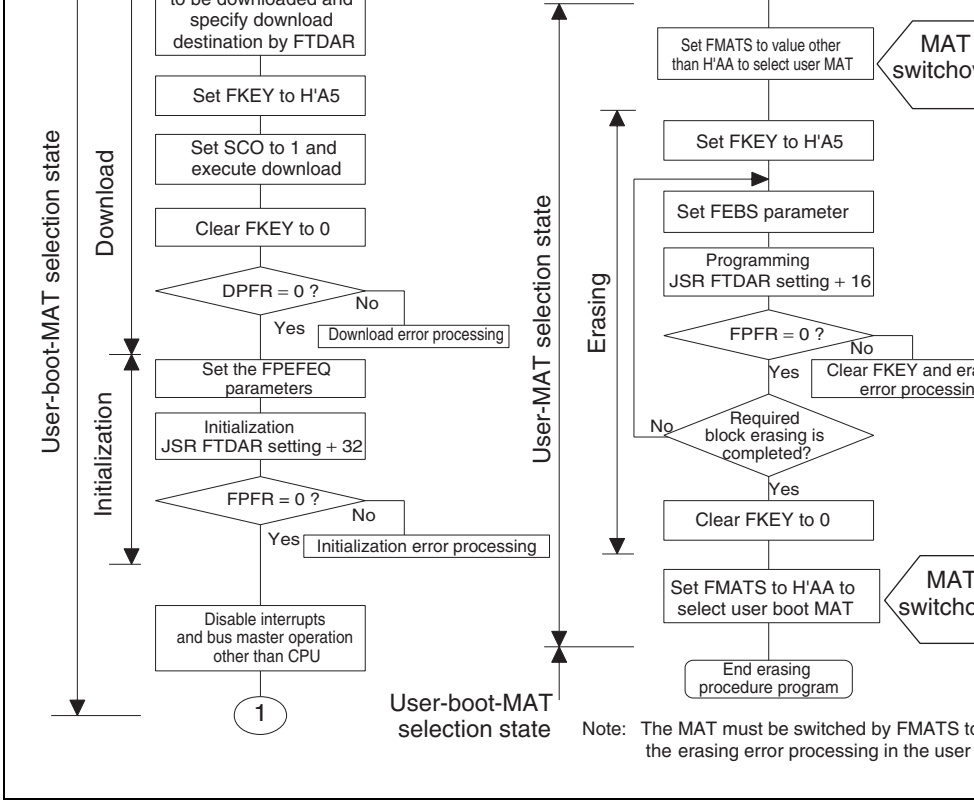


Figure 21.15 Procedure for Erasing User MAT in User Boot Mode

The difference between the erasing procedures in user program mode and user boot mode on whether the MAT is switched or not as shown in figure 21.15.

2. The on-chip programming/erasing program will use 128 bytes at the maximum as a st
make sure that this area is secured.
3. Download by setting the SCO bit to 1 will lead to switching of the MAT. If, therefore
operation is used, it should be executed from the on-chip RAM.
4. The flash memory is accessible until the start of programming or erasing, that is, until
result of downloading has been determined. When in a mode in which the external ad
space is not accessible, such as single-chip mode, the required procedure programs, N
handling vector and NMI handler should be transferred to the on-chip RAM before
programming/erasing of the flash memory starts.
5. The flash memory is not accessible during programming/erasing operations, therefore
operation program is downloaded to the on-chip RAM to be executed. The NMI-hand
vector and programs such as that which activate the operation program, and NMI han
should thus be stored in on-chip memory other than flash memory or the external addr
space.
6. After programming/erasing, the flash memory should be inhibited until FKEY is clear
The reset state ($\overline{RES}$ = 0) must be in place for more than 100 µs when the LSI mode is
to reset on completion of a programming/erasing operation.
Transitions to the reset state, and hardware standby mode are inhibited during
programming/erasing. When the reset signal is accidentally input to the chip, a longer
the reset state than usual (100 µs) is needed before the reset signal is released.
7. Switching of the MATs by FMATS should be needed when programming/erasing of t
boot MAT is operated in user-boot mode. The program which switches the MATs sho
executed from the on-chip RAM. See section 21.6, Switching between User MAT and
Boot MAT. Please make sure you know which MAT is selected when switching betw
them.
8. When the data storable area indicated by programming parameter FMPDR is within th
memory area, an error will occur even when the data stored is normal. Therefore, the

Programming	Table 21.8 (1)	Table 21.8 (3)
Erasing	Table 21.8 (2)	Table 21.8 (4)

Note: * Programming/Erasing is possible to user MATs.

Operation for Writing H'A5 to FKEY	○	○	○	○
Execution of Writing SCO = 1 to FCCS (Download)	○	×	×	
Operation for FKEY Clear	○	○	○	○
Determination of Download Result	○	○	○	○
Operation for Download Error	○	○	○	○
Operation for Settings of Initial Parameter	○	○	○	○
Execution of Initialization	○	×	×	○
Determination of Initialization Result	○	○	○	○
Operation for Initialization Error	○	○	○	○
NMI Handling Routine	○	×	○	○
Operation for Inhibit of Interrupt	○	○	○	○
Operation for Writing H'5A to FKEY	○	○	○	○
Operation for Settings of Program Parameter	○	×	○	○

	Clear

Note: * Transferring the data to the on-chip RAM enables this area to be used.

Execution of Writing SCO = 1 to FCCS (Download)	○	×	×	
Operation for FKEY Clear	○	○	○	○
Determination of Download Result	○	○	○	○
Operation for Download Error	○	○	○	○
Operation for Settings of Initial Parameter	○	○	○	○
Execution of Initialization	○	×	×	○
Determination of Initialization Result	○	○	○	○
Operation for Initialization Error	○	○	○	○
NMI Handling Routine	○	×	○	○
Operation for Inhibit of Interrupt	○	○	○	○
Operation for Writing H'5A to FKEY	○	○	○	○
Operation for Settings of Erasure Parameter	○	×	○	○
Execution of Erasure	○	×	×	○
Determination of Erasure Result	○	×	○	○

Downloaded					
Operation for Writing H'A5 to FKEY	○	○	○		○
Execution of Writing SCO = 1 to FCCS (Download)	○	×	×		
Operation for FKEY Clear	○	○	○		○
Determination of Download Result	○	○	○		○
Operation for Download Error	○	○	○		○
Operation for Settings of Initial Parameter	○	○	○		○
Execution of Initialization	○	×	×		○
Determination of Initialization Result	○	○	○		○
Operation for Initialization Error	○	○	○		○
NMI Handling Routine	○	×	○		○
Operation for Interrupt Inhibit	○	○	○		○
Switching MATs by FMATS	○	×	×	○	
Operation for Writing H'5A to FKEY	○	×	○	○	

Operation for Program Error	○	×*2	○	○	
Operation for FKEY Clear	○	×	○	○	
Switching MATs by FMATS	○	×	×		○

Notes: 1. Transferring the data to the on-chip RAM enables this area to be used.
2. Switching FMATS by a program in the on-chip RAM enables this area to be u

Execution of Writing SCO = 1 to FCCS (Download)	○	×	×		
Operation for FKEY Clear	○	○	○		○
Determination of Download Result	○	○	○		○
Operation for Download Error	○	○	○		○
Operation for Settings of Initial Parameter	○	○	○		○
Execution of Initialization	○	×	×		○
Determination of Initialization Result	○	○	○		○
Operation for Initialization Error	○	○	○		○
NMI Handling Routine	○	×	○		○
Operation for Interrupt Inhibit	○	○	○		○
Switching MATs by FMATS	○	×	×		○
Operation for Writing H'5A to FKEY	○	×	○	○	

Erasure Error				
Operation for FKEY Clear	○	×	○	○
Switching MATs by FMATS	○	×	×	○

Note: * Switching FMATS by a program in the on-chip RAM enables this area to be u

user MAT, and the error in programming/erasing is reported in the parameter FPFR.

(including a reset by the WDT) and standby mode and the program/erase-protected state is entered.

- The reset state will not be entered by a reset using the $\overline{\text{RES}}$ pin unless the $\overline{\text{RES}}$ pin is held low until oscillation has stabilized after power is initially supplied. In the case of a reset during operation, hold the $\overline{\text{RES}}$ pin low for the RES pulse width that is specified in the section on AC characteristics section. If a reset is input during programming or erasure, data values in the flash memory are not guaranteed. In this case, execute erasure and then execute program again.

	FCCS which disables the downloading of the programming/erasing programs.		
Protection by the FKEY register	• Downloading and programming/erasing are disabled unless the required key code is written in FKEY. Different key codes are used for downloading and for programming/erasing.	○	○

21.5.3 Error Protection

Error protection is a mechanism for aborting programming or erasure when an error occu
form of the microcomputer entering runaway during programming/erasing of the flash m
operations that are not according to the established procedures for programming/erasing.
programming or erasure in such cases prevents damage to the flash memory due to exces
programming or erasing.

If the microcomputer malfunctions during programming/erasing of the flash memory, the
bit in the FCCS register is set to 1 and the error-protection state is entered, and this aborts
programming or erasure.

The FLER bit is set in the following conditions:

1. When an interrupt such as NMI occurs during programming/erasing.
2. When the flash memory is read during programming/erasing (including a vector read
   instruction fetch).
3. When a SLEEP instruction (including software-standby mode) is executed during
   programming/erasing.

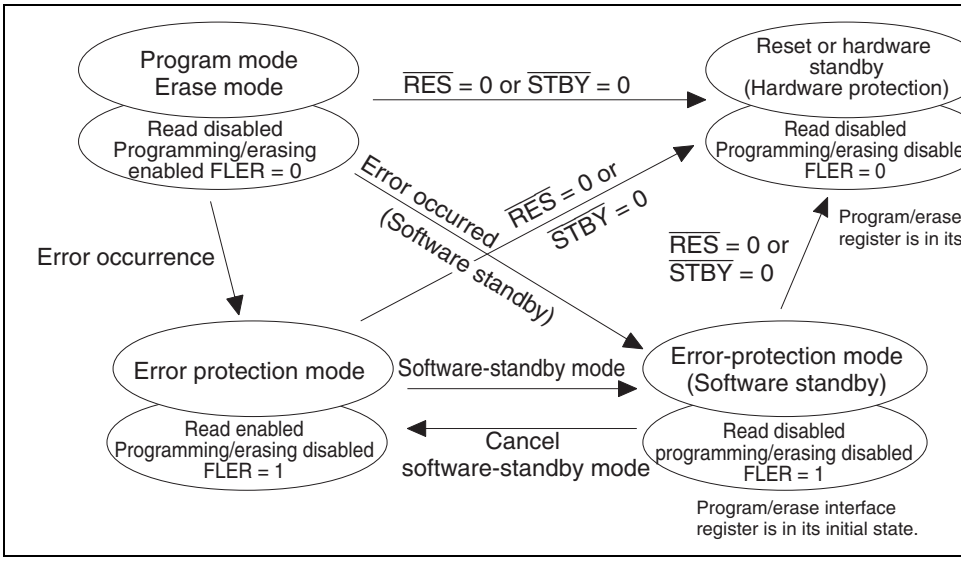


Figure 21.16 Transitions to Error-Protection State

3. If an interrupt has occurred during switching, there is no guarantee of which memory being accessed. Always mask the maskable interrupts before switching between MAT addition, configure the system so that NMI interrupts do not occur during MAT switc
4. After the MATs have been switched, take care because the interrupt vector table will been switched. If interrupt processing is to be the same before and after MAT switchi transfer the interrupt-processing routines to the on-chip RAM and set the WEINTE bi FCCS to place the interrupt-vector table in the on-chip RAM.
5. Memory sizes of the user MAT and user boot MAT are different. When accessing the boot MAT, do not access addresses above the top of its 16-Kbyte memory space. If ac goes beyond the 16-Kbyte space, the values read are undefined.

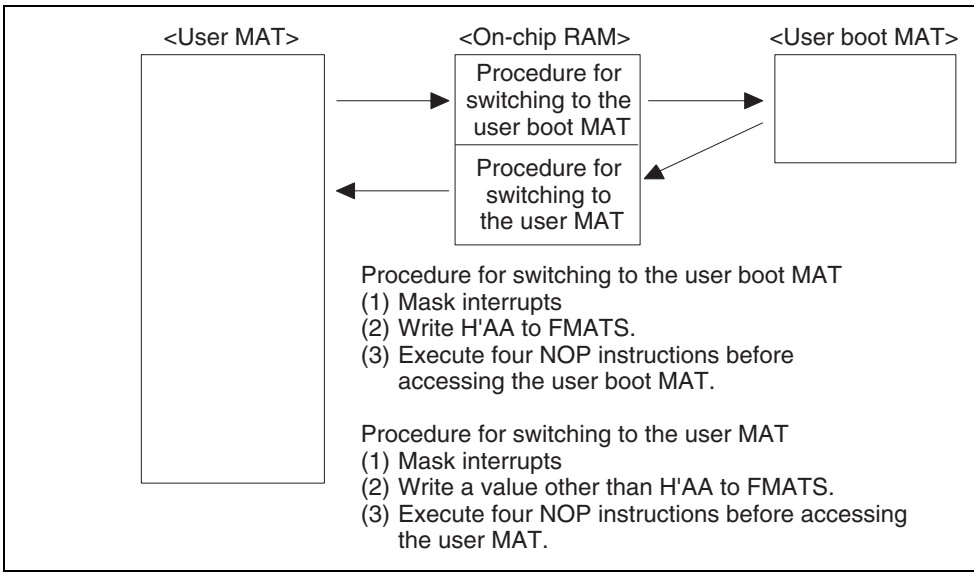


Figure 21.17 Switching between the User MAT and User Boot MAT

MHz input-clock signal.

Notes: 1. For the PROM programmer and the version of its program, see the instructio
for socket adapter.
2. In this LSI, set the programming voltage of the PROM programmer to 3.3 V.

mode enables starting of the boot program and entry to the bit-rate-adjustment state. T
program receives the command from the host to adjust the bit rate. After adjusting the
the program enters the inquiry/selection state.

2. Inquiry/Selection State

   In this state, the boot program responds to inquiry commands from the host. The devi
   clock mode, and bit rate are selected. After selection of these settings, the program is
   enter the programming/erasing state by the command for a transition to the
   programming/erasing state. The program transfers the libraries required for erasure to
   chip RAM and erases the user MATs and user boot MATs before the transition.

3. Programming/erasing state

   Programming and erasure by the boot program take place in this state. The boot progr
   made to transfer the programming/erasing programs to the RAM by commands from
   Sum checks and blank checks are executed by sending these commands from the host

These boot program states are shown in figure 21.18.

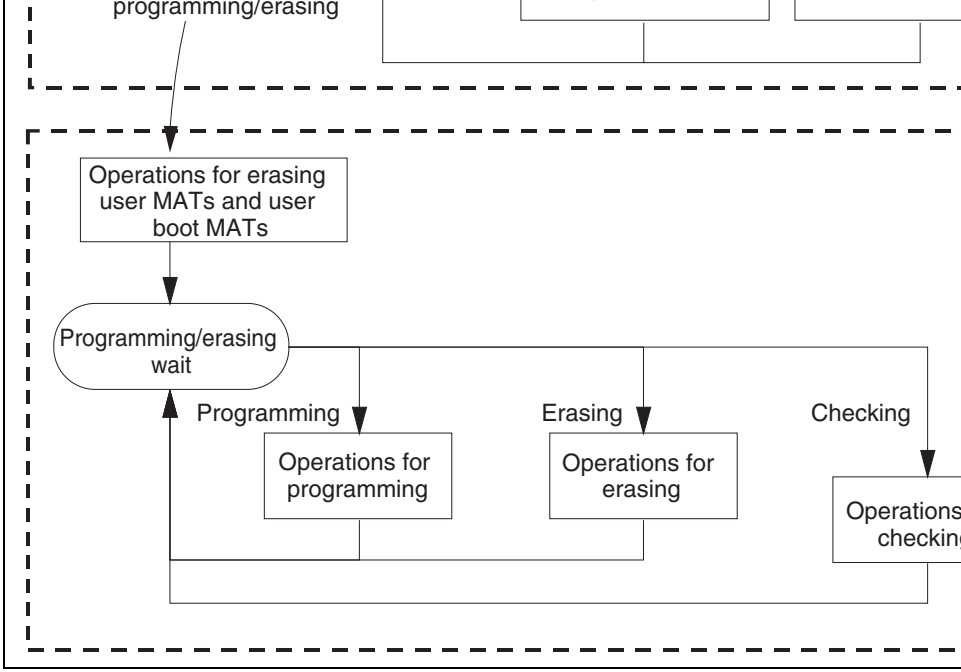


Figure 21.18 Boot Program States

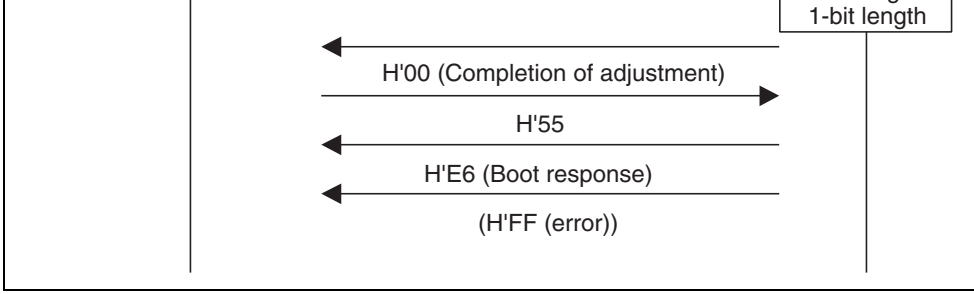


Figure 21.19 Bit-Rate-Adjustment Sequence

(3) Communications Protocol

After adjustment of the bit rate, the protocol for communications between the host and th program is as shown below.

1. 1-byte commands and 1-byte responses

   These commands and responses are comprised of a single byte. These are consists of inquiries and the ACK for successful completion.
2. n-byte commands or n-byte responses

   These commands and responses are comprised of n bytes of data. These are selections responses to inquiries.

   The amount of programming data is not included under this heading because it is dete in another command.
3. Error response

   The error response is a response to inquiries. It consists of an error response and an er and comes two bytes.

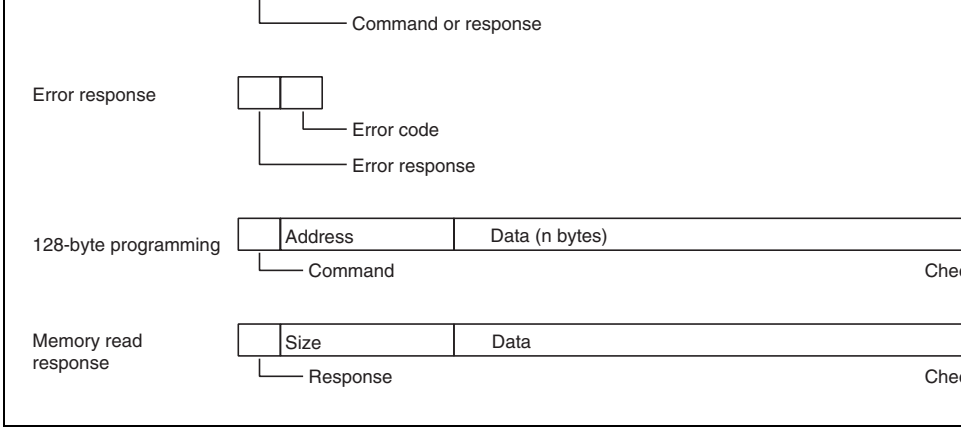


Figure 21.20 Communication Protocol Format

- 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 t command byte to the SUM byte becomes H′00.
- Data (n bytes): Detailed data of a command or response
- Error response (1 byte): Error response to a command
- Error code (1 byte): Type of the error
- Address (4 bytes): Address for programming
- Data (n bytes): Data to be programmed (the size is indicated in the response to the programming unit inquiry.)

Command	Command Name	Description
H'20	Supported Device Inquiry	Inquiry regarding device codes
H'10	Device Selection	Selection of device code
H'21	Clock Mode Inquiry	Inquiry regarding numbers of clock m values of each mode
H'11	Clock Mode Selection	Indication of the selected clock mode
H'22	Multiplication Ratio Inquiry	Inquiry regarding the number of frequ multiplied clock types, the number of multiplication ratios, and the values o multiple
H'23	Operating Clock Frequency Inquiry	Inquiry regarding the maximum and m values of the main clock and peripher
H'24	User Boot MAT Information Inquiry	Inquiry regarding the number of user MATs and the start and last addresse each MAT
H'25	User MAT Information Inquiry	Inquiry regarding the a number of use and the start and last addresses of ea
H'26	Block for Erasing Information Inquiry	Inquiry regarding the number of block the start and last addresses of each b
H'27	Programming Unit Inquiry	Inquiry regarding the unit of programm
H'3F	New Bit Rate Selection	Selection of new bit rate
H'40	Transition to Programming/Erasing State	Erasing of user MAT and user boot M entry to programming/erasing state
H'4F	Boot Program Status Inquiry	Inquiry into the operated status of the program

(a) Supported Device Inquiry

The boot program will return the device codes of supported devices and the product cod
response to the supported device inquiry.

Command	H'20

- Command, H'20, (1 byte): Inquiry regarding supported devices

Response	H'30	Size	Number of devices	
	Number of characters	Device code		Product name
	...			
	SUM			

- Response, H'30, (1 byte): Response to the supported device inquiry
- Size (1 byte): Number of bytes to be transmitted, excluding the command, size, and checksum, that is, the amount of data contributes by the number of devices, characte codes and product names
- Number of devices (1 byte): The number of device types supported by the boot prog
- Number of characters (1 byte): The number of characters in the device codes and bo program's name
- Device code (4 bytes): ASCII code of the supporting product
- Product name (n bytes): Type name of the boot program in ASCII-coded characters
- SUM (1 byte): Checksum

  The checksum is calculated so that the total number of all values from the command the SUM byte becomes H'00.

- SUM (1 byte): Checksum

Response	H'06

- Response, H'06, (1 byte): Response to the device selection command
  ACK will be returned when the device code matches.

Error response	H'90	ERROR

- Error response, H'90, (1 byte): Error response to the device selection command
  ERROR : (1 byte): Error code
  H'11: Sum check error
  H'21: Device code error, that is, the device code does not match

(c) Clock Mode Inquiry

The boot program will return the supported clock modes in response to the clock mode in

Command	H'21

- Command, H'21, (1 byte): Inquiry regarding clock mode

Response	H'31	Size	Number of modes	Mode	···	SUM

- Response, H'31, (1 byte): Response to the clock-mode inquiry
- Size (1 byte): Amount of data that represents the number of modes and modes
- Number of clock modes (1 byte): The number of supported clock modes
  H'00 indicates no clock mode or the device allows to read the clock mode.
- Mode (1 byte): Values of the supported clock modes (i.e. H'01 means clock mode 1.)
- SUM (1 byte): Checksum

- SUM (1 byte): Checksum

Response	H'06

- Response, H'06, (1 byte): Response to the clock mode selection command
  ACK will be returned when the clock mode matches.

Error Response	H'91	ERROR

- Error response, H'91, (1 byte) : Error response to the clock mode selection comma
- ERROR : (1 byte): Error code
  H'11: Checksum error
  H'22: Clock mode error, that is, the clock mode does not match.

Even if the clock mode numbers are H'00 and H'01 by a clock mode inquiry, the clock m
be selected using these respective values.

...						
SUM						

- Response, H'32, (1 byte): Response to the multiplication ratio inquiry
- Size (1 byte): The amount of data that represents the number of clock sources and multiplication ratios and the multiplication ratios
- Number of types (1 byte): The number of supported multiplied clock types (e.g. when there are two multiplied clock types, which are the main and peripheral clo number of types will be H'02.)
- Number of multiplication ratios (1 byte): The number of multiplication ratios for eac (e.g. the number of multiplication ratios to which the main clock can be set and the pe clock can be set.)
- Multiplication ratio (1 byte)

  Multiplication ratio: The value of the multiplication ratio (e.g. when the clock-frequen multiplier is four, the value of multiplication ratio will be H'04.)

  Division ratio: The inverse of the division ratio, i.e. a negative number (e.g. when the divided by two, the value of division ratio will be H'FE. H'FE = D'-2)

  The number of multiplication ratios returned is the same as the number of multiplicati and as many groups of data are returned as there are types.
- SUM (1 byte): Checksum

clock frequency	frequency
...	
SUM	

- Response, H'33, (1 byte): Response to operating clock frequency inquiry
- Size (1 byte): The number of bytes that represents the minimum values, maximum v the number of frequencies.
- Number of operating clock frequencies (1 byte): The number of supported operating frequency types
  (e.g. when there are two operating clock frequency types, which are the main and pe clocks, the number of types will be H'02.)
- Minimum value of operating clock frequency (2 bytes): The minimum value of the or divided clock frequency.
  The minimum and maximum values represent the values in MHz, valid to the hundre of MHz, and multiplied by 100. (e.g. when the value is 20.00 MHz, it will be 2000, v H'07D0.)
- Maximum value (2 bytes): Maximum value among the multiplied or divided clock frequencies.
  There are as many pairs of minimum and maximum values as there are operating clo frequencies.
- SUM (1 byte): Checksum

- Response, H'34, (1 byte): Response to user boot MAT information inquiry
- Size (1 byte): The number of bytes that represents the number of areas, area-start 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 are
- SUM (1 byte): Checksum

(h) User MAT Information Inquiry

The boot program will return the number of user MATs and their addresses.

Command

H'25

- Command, H'25, (1 byte): Inquiry regarding user MAT information

Response

H'35	Size	Number of areas	
Start address area			Last address area
...			
SUM			

- Response, H'35, (1 byte): Response to the user MAT information inquiry
- Size (1 byte): The number of bytes that represents the number of areas, area-start 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

Block start address	Block last address
...	
SUM	

- Response, H'36, (1 byte): Response to the number of erased blocks and addresses
- Size (three bytes): The number of bytes that represents the number of blocks, block-addresses, and block-last addresses.
- Number of blocks (1 byte): The number of erased blocks
- Block start address (4 bytes): Start address of a block
- Block last Address (4 bytes): Last address of a block
  There are as many groups of data representing the start and last addresses as there ar
- SUM (1 byte): Checksum

(j) Programming Unit Inquiry

The boot program will return the programming unit used to program data.

Command	H'27

- Command, H'27, (1 byte): Inquiry regarding programming unit

Response	H'37	Size	Programming unit	SUM

- Response, H'37, (1 byte): Response to programming unit inquiry
- Size (1 byte): The number of bytes that indicate the programming unit, which is fix
- Programming unit (2 bytes): A unit for programming
  This is the unit for reception of programming.
- SUM (1 byte): Checksum

- Size(1 byte): The number of bytes that represents the bit rate, input frequency, numb multiplication ratios, and multiplication ratio
- Bit rate (2 bytes): New bit rate
  One hundredth of the value (e.g. when the value is 19200 bps, it will be 192, which is H'00C0.)
- Input frequency (2 bytes): Frequency of the clock input to the boot program
  This is valid to the hundredths place and represents the value in MHz multiplied by 1 when the value is 20.00 MHz, it will be 2000, which is H'07D0.)
- Number of multiplication ratios (1 byte): The number of multiplication ratios to whic device can be set.
- Multiplication ratio 1 (1 byte) : The value of multiplication or division ratios for the operating frequency
  Multiplication ratio (1 byte): The value of the multiplication ratio (e.g. when the clock 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 c 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 clock frequency is multiplied by four, the multiplication ratio will be H'04.)
  (Division ratio: The inverse of the division ratio, as a negative number (E.g. when the divided by two, the value of division ratio will be H'FE. H'FE = D'-2)
- SUM (1 byte): Checksum

Response | H'06 |

- Response, H'06, (1 byte): Response to selection of a new bit rate
  When it is possible to set the bit rate, the response will be ACK.

H'27: Operating frequency error
The frequency is not within the specified range.

(5) Received Data Check

The methods for checking of received data are listed below.

1. Input frequency

The received value of the input frequency is checked to ensure that it is within the range minimum to maximum frequencies which matches the clock modes of the specified dev the value is out of this range, an input-frequency error is generated.

2. Multiplication ratio

The received value of the multiplication ratio or division ratio is checked to ensure that i the clock modes of the specified device. When the value is out of this range, an multipli ratio error is generated.

3. Operating frequency

Operating frequency is calculated from the received value of the input frequency and the multiplication or division ratio. The input frequency is input to the LSI and the LSI is op the operating frequency. The expression is given below.

Operating frequency = Input frequency × Multiplication ratio, or
Operating frequency = Input frequency ÷ Division ratio

The calculated operating frequency should be checked to ensure that it is within the rang minimum to maximum frequencies which are available with the clock modes of the spec device. When it is out of this range, an operating frequency error is generated.

response. The host will send an ACK with the new bit rate for confirmation and the boot will response with that rate.

Confirmation	H'06

- Confirmation, H'06, (1 byte): Confirmation of a new bit rate

Response	H'06

- Response, H'06, (1 byte): Response to confirmation of a new bit rate

The sequence of new bit-rate selection is shown in figure 21.21.

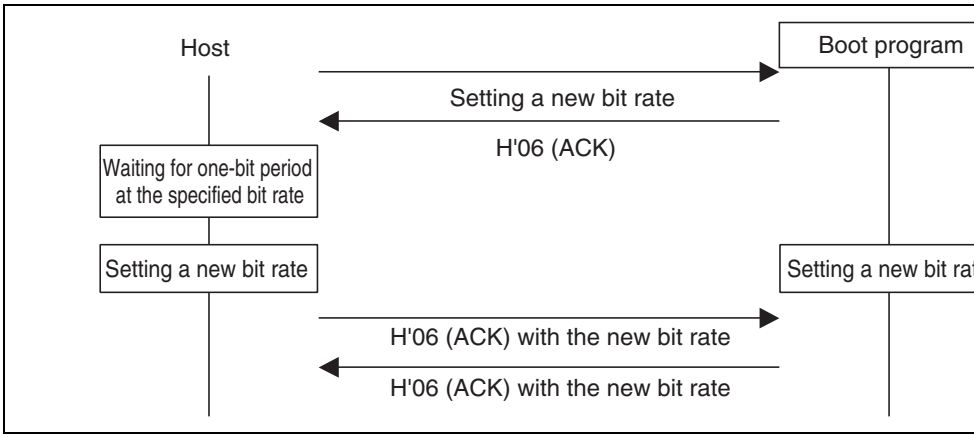


Figure 21.21 New Bit-Rate Selection Sequence

Command	H'40

- Command, H'40, (1 byte): Transition to programming/erasing state

Response	H'06

- Response, H'06, (1 byte): Response to transition to programming/erasing state
  The boot program will send ACK when the user MAT and user boot MAT have beer
  by the transferred erasing program.

Error Response	H'C0	H'51

- Error response, H'C0, (1 byte): Error response for user boot MAT blank check
- Error code, H'51, (1 byte): Erasing error
  An error occurred and erasure was not completed.

(7) Command Error

A command error will occur when a command is undefined, the order of commands is i
or a command is unacceptable. Issuing a clock-mode selection command before a devic
or an inquiry command after the transition to programming/erasing state command, are

Error Response	H'80	H'XX

- Error response, H'80, (1 byte): Command error
- Command, H'XX, (1 byte): Received command

be made, such as the multiplication-ratio inquiry (H'22) or operating frequency inquir which are needed for a new bit-rate selection.

6. A new bit rate should be selected with the new bit-rate selection (H'3F) command, acc to the returned information on multiplication ratios and operating frequencies.
7. After selection of the device and clock mode, the information of the user boot MAT a MAT should be made to inquire about the user boot MATs information inquiry (H'24 MATs information inquiry (H'25), erased block information inquiry (H'26), and progr unit inquiry (H'27).
8. After making inquiries and selecting a new bit rate, issue the transition to programming/erasing state command (H'40). The boot program will then enter the programming/erasing state.

H'43	User MAT programming selection	Transfers the user MAT programm program
H'50	128-byte programming	Programs 128 bytes of data
H'48	Erasing selection	Transfers the erasing program
H'58	Block erasing	Erases a block of data
H'52	Memory read	Reads the contents of memory
H'4A	User boot MAT sum check	Checks the checksum of the user
H'4B	User MAT sum check	Checks the checksum of the user
H'4C	User boot MAT blank check	Checks whether the contents of th boot MAT are blank
H'4D	User MAT blank check	Checks whether the contents of th MAT are blank
H'4F	Boot program status inquiry	Inquires into the boot program's st

programming command. The 128-byte programming command that follows the select
command represents the data programmed according to the method specified by the se
command. When more than 128-byte data is programmed, 128-byte commands should
repeatedly be executed. Sending a 128-byte programming command with H'FFFFFFF
address will stop the programming. On completion of programming, the boot program
wait for selection of programming or erasing.

Where the sequence of programming operations that is executed includes programmin
another method or of another MAT, the procedure must be repeated from the program
selection command.

The sequence for programming-selection and 128-byte programming commands is sh
figure 21.22.

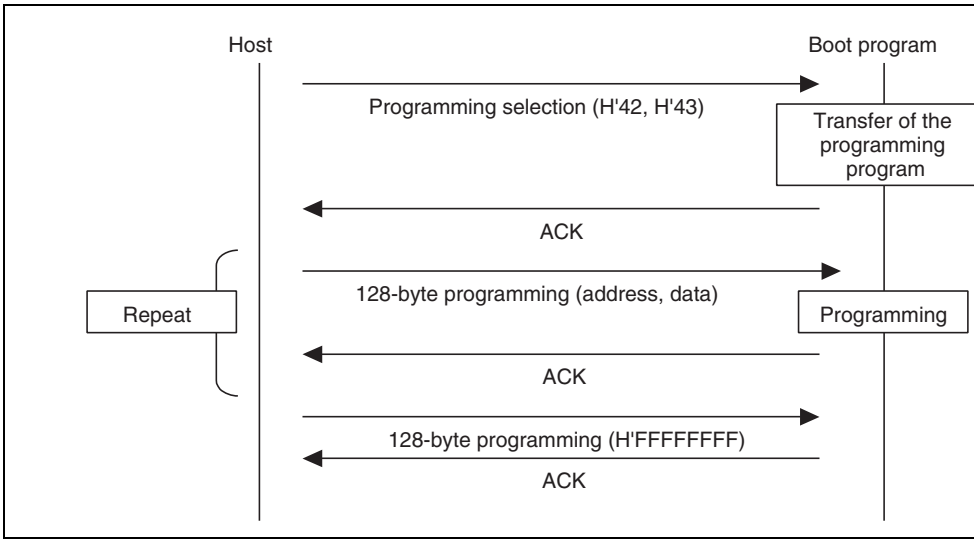


Figure 21.22 Programming Sequence

Error Response	H'C2	ERROR

- Error response : H'C2 (1 byte): Error response to user boot MAT programming sele
- ERROR : (1 byte): Error code

  H'54: Selection processing error (transfer error occurs and processing is not complet

- User MAT programming selection

  The boot program will transfer a program for programming. The data is programmed user MATs by the transferred program for programming.

Command	H'43

- Command, H'43, (1 byte): User MAT programming selection

Response	H'06

- Response, H'06, (1 byte): Response to user MAT programming selection

  When the programming program has been transferred, the boot program will return A

Error Response	H'C3	ERROR

- Error response : H'C3 (1 byte): Error response to user MAT programming selection
- ERROR : (1 byte): Error code

  H'54: Selection processing error (transfer error occurs and processing is not complet

(b) 128-byte programming

The boot program will use the programming program transferred by the programming se program the user boot MATs or user MATs in response to 128-byte programming.

Command	H'50	Address						
	Data	...						
	...							
	SUM							

On completion of programming, the boot program will return ACK.

Error Response	H'D0	ERROR

- Error response, H'D0, (1 byte): Error response for 128-byte programming
- ERROR: (1 byte): Error code

  H'11: Checksum Error

  H'2A: Address Error

  H'53: Programming error

  A programming error has occurred and programming cannot be 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.

Firstly, erasure is selected by the erasure selection command and the boot program then
specified block. The command should be repeatedly executed if two or more blocks are
erased. Sending a block-erasure command from the host with the block number H'FF wi
erasure operating. On completion of erasing, the boot program will wait for selection of
programming or erasing.

The sequences of issuing the erasure selection command and block-erasure command ar
figure 21.23.

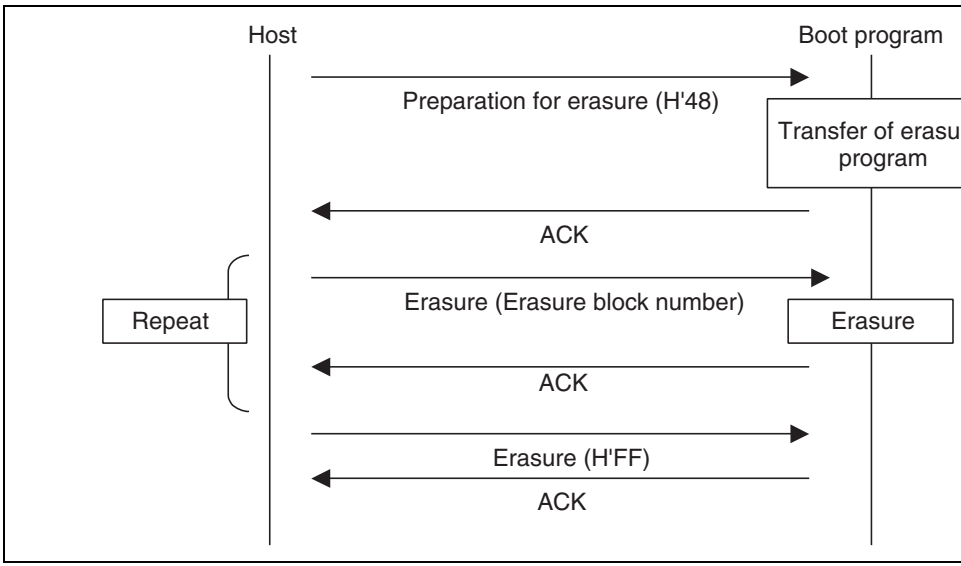


Figure 21.23 Erasure Sequence

Error Response	H'C8	ERROR

- Error Response, H'C8, (1 byte): Error response to erasure selection
- ERROR: (1 byte): Error code

  H'54: Selection processing error (transfer error occurs and processing is not complet

(b) Block Erasure

The boot program will erase the contents of the specified block.

Command	H'58	Size	Block number	SUM

- Command, H'58, (1 byte): Erasure
- Size (1 byte): The number of bytes that represents the erasure block number
  This is fixed to 1.
- Block number (1 byte): Number of the block to be erased
- SUM (1 byte): Checksum

Response	H'06

- Response, H'06, (1 byte): Response to Erasure
  After erasure has been completed, the boot program will return ACK.

Error Response	H'D8	ERROR

- Error Response, H'D8, (1 byte): Response to Erasure
- ERROR (1 byte): Error code

  H'11: Sum check error
  H'29: Block number error
  Block number is incorrect.
  H'51: Erasure error
  An error has occurred during erasure.

Response	H'06

- Response, H'06, (1 byte): Response to end of erasure (ACK)
  When erasure is to be performed after the block number H'FF has been sent, the proc
  should be executed from the erasure selection command.

(11) Memory read

The boot program will return the data in the specified address.

Command	H'52	Size	Area	Read address	
	Read size				SUM

- Command: H'52 (1 byte): Memory read
- Size (1 byte): Amount of data that represents the area, read address, and read size (f
- Area (1 byte)
  H'00: User boot MAT
  H'01: User MAT
  An address error occurs when the area setting is incorrect.
- Read address (4 bytes): Start address to be read from
- Read size (4 bytes): Size of data to be read
- SUM (1 byte): Checksum

Response	H'52	Read size						
	Data	···						
	SUM							

- Response: H'52 (1 byte): Response to memory read
- Read size (4 bytes): Size of data to be read
- Data (n bytes): Data for the read size from the read address
- SUM (1 byte): Checksum

The boot program will return the byte-by-byte total of the contents of the bytes of the use MAT, as a 4-byte value.

Command	H'4A

- Command, H'4A, (1 byte): Sum check for user-boot MAT

Response	H'5A	Size	Checksum of user boot program	SUM

- Response, H'5A, (1 byte): Response to the sum check of user-boot MAT
- Size (1 byte): The number of bytes that represents the checksum
  This is fixed to 4.
- Checksum of user boot program (4 bytes): Checksum of user boot MATs
  The total of the data is obtained in byte units.
- SUM (1 byte): Sum check for data being transmitted

(13) User MAT Sum Check

The boot program will return the byte-by-byte total of the contents of the bytes of the use

Command	H'4B

- Command, H'4B, (1 byte): Sum check for user MAT

Response	H'5B	Size	Checksum of user program	SUM

- Response, H'5B, (1 byte): Response to the sum check of the user MAT
- Size (1 byte): The number of bytes that represents the checksum
  This is fixed to 4.
- Checksum of user boot program (4 bytes): Checksum of user MATs
  The total of the data is obtained in byte units.
- SUM (1 byte): Sum check for data being transmitted

Error Response	H'CC	H'52

- Error Response, H'CC, (1 byte): Response to blank check for user boot MAT
- Error Code, H'52, (1 byte): Erasure has not been completed.

(15) User MAT Blank Check

The boot program will check whether or not all user MATs are blank and return the resu

Command	H'4D

- Command, H'4D, (1 byte): Blank check for user MATs

Response	H'06

- Response, H'06, (1 byte): Response to the blank check for user boot MATs
  If the contents of all user MATs are blank (H'FF), the boot program will return ACK

Error Response	H'CD	H'52

- Error Response, H'CD, (1 byte): Error response to the blank check of user MATs.
- Error code, H'52, (1 byte): Erasure has not been completed.

- Status (1 byte): State of the boot program
- ERROR (1 byte): Error status

  ERROR = 0 indicates normal operation.
  ERROR = 1 indicates error has occurred.

- SUM (1 byte): Sum check

Table 21.13 Status Code

Code	Description
H'11	Device Selection Wait
H'12	Clock Mode Selection Wait
H'13	Bit Rate Selection Wait
H'1F	Programming/Erasing State Transition Wait (Bit rate selection is completed)
H'31	Programming State for Erasure
H'3F	Programming/Erasing Selection Wait (Erasure is completed)
H'4F	Programming Data Receive Wait (Programming is completed)
H'5F	Erasure Block Specification Wait (Erasure is completed)

H'26	Multiplication Ratio Error
H'27	Operating Frequency Error
H'29	Block Number Error
H'2A	Address Error
H'2B	Data Length Error
H'51	Erasure Error
H'52	Erasure Incomplete Error
H'53	Programming Error
H'54	Selection Processing Error
H'80	Command Error
H'FF	Bit-Rate-Adjustment Confirmation Error

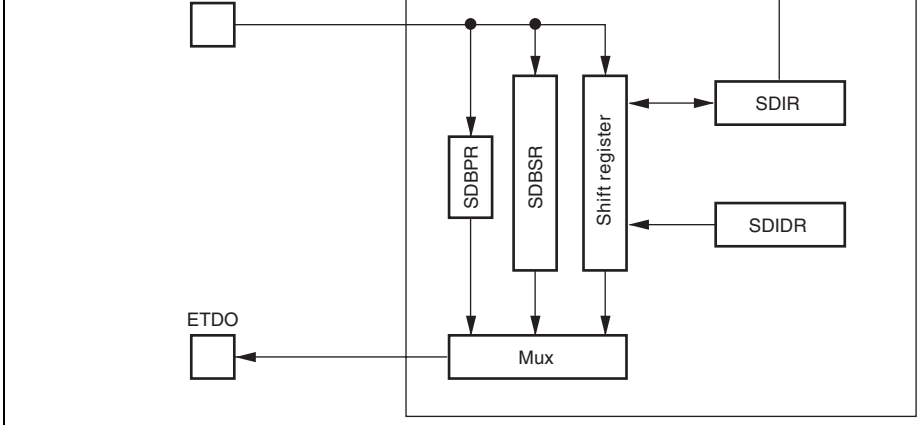
PROM programmer that supports the 512-Kbyte flash memory on-chip MCU device a
Do not set the programmer to HN28F101 or the programming voltage to 5.0 V. Use o
specified socket adapter. If other adapters are used, the product may be damaged.

5. Do not remove the chip from the PROM programmer nor input a reset signal during programming/erasing. As a high voltage is applied to the flash memory during programming/erasing, doing so may damage or destroy flash memory permanently. If executed accidentally, reset must be released after the reset input period of 100 μs wh longer than normal.
6. The flash memory is not accessible until FKEY is cleared after programming/erasing completes. If this LSI is restarted by a reset immediately after programming/erasing h finished, secure the reset period (period of $\overline{RES}$ = 0) of more than 100 μs. Though tra the reset state or hardware standby state during programming/erasing is prohibited, if executed accidentally, reset must be released after the reset input period of 100 μs wh longer than normal.
7. At powering on or off the Vcc power supply, fix the $\overline{RES}$ pin to low and set the flash to hardware protection state. This power on/off timing must also be satisfied at a pow power-on caused by a power failure and other factors.
8. Program the area with 128-byte programming-unit blocks in on-board programming o programmer mode only once. Perform programming in the state where the programm block is fully erased.
9. When the chip is to be reprogrammed with the programmer after execution of progra erasure in on-board programming mode, it is recommend that automatic programming performed after execution of automatic erasure.
10. To write data or programs to the flash memory, data or programs must be allocated to addresses higher than that of the external interrupt vector table (H'000040) and H'FF r written to the areas that are reserved for the system in the exception handling vector ta

13. While an instruction in on-chip RAM is being executed, the DTC can write to the SC FCCS that is used for a download request or FMATS that is used for MAT switching sure that these registers are not accidentally written to, otherwise an on-chip program downloaded and damage RAM or a MAT switchover may occur and the CPU get ou control. Do not use DTC to program flash related registers.
14. A programming/erasing program for flash memory used in the conventional H8S F-Z microcomputer which does not support download of the on-chip program by a SCO request cannot run in this LSI. Be sure to download the on-chip program to execute programming/erasing of flash memory in this LSI.
15. Unlike the conventional H8S F-ZTAT microcomputer, no countermeasures are avail runaway by WDT during programming/erasing. Prepare countermeasures (e.g. use o periodic timer interrupts) for WDT with taking the programming/erasing time into consideration as required.

22.1 Features

- Five test pins (ETCK, ETDI, ETDO, ETMS, and $\overline{\text{ETRST}}$)
- TAP controller
- Six instructions
  BYPASS mode
  EXTEST mode
  SAMPLE/PRELOAD mode
  CLAMP mode
  HIGHZ mode
  IDCODE mode
  (These instructions are test modes corresponding to IEEE 1149.1.)



[Legend]
SDIR: Instruction register
SDBPR: Bypass register
SDBSR: Boundary scan register
SDIDR: ID code register

Figure 22.1 JTAG Block Diagram

			with a duty cycle close to 50% should b For details, see section 26, Electrical 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.

input pin (ETDI). Data from SDIR can be output via the test data output pin (ETDO). The register (SDBPR) is a 1-bit register to which the ETDI and ETDO pins are connected in B CLAMP, or HIGHZ mode. The boundary scan register (SDBSR) is a 337-bit register to w 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 22.2 shows the kinds of serial transfer possible with each JTAG register.

Table 22.2 JTAG Register Serial Transfer

Register	Serial Input	Serial Output
SDIR	Possible	Possible
SDBPR	Possible	Possible
SDBSR	Possible	Possible
SDIDR	Impossible	Possible

31	TS3	1	R/W	Test Set Bits
30	TS2	1	R/W	0000: EXTEST mode
29	TS1	1	R/W	0001: Setting prohibited
28	TS0	0	R/W	0010: CLAMP mode
				0011: HIGHZ mode
				0100: SAMPLE/PRELOAD mode
				0101: Setting prohibited
				: :
				1101: Setting prohibited
				1110: IDCODE mode (Initial value)
				1111: BYPASS mode
27 to 14	—	All 0	R	Reserved These bits are always read as 0 and cannot be m
13	—	1	R	Reserved This bit is always read as 1 and cannot be modi
12	—	0	R	Reserved This bit is always read as 0 and cannot be modi
11	—	1	R	Reserved This bit is always read as 1 and cannot be modi
10 to 1	—	All 0	R	Reserved These bits are always read as 0 and cannot be m
0	—	1	R	Reserved This bit is always read as 1 and cannot be modi

Table 22.3 shows the relationship between the pins of this LSI and the boundary scan reg

		Enable	335			—
		Output	334			—
3	P46	Input	333	13	VCL	—
		Enable	332			—
		Output	331			—
4	P47	Input	330	14	$\overline{\text{MD2}}$	Input
		Enable	329			—
		Output	328			—
5	P56	Input	327	15	P51	Input
		Enable	326			Enable
		Output	325			Output
6	P57	Input	324	16	P50	Input
		Enable	323			Enable
		Output	322			Output
7	VSS	—	—	17	P97	Input
		—	—			Enable
		—	—			Output
8	$\overline{\text{RES}}$	—	—	18	P96	Input
		—	—			Enable
		—	—			Output
9	MD1	Input	321	19	P95	Input
		—	—			Enable
		—	—			Output

		Output	294			Output	2
23	P91	Input	293	33	PA7	Input	2
		Enable	292			Enable	2
		Output	291			Output	2
24	P90	Input	290	34	PA6	Input	2
		Enable	289			Enable	2
		Output	288			Output	2
25	PC7	Input	287	35	PA5	Input	2
		Enable	286			Enable	2
		Output	285			Output	2
26	PC6	Input	284	36	VCC	—	–
		Enable	283			—	–
		Output	282			—	–
27	PC5	Input	281	37	PA4	Input	2
		Enable	280			Enable	2
		Output	279			Output	2
28	PC4	Input	278	38	PA3	Input	2
		Enable	277			Enable	2
		Output	276			Output	2
29	PC3	Input	275	39	PA2	Input	2
		Enable	274			Enable	2
		Output	273			Output	2

		—	—				Output
43	P87	Input	239	53	PE5	Input	
		Enable	238			Enable	
		Output	237			Output	
44	P86	Input	236	54	PE4	Input	
		Enable	235			Enable	
		Output	234			Output	
45	P85	Input	233	55	PE3	Input	
		Enable	232			Enable	
		Output	231			Output	
46	P84	Input	230	56	PE2	Input	
		Enable	229			Enable	
		Output	228			Output	
47	P83	Input	227	57	PE1	Input	
		Enable	226			Enable	
		Output	225			Output	
48	P82	Input	224	58	PE0	Input	
		Enable	223			Enable	
		Output	222			Output	
49	P81	Input	221	59	PD7	Input	
		Enable	220			Enable	
		Output	219			Output	

		Output	180			—	–
63	PD3	Input	179	73	P75	Input	1
		Enable	178			—	–
		Output	177			—	–
64	PD2	Input	176	74	P76	Input	1
		Enable	175			—	–
		Output	174			—	–
65	PD1	Input	173	75	P77	Input	1
		Enable	172			—	–
		Output	171			—	–
66	PD0	Input	170	76	AVCC	—	–
		Enable	169			—	–
		Output	168			—	–
67	AVSS	—	—	77	AVref	—	–
		—	—			—	–
		—	—			—	–
68	P70	Input	167	78	P60	Input	1
		—	—			Enable	1
		—	—			Output	1
69	P71	Input	166	79	P61	Input	1
		—	—			Enable	1
		—	—			Output	1

		Output	145			Output	
83	P65	Input	144	93	PF1	Input	
		Enable	143			Enable	
		Output	142			Output	
84	P66	Input	141	94	PF0	Input	
		Enable	140			Enable	
		Output	139			Output	
85	P67	Input	138	95	VSS	—	
		Enable	137			—	
		Output	136			—	
86	VCC	—	—	96	P27	Input	
		—	—			Enable	
		—	—			Output	
87	ETMS	—	—	97	P26	Input	
		—	—			Enable	
		—	—			Output	
88	ETDO	—	—	98	P25	Input	
		—	—			Enable	
		—	—			Output	
89	ETDI	—	—	99	P24	Input	
		—	—			Enable	
		—	—			Output	

		Output	106			Output	7
103	P20	Input	105	113	PB7	Input	7
		Enable	104			Enable	7
		Output	103			Output	7
104	P17	Input	102	114	PB6	Input	7
		Enable	101			Enable	7
		Output	100			Output	7
105	P16	Input	99	115	PB5	Input	7
		Enable	98			Enable	7
		Output	97			Output	7
106	P15	Input	96	116	PB4	Input	6
		Enable	95			Enable	6
		Output	94			Output	6
107	P14	Input	93	117	PB3	Input	6
		Enable	92			Enable	6
		Output	91			Output	6
108	P13	Input	90	118	PB2	Input	6
		Enable	89			Enable	6
		Output	88			Output	6
109	P12	Input	87	119	PB1	Input	6
		Enable	86			Enable	5
		Output	85			Output	5

		Output	49			Output	
123	P32	Input	48	133	P52	Input	
		Enable	47			Enable	
		Output	46			Output	
124	P33	Input	45	134	P53	Input	
		Enable	44			Enable	
		Output	43			Output	
125	P34	Input	42	135	FWE	Input	
		Enable	41			—	
		Output	40			—	
126	P35	Input	39	136	P54	Input	
		Enable	38			Enable	
		Output	37			Output	
127	P36	Input	36	137	P55	Input	
		Enable	35			Enable	
		Output	34			Output	
128	P37	Input	33	138	P44	Input	
		Enable	32			Enable	
		Output	31			Output	
129	P40	Input	30	139	VSS	—	
		Enable	29			—	
		Output	28			—	

		—	—
143	XTAL	—	—
		—	—
		—	—
144	EXTAL	—	—
		—	—
		—	—
		to ETDO	

22.3.4 ID Code Register (SDIDR)

SDIDR is a 32-bit register. In IDCODE mode, SDIDR can output a fixed code, H'080394 the ETDO pin. However, no serial data can be written to SDIDR via the ETDI pin.

31 28	27 12	11 1	
0000	1000 0000 0011 1001	0100 0100 011	
Version (4 bits)	Part Number (16 bits)	Manufacture Identify (11 bits)	Fix

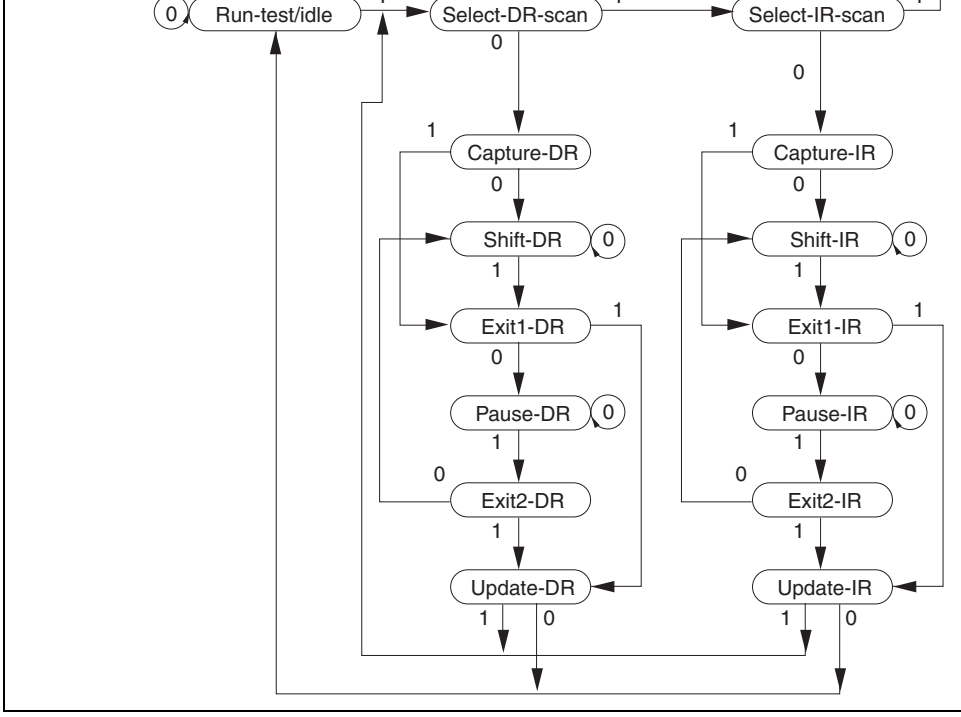


Figure 22.2 TAP Controller State Transitions

by setting a command in SDIR.

22.5.1 Supported Instructions

This LSI supports the three essential instructions defined in the IEEE1149.1 standard (BY
SAMPLE/PRELOAD, and EXTEST) and optional instructions (CLAMP, HIGHZ, and II

(1) BYPASS (Instruction code: B'1111)

The BYPASS instruction is an instruction that operates the bypass register. This instructi
shortens the shift path to speed up serial data transfer involving other chips on the printed
board. While this instruction is being executed, the test circuit has no effect on the syster
circuits.

(2) SAMPLE/PRELOAD (Instruction code: B'0100)

The SAMPLE/PRELOAD instruction inputs values from this LSI internal circuitry to the
boundary scan register, outputs values from the scan path, and loads data onto the scan pa
When this instruction is being executed, this LSI's input pin signals are transmitted direct
internal circuitry, and internal circuit values are directly output externally from the output
This LSI system circuits are not affected by execution of this instruction.

In a SAMPLE operation, a snapshot of a value to be transferred from an input pin to the i
circuitry, or a value to be transferred from the internal circuitry to an output pin, is latche
boundary scan register and read from the scan path. Snapshot latching does not affect nor
operation of this LSI.

In a PRELOAD operation, an initial value is set in the parallel output latch of the bounda
register from the scan path prior to the EXTEST instruction. Without a PRELOAD opera
when the EXTEST instruction was executed an undefined value would be output from th

Data loaded into the output pin boundary scan register in the Capture-DR state is not use
external circuit testing (it is replaced by a shift operation).

(4) CLAMP (Instruction code: B'0010)

When the CLAMP instruction is enabled, the output pin outputs the value of the bounda
register that has been previously set by the SAMPLE/PRELOAD instruction. While the
instruction is enabled, the state of the boundary scan register maintains the previous stat
regardless of the state of the TAP controller.

A bypass register is connected between the ETDI and ETDO pins. The related circuit op
the same way when the BYPASS instruction is enabled.

(5) HIGHZ (Instruction code: B'0011)

When the HIGHZ instruction is enabled, all output pins enter a high-impedance state. W
HIGHZ instruction is enabled, the state of the boundary scan register maintains the prev
regardless of the state of the TAP controller.

A bypass register is connected between the ETDI and ETDO pins. The related circuit op
the same way when the BYPASS instruction is enabled.

2. Boundary scan mode does not cover clock-related pins (EXTAL, XTAL).
3. Boundary scan mode does not cover reset- and standby-related pins ($\overline{\text{RES}}$, $\overline{\text{ST}}$
$\overline{\text{RESO}}$).
4. Boundary scan mode does not cover JTAG-related pins (ETCK, ETDI, ETDO
and $\overline{\text{ETRST}}$).
5. Fix the $\overline{\text{MD2}}$ pin high.
6. Use the $\overline{\text{STBY}}$ pin in high state.

— To prevent the LSI system operation from being affected by the ETRST pin of th
tester, circuits must be separated .

— Alternatively, to prevent the $\overline{\text{ETRST}}$ pin of the board tester from being affected b
system reset, circuits must be separated.

Figure 22.3 shows a design example of the reset signal circuit wherein no reset signa
interference occurs.

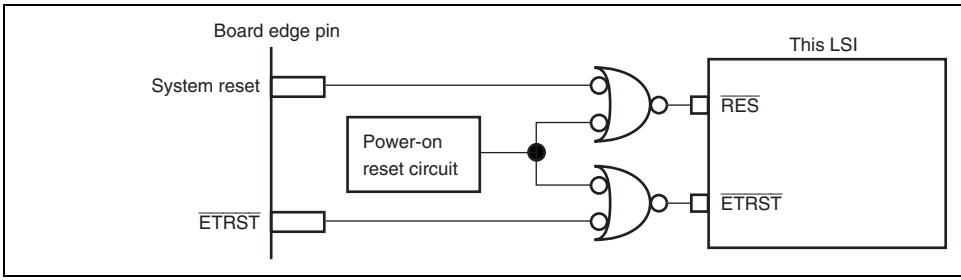


Figure 22.3 Reset Signal Circuit Without Reset Signal Interference

should then be retried, regardless of the transfer operation.

8. If a pin with a pull-up function is sampled while its pull-up function is enabled, 1 can
detected at the corresponding input scan register. In this case, the corresponding enab
register should be cleared to 0.
9. If a pin with an open-drain function is sampled while its open-drain function is enable
corresponding output scan register is 1, 0 can be detected at the corresponding enable
register.

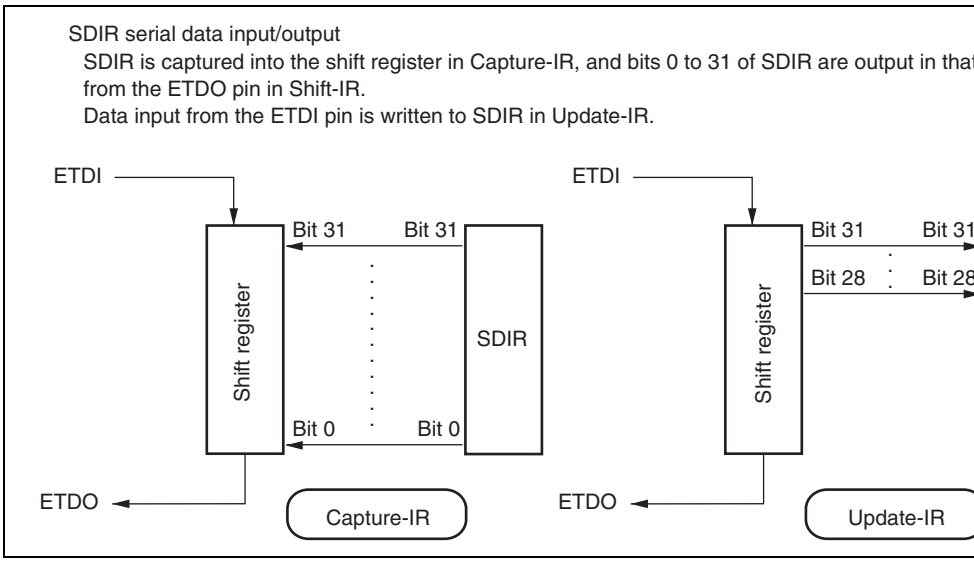


Figure 22.4 Serial Data Input/Output (1)

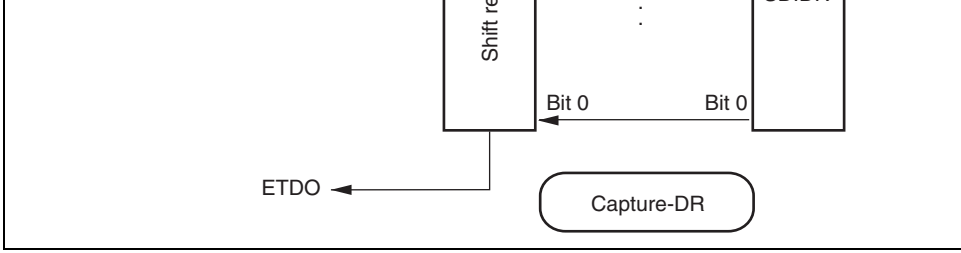


Figure 22.5 Serial Data Input/Output (2)

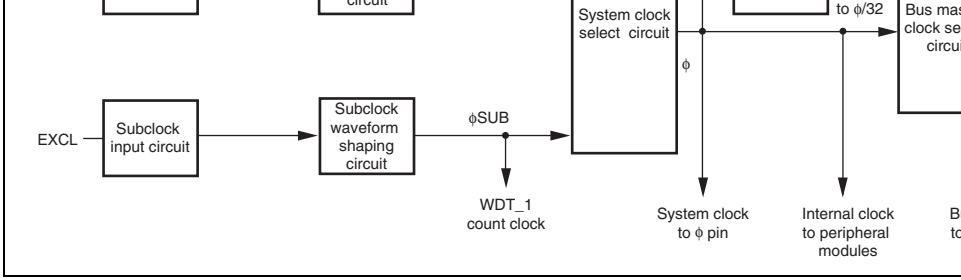


Figure 23.1 Block Diagram of Clock Pulse Generator

The bus master clock is selected as either high-speed mode or medium-speed mode by s
according to the settings of the SCK2 to SCK0 bits in the standby control register. Use o
medium-speed clock (ϕ/2 to ϕ/32) may be limited during CPU operation and when acces
internal memory of the CPU. The operation speed of the DTC and the external space acc
are thus stabilized regardless of the setting of medium-speed mode. For details on the sta
control register, see section 24.1.1, 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 24.1.2, Low-P
Control Register (LPWRCR).

Figure 23.3 shows the equivalent circuit of a crystal resonator. A crystal resonator having characteristics given in table 23.2 should be used.

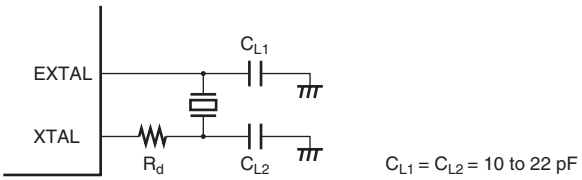


Figure 23.2 Typical Connection to Crystal Resonator

Table 23.1 Damping Resistance Values

Frequency (MHz)	5	8	8.5
R_d (Ω)	300	200	0

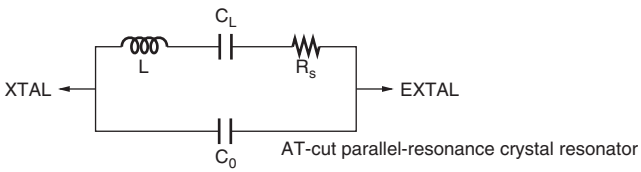


Figure 23.3 Equivalent Circuit of Crystal Resonator

To input an inverted clock to the XTAL pin, the external clock should be tied to high in
mode.

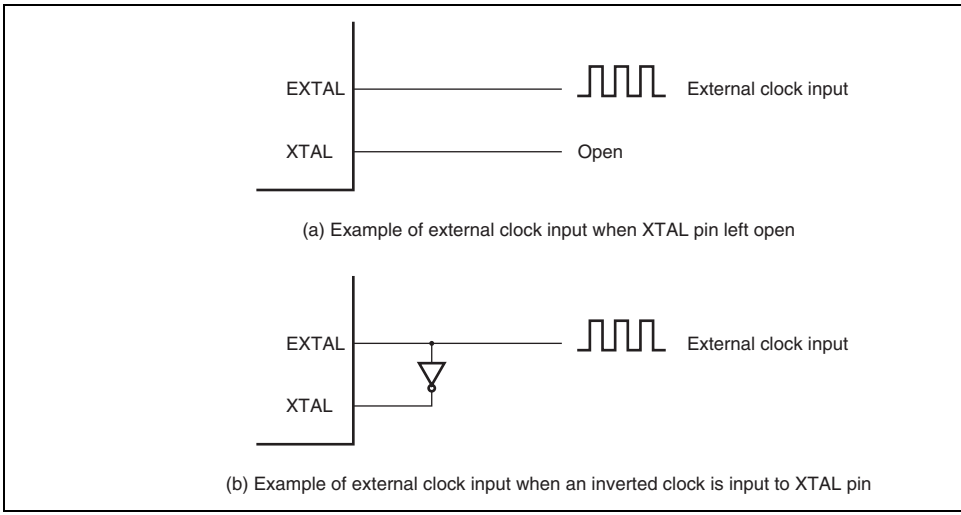


Figure 23.4 Example of External Clock Input

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 26.5 ar
26.8 in section 26, Electrical Characteristics.

23.3 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.

23.4 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.

23.5 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.

23.6 Subclock Waveform Shaping Circuit

To remove noise from the subclock input at the EXCL pin, the subclock is sampled by a clock. The sampling frequency is set by the NESEL bit in LPWRCR.

23.8.2 Notes on Board Design

When using a crystal resonator, the crystal resonator and its load capacitors should be pla
close as possible to the EXTAL and XTAL pins. Other signal lines should be routed awa
the oscillation circuit to prevent inductive interference with the correct oscillation as sho
figure 23.5.

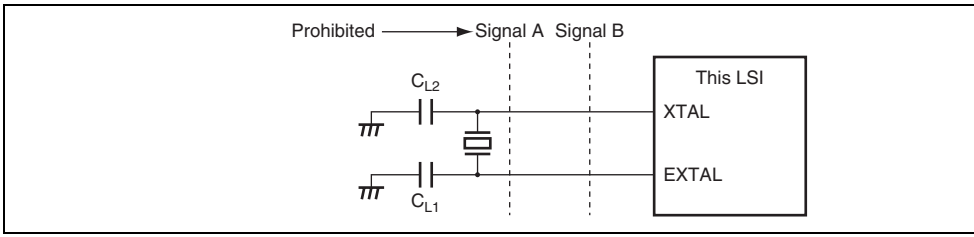


Figure 23.5 Note on Board Design of Oscillation Circuit Section

23.8.3 Note on Operation Check

This LSI may oscillate at several kHz of frequency even when a crystal resonator is not c
to the EXTAL and XTAL pins or an external clock is not input. Use this LSI after confir
the LSI operates with appropriate frequency.

The CPU stops but on-chip peripheral modules continue operating.

- Software standby mode
  Clock oscillation stops, and the CPU and on-chip peripheral modules stop operating.
- Hardware standby mode
  Clock oscillation stops, and the CPU and on-chip peripheral modules enter reset stat
- Module stop mode
  Independently of above operating modes, on-chip peripheral modules that are not us
  stopped individually.

- Module stop control register A (MSTPCRA)
- Sub-chip module stop control register BH, BL (SUBMSTPBH, SUBMSTPBL)

24.1.1 Standby Control Register (SBYCR)

SBYCR controls power-down modes.

Bit	Bit Name	Initial Value	R/W	Description
7	SSBY	0	R/W	Software Standby Specifies the operating mode to be entered after executing the SLEEP instruction. When the SLEEP instruction is executed in high-s mode or medium-speed mode: 0: Shifts to sleep mode 1: Shifts to software standby mode Note that the SSBY bit is not changed even if a m transition occurs by an interrupt.

				STS0 values and wait time.
3	DTSPEED	0	R/W	DTC Speed Specifies the operating clock for the bus masters other than the CPU in medium-speed mode. 0: All bus masters operate based on the mediu clock. 1: The DTC operates based on the system cloc The operating clock is changed when a DTC tra requested even if the CPU operates based on th medium-speed clock.
2	SCK2	0	R/W	System Clock Select 2 to 0 Select a clock for the bus master in high-speed medium-speed mode. 000: High-speed mode (Initial value) 001: Medium-speed clock: ϕ/2 010: Medium-speed clock: ϕ/4 011: Medium-speed clock: ϕ/8 100: Medium-speed clock: ϕ/16 101: Medium-speed clock: ϕ/32 11X: Must not be set.
1	SCK1	0	R/W	
0	SCK0	0	R/W	

[Legend]

X: Don't care

Recommended specification

Note: * Setting prohibited.

[Legend] X: Don't care

24.1.2 Low-Power Control Register (LPWRCR)

LPWRCR controls power-down modes.

Bit	Bit Name	Initial Value	R/W	Description
7, 6	—	0	R/W	Reserved The initial value should not be changed.
5	NESEL	0	R/W	Noise Elimination Sampling Frequency Select Selects the frequency by which the subclock (φSU from the EXCL pin is sampled using the clock (φ) generated by the system clock pulse generator. 0: Sampling using φ/32 clock 1: Sampling using φ/4 clock
4	EXCLE	0	R/W	Subclock Input Enable Enables/disables subclock input from the EXCL p 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.

				0: Outputs AH 1: Outputs AH
0	—	0	R/W	Reserved The initial value should not be changed.

5	MSTP13	1	R/W	16-bit free-running timer (FRT)
4	MSTP12	1	R/W	8-bit timers (TMR_0, TMR_1)
3	MSTP11	1	R/W	14-bit PWM timer (PWMX)
2	MSTP10	1	R/W	Reserved The initial value should not be changed.
1	MSTP9	1	R/W	A/D converter
0	MSTP8	1	R/W	8-bit timers (TMR_X, TMR_Y)

- MSTPCRL

Bit	Bit Name	Initial Value	R/W	Corresponding Module
7	MSTP7	1	R/W	Serial communication interface 3 (SCI_3)
6	MSTP6	1	R/W	Serial communication interface 1 (SCI_1)
5	MSTP5	1	R/W	Reserved The initial value should not be changed.
4	MSTP4	1	R/W	I^2C bus interface channel 0 (IIC_0)
3	MSTP3	1	R/W	I^2C bus interface channel 1 (IIC_1)
2	MSTP2	1	R/W	I^2C bus interface channel 2, 3 (IIC_2, IIC_3)
1	MSTP1	1	R/W	CRC operation circuit
0	MSTP0	1	R/W	I^2C bus interface channel 4, 5 (IIC_4, IIC_5)

MSTPCR sets operation and stop by the combination of bits as follows:

MSTPCRH (bit 3) MSTP11	MSTPCRA (bit 2) MSTPA2	Function
0	0	14-bit PWM timer (PWMX_1) operates.
0	1	14-bit PWM timer (PWMX_1) stops.
1	x	Reserved

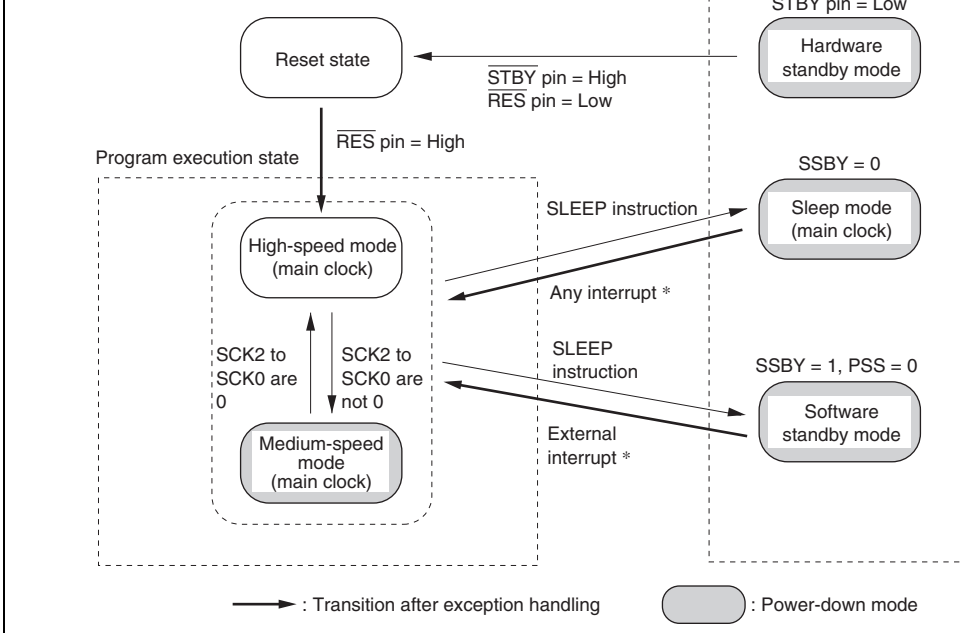
MSTPCRH (bit 3) MSTP11	MSTPCRA (bit 1) MSTPA1	Function
0	0	14-bit PWM timer (PWMX_0) operates.
0	1	14-bit PWM timer (PWMX_0) stops.
1	x	Reserved

Note: Bit 3 of MSTPCRH is the module stop bit for PWMX_0 and PWMX_1.

[Legend] X: Don't care

- SUBMSTPBL

Bit	Bit Name	Initial Value	R/W	Corresponding Module
7 to 4	SMSTPB7 to SMSTPB4	All 1	R/W	Reserved The initial values should not be changed.
3	SMSTPB3	1	R/W	Serial communication interface with FIFO (SCI
2	SMSTPB2	1	R/W	Reserved The initial values should not be changed.
1	SMSTPB1	1	R/W	LPC interface (LPC)
0	SMSTPB0	1	R/W	Reserved The initial values should not be changed.



Note: When a transition is made between modes by means of an interrupt, the transition cannot be m
on interrupt source generation alone. Ensure that interrupt handling is performed after acceptin
interrupt request.

* NMI, IRQ0 to IRQ15

Figure 24.1 Mode Transition Diagram

modules			medium-speed mode/ Functioning		Halted (retained)	(retained)	
	WDT_1		Functioning		Functioning		
	WDT_0						
	TMR_0,TMR_1				Functioning/ Halted (retained)		
	LPC						
	FRT						
	TMR_X, TMR_Y						
	IIC_0 to IIC_5						
	CRC						
	SCI_1, SCI_3	Functioning	Functioning	Functioning	Functioning /Halted (retained/ reset)	Halted (retained/ reset)	Hal
	SCIF						
	PWMX_0,PWMX_1				Functioning/ Halted (reset)	Halted(reset)	
	A/D converter						
	RAM			Functioning (DTC)	Functioning	Retained	Ret
	I/O			Functioning			Hig

Notes: Halted (retained) means that internal register values are retained. The internal stat operation suspended.

Halted (reset) means that internal register values and internal states are initialized.

In module stop mode, only modules for which a stop setting has been made are ha (reset or retained).

In medium-speed mode, a bus access is executed in the specified number of states with r
the bus master operating clock. For example, if ϕ/4 is selected as the operating clock, on
memory is accessed in 4 states, and internal I/O registers in 8 states.

By clearing all of bits SCK2 to SCK0 to 0, a transition is made to high-speed mode at th
the current bus cycle.

If a SLEEP instruction is executed when the SSBY bit in SBYCR is cleared to 0, a trans
made to sleep mode. When sleep mode is cleared by an interrupt, medium-speed mode i
When the SLEEP instruction is executed with the SSBY bit set to 1 and the PSS bit in T
(WDT_1) cleared to 0, operation shifts to software standby mode. When software standb
cleared by an external interrupt, medium-speed mode is restored.

When the $\overline{\text{RES}}$ pin is set low, medium-speed mode is cancelled and operation shifts to th
state. The same applies in the case of a reset caused by overflow of the watchdog timer.

When the $\overline{\text{STBY}}$ pin is driven low, a transition is made to hardware standby mode.

Figure 24.2 shows an example of medium-speed mode timing.

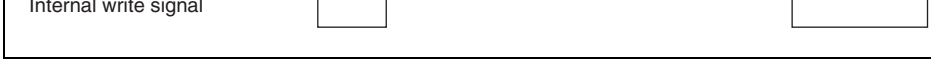


Figure 24.2 Medium-Speed Mode Timing

24.4 Sleep Mode

The CPU makes a transition to sleep mode if the SLEEP instruction is executed when the bit in SBYCR is cleared to 0. In sleep mode, CPU operation stops but the peripheral mod not stop. The contents of the CPU's internal registers are retained.

Sleep mode is exited by any interrupt, the $\overline{\text{RES}}$ pin, or the $\overline{\text{STBY}}$ pin.

When an interrupt occurs, sleep mode is exited and interrupt exception handling starts. Sl mode is not exited if the interrupt is disabled, or interrupts other than NMI are masked by CPU.

Setting the $\overline{\text{RES}}$ pin level low cancels sleep mode and selects the reset state. After the osc settling time has passed, driving the $\overline{\text{RES}}$ pin high causes the CPU to start reset exception handling.

When the $\overline{\text{STBY}}$ pin level is driven low, a transition is made to hardware standby mode.

input, or $\overline{\text{STBY}}$ pin input.

When an external interrupt request signal is input, system clock oscillation starts, and af
elapse of the time set in bits STS2 to STS0 in SBYCR, software standby mode is cleared
interrupt exception handling is started. When exiting software standby mode by IRQ0 to
interrupt, set the corresponding enable bit to 1 and ensure that any interrupt with a highe
than IRQ0 to IRQ15 is not generated. Software standby mode is not exited if the corresp
enable bit is cleared to 0 or if the interrupt has been masked by the CPU.

When the $\overline{\text{RES}}$ pin is driven low, system clock oscillation is started. At the same time as
clock oscillation starts, the system clock is supplied to the entire LSI. Note that the $\overline{\text{RES}}$
be held low until clock oscillation settles. When the $\overline{\text{RES}}$ pin goes high after clock oscill
settles, the CPU begins reset exception handling.

When the $\overline{\text{STBY}}$ pin is driven low, software standby mode is cancelled and a transition i
hardware standby mode.

Figure 24.3 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.

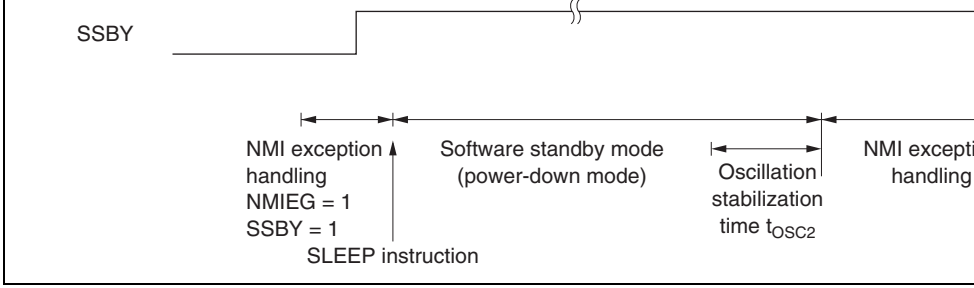


Figure 24.3 Software Standby Mode Application Example

Hardware standby mode is cleared by the $\overline{\text{STBY}}$ pin input or the $\overline{\text{RES}}$ pin input.

When the $\overline{\text{STBY}}$ pin is driven high while the $\overline{\text{RES}}$ pin is low, clock oscillation is started. that the $\overline{\text{RES}}$ pin is held low until system clock oscillation settles. When the $\overline{\text{RES}}$ pin is subsequently driven high after the clock oscillation settling time has passed, reset excep handling starts.

Figure 24.4 shows an example of hardware standby mode timing.

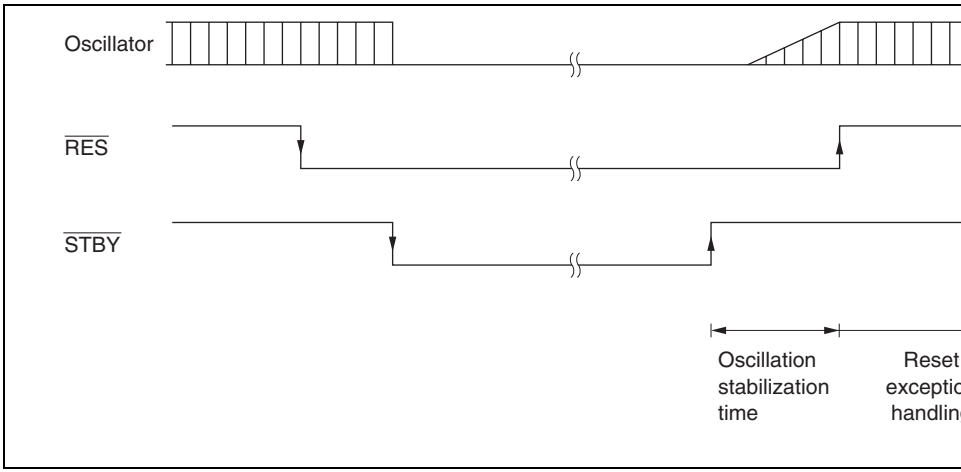


Figure 24.4 Hardware Standby Mode Timing

While an on-chip peripheral module is in module stop mode, read/write access to its regis
disabled.

24.8 Usage Notes

24.8.1 I/O Port Status

The status of the I/O ports is retained in software standby mode. Therefore, when a high
output, the current consumption is not reduced by the amount of current to support the hi
output.

24.8.2 Current Consumption when Waiting for Oscillation Settling

The current consumption increases during oscillation settling.

24.8.3 DTC Module Stop Mode

If the DTC module stop mode specification and DTC bus request occur simultaneously, t
released to the DTC and the MSTP bit cannot be set to 1. After completing the DTC bus
the MSTP bit to 1 again.

24.8.4 Notes on Subclock Usage

When using the subclock, make a transition to power-down mode after setting the EXCL
LPWRCR to 1 and loading the subclock two or more cycles. When not using the subcloc
EXCLE bit should not be set to 1.

- The access size is indicated.

2. Register Bits

- Bit configurations of the registers are described in the same order as the Register Ad (address order) above.
- Reserved bits are indicated by — in the bit name column.
- The bit number in the bit-name column indicates that the whole register is allocated counter or for holding data.
- 16-bit registers are indicated from the bit on the MSB side.

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

Receive buffer register	FRBR	8	H'FC80	SCIF	16
Transmitter holding register	FTHR	8	H'FC80	SCIF	16
Divisor latch L	FDLL	8	H'FC80	SCIF	16
Interrupt enable register	FIER	8	H'FC81	SCIF	16
Divisor latch H	FDLH	8	H'FC81	SCIF	16
Interrupt identification register	FIIR	8	H'FC82	SCIF	16
FIFO control register	FFCR	8	H'FC82	SCIF	16
Line control register	FLCR	8	H'FC83	SCIF	16
Modem control register	FMCR	8	H'FC84	SCIF	16
Line status register	FLSR	8	H'FC85	SCIF	16
Modem status register	FMSR	8	H'FC86	SCIF	16
Scratch pad register	FSCR	8	H'FC87	SCIF	16
SCIF control register	SCIFCR	8	H'FC88	SCIF	16
Host interface control register 4	HICR4	8	H'FD00	LPC	16
BT status register 0	BTSR0	8	H'FD02	LPC	16
BT status register 1	BTSR1	8	H'FD03	LPC	16
BT control/status register 0	BTCSR0	8	H'FD04	LPC	16
BT control/status register 1	BTCSR1	8	H'FD05	LPC	16
BT control register	BTCR	8	H'FD06	LPC	16
BT interrupt mask register	BTIMSR	8	H'FD07	LPC	16
SMIC flag register	SMICFLG	8	H'FD08	LPC	16
Host interface control register 5	HICR5	8	H'FD09	LPC	16
SMIC control/status register	SMICCSR	8	H'FD0A	LPC	16
SMIC data register	SMICDTR	8	H'FD0B	LPC	16

Bidirectional data register 3	TWR3	8	H'FD13	LPC	16
Bidirectional data register 4	TWR4	8	H'FD14	LPC	16
Bidirectional data register 5	TWR5	8	H'FD15	LPC	16
Bidirectional data register 6	TWR6	8	H'FD16	LPC	16
Bidirectional data register 7	TWR7	8	H'FD17	LPC	16
Bidirectional data register 8	TWR8	8	H'FD18	LPC	16
Bidirectional data register 9	TWR9	8	H'FD19	LPC	16
Bidirectional data register 10	TWR10	8	H'FD1A	LPC	16
Bidirectional data register 11	TWR11	8	H'FD1B	LPC	16
Bidirectional data register 12	TWR12	8	H'FD1C	LPC	16
Bidirectional data register 13	TWR13	8	H'FD1D	LPC	16
Bidirectional data register 14	TWR14	8	H'FD1E	LPC	16
Bidirectional data register 15	TWR15	8	H'FD1F	LPC	16
Input data register 3	IDR3	8	H'FD20	LPC	16
Output data register 3	ODR3	8	H'FD21	LPC	16
Status register 3	STR3	8	H'FD22	LPC	16
SERIRQ control register 4	SIRQCR4	8	H'FD23	LPC	16
LPC channel 3 address register H	LADR3H	8	H'FD24	LPC	16
LPC channel 3 address register L	LADR3L	8	H'FD25	LPC	16
SERIRQ control register 0	SIRQCR0	8	H'FD26	LPC	16
SERIRQ control register 1	SIRQCR1	8	H'FD27	LPC	16
Input data register 1	IDR1	8	H'FD28	LPC	16
Output data register 1	ODR1	8	H'FD29	LPC	16
Status register 1	STR1	8	H'FD2A	LPC	16

Host interface control register 2	HICR2	8	H'FD32	LPC	16
Host interface control register 3	HICR3	8	H'FD33	LPC	16
BT data buffer	BTDTR	8	H'FD35	LPC	16
BT FIFO valid size register 0	BTFVSR0	8	H'FD36	LPC	16
BT FIFO valid size register 1	BTFVSR1	8	H'FD37	LPC	16
LPC channel 1, 2 address register H	LADR12H	8	H'FD38	LPC	16
LPC channel 1, 2 address register L	LADR12L	8	H'FD39	LPC	16
SCIF address register H	SCIFADRH	8	H'FD3A	LPC	16
SCIF address register L	SCIFADRL	8	H'FD3B	LPC	16
Sub-chip module stop control register BH	SUBMSTPBH	8	H'FE3E	SYSTEM	8
Sub-chip module stop control register BL	SUBMSTPBL	8	H'FE3F	SYSTEM	8
Event count status register	ECS	16	H'FE40	EVC	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 mode control register	P6NCMC	8	H'FE45	PORT	8
Noise canceler cycle setting register	NCCS	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	PORT	8
Port E data direction register	PEDDR	8	H'FE4A	PORT	8
Port F input data register	PFPIN	8	H'FE4B	PORT	8
Port F data direction register	PFDDR	8	H'FE4B	PORT	8

Flash program code select register	FPCS	8	H'FE89	FLASH	8
Flash erase code select register	FECS	8	H'FE8A	FLASH	8
Flash key code register	FKEY	8	H'FE8C	FLASH	8
Flash MAT select register	FMATS	8	H'FE8D	FLASH	8
Flash transfer destination address register	FTDAR	8	H'FE8E	FLASH	8
I^2C bus control register_4	ICCR_4	8	H'FE90	IIC_4	8
I^2C bus status register_4	ICSR_4	8	H'FE91	IIC_4	8
I^2C bus data register_4	ICDR_4	8	H'FE92	IIC_4	8
Second slave address register_4	SARX_4	8	H'FE92	IIC_4	8
I^2C bus mode register_4	ICMR_4	8	H'FE93	IIC_4	8
Slave address register_4	SAR_4	8	H'FE93	IIC_4	8
I^2C bus control register_5	ICCR_5	8	H'FE94	IIC_5	8
I^2C bus status register_5	ICSR_5	8	H'FE95	IIC_5	8
I^2C bus data register_5	ICDR_5	8	H'FE96	IIC_5	8
Second slave address register_5	SARX_5	8	H'FE96	IIC_5	8
I^2C bus mode register_5	ICMR_5	8	H'FE97	IIC_5	8
Slave address register_5	SAR_5	8	H'FE97	IIC_5	8
Serial mode register_1	SMR_1	8	H'FE98	SCI_1	8
Bit rate register_1	BRR_1	8	H'FE99	SCI_1	8
Serial control register_1	SCR_1	8	H'FE9A	SCI_1	8
Transmit data register_1	TDR_1	8	H'FE9B	SCI_1	8
Serial status register_1	SSR_1	8	H'FE9C	SCI_1	8
Receive data register_1	RDR_1	8	H'FE9D	SCI_1	8

A/D data register G	ADDRG	16	H'FEAC	ADC	16
A/D data register H	ADDRH	16	H'FEAE	ADC	16
A/D control/status register	ADCSR	8	H'FEB0	ADC	8
A/D control register	ADCR	8	H'FEB1	ADC	8
Serial multiplexed mode register 0	SMR0	8	H'FEB8	SMX	8
Serial multiplexed mode register 1	SMR1	8	H'FEB9	SMX	8
Port 6 pull-up MOS control register	P6PCR	8	H'FEBC	PORT	8
Pin function control register	PINFNCR	8	H'FEBE	PORT	8
Port 4 pull-up MOS control register	P4PCR	8	H'FEBF	PORT	8
I^2C bus control register_3	ICCR_3	8	H'FEC0	IIC_3	8
I^2C bus status register_3	ICSR_3	8	H'FEC1	IIC_3	8
I^2C 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
I^2C bus mode register_3	ICMR_3	8	H'FEC3	IIC_3	8
Slave address register_3	SAR_3	8	H'FEC3	IIC_3	8
I^2C bus control register_2	ICCR_2	8	H'FEC8	IIC_2	8
I^2C bus status register_2	ICSR_2	8	H'FEC9	IIC_2	8
I^2C bus data register_2	ICDR_2	8	H'FECA	IIC_2	8
Second slave address register_2	SARX_2	8	H'FECA	IIC_2	8
I^2C bus mode register_2	ICMR_2	8	H'FECB	IIC_2	8
Slave address register_2	SAR_2	8	H'FECB	IIC_2	8
PWMX (D/A) data register A_1	DADRA_1	16	H'FECC	PWMX_1	8
PWMX (D/A) control register_1	DACR_1	8	H'FECC	PWMX_1	8
PWMX (D/A) data register B_1	DADRB_1	16	H'FECE	PWMX_1	8

I^2C bus extended control register_2	ICXR_2	8	H'FEDC	IIC_2	8
I^2C bus extended control 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 extended control register_4	ICXR_4	8	H'FEE0	IIC_4	8
I^2C bus extended control register_5	ICXR_5	8	H'FEE1	IIC_5	8
Keyboard comparator control register	KBCOMP	8	H'FEE4	EVC	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
DTC enable register A	DTCERA	8	H'FEEE	DTC	8
DTC enable register B	DTCERB	8	H'FEEF	DTC	8
DTC enable register C	DTCERC	8	H'FEF0	DTC	8
DTC enable register D	DTCERD	8	H'FEF1	DTC	8
DTC enable register E	DTCERE	8	H'FEF2	DTC	8
DTC vector register	DTVECR	8	H'FEF3	DTC	8
Address break control register	ABRKCR	8	H'FEF4	INT	8
Break address register A	BARA	8	H'FEF5	INT	8
Break address register B	BARB	8	H'FEF6	INT	8
Break address register C	BARC	8	H'FEF7	INT	8
IRQ enable register 16	IER16	8	H'FEF8	INT	8

Wait state control register 2	WSCR2	8	H'FF81	BSC	8
Peripheral clock select register	PCSR	8	H'FF82	PWMX	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
I^2C bus control register_1	ICCR_1	8	H'FF88	IIC_1	8
I^2C bus status register _1	ICSR_1	8	H'FF89	IIC_1	8
I^2C bus data register _1	ICDR_1	8	H'FF8E	IIC_1	8
Second slave address register _1	SARX_1	8	H'FF8E	IIC_1	8
I^2C bus mode register_1	ICMR_1	8	H'FF8F	IIC_1	8
Slave address register _1	SAR_1	8	H'FF8F	IIC_1	8
Timer interrupt enable register	TIER	8	H'FF90	FRT	8
Timer control/status register	TCSR	8	H'FF91	FRT	8
Free-running counter	FRC	16	H'FF92	FRT	16
Output compare register A	OCRA	16	H'FF94	FRT	16
Output compare register B	OCRB	16	H'FF95	FRT	16
Timer control register	TCR	8	H'FF96	FRT	16
Timer output compare control register	TOCR	8	H'FF97	FRT	16
Output compare register AR	OCRAR	16	H'FF98	FRT	16
Output compare register AF	OCRAF	16	H'FF9A	FRT	16
PWMX (D/A) data register A_0	DADRA_0	16	H'FFA0	PWMX_0	8
PWMX (D/A) control register_0	DACR_0	8	H'FFA0	PWMX_0	8

Port A input data register	PAPIN	8	H'FFAB	PORT	8
Port A data direction register	PADDR	8	H'FFAB	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
Port 6 data register	P6DR	8	H'FFBB	PORT	8
Port B output data register	PBODR	8	H'FFBC	PORT	8
Port B input data register	PBPIN	8	H'FFBD	PORT	8
Port 8 data direction register	P8DDR	8	H'FFBD	PORT	8
Port 7 input data register	P7PIN	8	H'FFBE	PORT	8
Port B data direction register	PBDDR	8	H'FFBE	PORT	8
Port 8 data register	P8DR	8	H'FFBF	PORT	8
Port 9 data direction register	P9DDR	8	H'FFC0	PORT	8

Timer control register_0	TCR_0	8	H'FFC8	TMR_0	8
Timer control register_1	TCR_1	8	H'FFC9	TMR_1	8
Timer control/status register_0	TCSR_0	8	H'FFCA	TMR_0	8
Timer control/status register_1	TCSR_1	8	H'FFCB	TMR_1	8
Time constant register A_0	TCORA_0	8	H'FFCC	TMR_0	8
Time constant register A_1	TCORA_1	8	H'FFCD	TMR_1	8
Time constant register B_0	TCORB_0	8	H'FFCE	TMR_0	8
Time constant register B_1	TCORB_1	8	H'FFCF	TMR_1	8
Timer counter_0	TCNT_0	8	H'FFD0	TMR_0	8
Timer counter_1	TCNT_1	8	H'FFD1	TMR_1	8
I^2C bus control register_0	ICCR_0	8	H'FFD8	IIC_0	8
I^2C bus status register_0	ICSR_0	8	H'FFD9	IIC_0	8
I^2C bus data register_0	ICDR_0	8	H'FFDE	IIC_0	8
Second slave address register_0	SARX_0	8	H'FFDE	IIC_0	8
I^2C bus mode register_0	ICMR_0	8	H'FFDF	IIC_0	8
Slave address register_0	SAR_0	8	H'FFDF	IIC_0	8
Serial mode register_3	SMR_3	8	H'FFE0	SCI_3	8
Bit rate register_3	BRR_3	8	H'FFE1	SCI_3	8
Serial control register_3	SCR_3	8	H'FFE2	SCI_3	8
Transmit data register_3	TDR_3	8	H'FFE3	SCI_3	8
Serial status register_3	SSR_3	8	H'FFE4	SCI_3	8
Receive data register_3	RDR_3	8	H'FFE5	SCI_3	8
Smart card mode register_3	SCMR_3	8	H'FFE6	SCI_3	8
Timer control/ status register_1 (read)	TCSR_1	8	H'FFEA	WDT_1	16

Time constant register B_X	TCORB_X	8	H'FFF7	TMR_X	8
Timer control register_Y	TCR_Y	8	H'FFF0	TMR_Y	8
Timer control/status register_Y	TCSR_Y	8	H'FFF1	TMR_Y	8
Time constant register A_Y	TCORA_Y	8	H'FFF2	TMR_Y	8
Time constant register B_Y	TCORB_Y	8	H'FFF3	TMR_Y	8
Timer counter_Y	TCNT_Y	8	H'FFF4	TMR_Y	8
Timer connection register S	TCONRS	8	H'FFFE	TMR	8

FDLL	bit 7	bit 6	bit 5	bit 4	bit 3	bit 2	bit 1	bit 0
FIER	—	—	—	—	EDSSI	ELSI	ETBEI	ERBFI
FDLH	bit 7	bit 6	bit 5	bit 4	bit 3	bit 2	bit 1	bit 0
FIIR	FIFOE1	FIFOE0	—	—	INTID2	INTID1	INTID0	INTPEND
FFCR	RCVRTRIG1	RCVRTRIG0	—	—	DMAMODE	XMITFRST	RCVRFRST	FIFOE
FLCR	DLAB	BREAK	STICKPARITY	EPS	PEN	STOP	CLS1	CLS0
FMCR	—	—	—	LOOPBACK	OUT2	OUT1	RTS	DTR
FLSR	RXFIFOERR	TEMT	THRE	BI	FE	PE	OE	DR
FMSR	DCR	RI	DSR	CTS	DDCD	TERI	DDSR	DCTS
FSCR	bit 7	bit 6	bit 5	bit 4	bit 3	bit 2	bit 1	bit 0
SCIFCR	SCIFOE1	SCIFOE0	bit 5	OUT2LOOP	CKSEL1	CKSEL0	SCIFRST	REGRST
HICR4	LADR12SEL	—	—	—	SWENBL	KCSENBL	SMCENBL	BTENBL
BTSR0	—	—	—	FRDI	HRDI	HWRI	HBTWI	HBTRI
BTSR1	—	HRSTI	IRQCRI	BEVTI	B2HI	H2BI	CRRPI	CRWPI
BTCSR0	—	FSEL1	FSEL0	FRDIE	HRDIE	HWRIE	HBTWIE	HBTRIE
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	—	—	OEM3	OEM2	OEM1	B2H_IRQ	B2H_IRQ EN
SMICFLG	RX_DATA_RDY	TX_DATA_RDY	—	SMI	SEVT_ATN	SMS_ATN	—	BUSY
HICR5	—	—	—	—	—	—	SCIFE	—
SMICCSR	bit 7	bit 6	bit 5	bit 4	bit 3	bit 2	bit 1	bit 0
SMICDTR	bit 7	bit 6	bit 5	bit 4	bit 3	bit 2	bit 1	bit 0
SMICIR0	—	—	—	HDTWI	HDTRI	STARI	CTLWI	BUSYI

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
TWR14	bit 7	bit 6	bit 5	bit 4	bit 3	bit 2	bit 1	bit 0
TWR15	bit 7	bit 6	bit 5	bit 4	bit 3	bit 2	bit 1	bit 0
IDR3	bit 7	bit 6	bit 5	bit 4	bit 3	bit 2	bit 1	bit 0
ODR3	bit 7	bit 6	bit 5	bit 4	bit 3	bit 2	bit 1	bit 0
STR3*1	IBF3B	OBF3B	MWMF	SWMF	C/$\overline{D}$3	DBU32	IBF3A	OBF3A
STR3*2	DBU37	DBU36	DBU35	DBU34	C/$\overline{D}$3	DBU32	IBF3A	OBF3A
SIRQCR4	IRQ15E	IRQ14E	IRQ13E	IRQ8E	IRQ7E	IRQ5E	IRQ4E	IRQ3E
LADR3H	bit 15	bit 14	bit 13	bit 12	bit 11	bit 10	bit 9	bit 8
LADR3L	bit 7	bit 6	bit 5	bit 4	bit 3	—	bit 1	TWRE
SIRQCR0	Q/C	SELREQ	IEDIR2	SMIE3B	SMIE3A	SMIE2	IRQ12E1	IRQ1E1
SIRQCR1	IRQ11E3	IRQ10E3	IRQ9E3	IRQ6E3	IRQ11E2	IRQ10E2	IRQ9E2	IRQ6E2
IDR1	bit 7	bit 6	bit 5	bit 4	bit 3	bit 2	bit 1	bit 0
ODR1	bit 7	bit 6	bit 5	bit 4	bit 3	bit 2	bit 1	bit 0
STR1	DBU17	DBU16	DBU15	DBU14	C/$\overline{D}$1	DBU12	IBF1	OBF1
SIRQCR5	SELIRQ15	SELIRQ14	SELIRQ13	SELIRQ8	SELIRQ7	SELIRQ5	SELIRQ4	SELIRQ

HCR3	LFRAME	CERRON	SERIRQ	LRESET	LPCPD	PME	ESMI	ESCI
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
SCIFADRH	bit 15	bit 14	bit 13	bit 12	bit 11	bit 10	bit 9	bit 8
SCIFADRL	bit 7	bit 6	bit 5	bit 4	bit 3	—	—	—
SUBMSTPBH	SMSTPB15	SMSTPB14	SMSTPB13	SMSTPB12	SMSTPB11	SMSTPB10	SMSTPB9	SMSTPB8
SUBMSTPBL	SMSTPB7	SMSTPB6	SMSTPB5	SMSTPB4	SMSTPB3	SMSTPB2	SMSTPB1	SMSTPB0
ECS	E15	E14	E13	E12	E11	E10	E9	E8
	E7	E6	E5	E4	E3	E2	E1	E0
ECCR	EDSB	—	—	—	ECSB3	ECSB2	ECSB1	ECSB0
MSTPCRA	MSTPA7	MSTPA6	MSTPA5	MSTPA4	MSTPA3	MSTPA2	MSTPA1	MSTPA0
P6NCE	P67NCE	P66NCE	P65NCE	P64NCE	P63NCE	P62NCE	P61NCE	P60NCE
P6NCMC	P67NCMC	P66NCMC	P65NCMC	P64NCMC	P63NCMC	P62NCMC	P61NCMC	P60NCMC
NCCS	—	—	—	—	—	NCCK2	NCCK1	NCCK0
PEODR	PE7ODR	PE6ODR	PE5ODR	PE4ODR	PE3ODR	PE2ODR	PE1ODR	PE0ODR
PFODR	—	—	—	—	PF3ODR	PF2ODR	PF0ODR	PF0ODR
PEPIN	PE7PIN	PE6PIN	PE5PIN	PE4PIN	PE3PIN	PE2PIN	PE1PIN	PE0PIN
PEDDR	PE7DDR	PE6DDR	PE5DDR	PE4DDR	PE3DDR	PE2DDR	PE1DDR	PE0DDR
PFPIN	—	—	—	—	PF3PIN	PF2PIN	PF1PIN	PF0PIN
PFDDR	—	—	—	—	PF3DDR	PF2DDR	PF1DDR	PF0DDR
PCODR	PC7ODR	PC6ODR	PC5ODR	PC4ODR	PC3ODR	PC2ODR	PC1ODR	PC0ODR

FLCS	—	—	—	—	—	—	—	LVVB
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
ICCR_5	ICE	IEIC	MST	TRS	ACKE	BBSY	IRIC	SCP
ICSR_5	ESTP	STOP	IRTR	AASX	AL	AAS	ADZ	ACKB
ICDR_5	bit 7	bit 6	bit 5	bit 4	bit 3	bit 2	bit 1	bit 0
SARX_5	SVAX6	SVAX5	SVAX4	SVAX3	SVAX2	SVAX1	SVAX0	FSX
ICMR_5	MLS	WAIT	CKS2	CKS1	CKS0	BC2	BC1	BC0
SAR_5	SVA6	SVA5	SVA4	SVA3	SVA2	SVA1	SVA0	FS
SMR_1※3	C/$\overline{A}$ (GM)	CHR (BLK)	PE (PE)	O/$\overline{E}$ (O/$\overline{E}$)	STOP (BCP1)	MP (BCP0)	CKS1 (CKS1)	CKS0 (CKS0)
BRR_1	bit 7	bit 6	bit 5	bit 4	bit 3	bit 2	bit 1	bit 0
SCR_1	TIE	RIE	TE	RE	MPIE	TEIE	CKE1	CKE0
TDR_1	bit 7	bit 6	bit 5	bit 4	bit 3	bit 2	bit 1	bit 0
SSR_1※3	TDRE (TDRE)	RDRF (RRF)	ORER (ORER)	FER (ERS)	PER (PER)	TEND (TEND)	MPB (MPB)	MPBT (MPBT)
RDR_1	bit 7	bit 6	bit 5	bit 4	bit 3	bit 2	bit 1	bit 0
SCMR_1	—	—	—	—	SDIR	SINV	—	SMIF

	AD1	AD0	—	—	—	—	—	—
ADDRE	AD9	AD8	AD7	AD6	AD5	AD4	AD3	AD2
	AD1	AD0	—	—	—	—	—	—
ADDRF	AD9	AD8	AD7	AD6	AD5	AD4	AD3	AD2
	AD1	AD0	—	—	—	—	—	—
ADDRG	AD9	AD8	AD7	AD6	AD5	AD4	AD3	AD2
	AD1	AD0	—	—	—	—	—	—
ADDRH	AD9	AD8	AD7	AD6	AD5	AD4	AD3	AD2
	AD1	AD0	—	—	—	—	—	—
ADCSR	ADF	ADIE	ADST	—	—	CH2	CH1	CH0
ADCR	TRGS1	TRGS0	SCANE	SCANS	CKS1	CKS0	ADSTCLR	EXTRGS
SMR0	DCD1	RI1	DSR1	SME	—	SM2	SM1	SM0
SMR1	CTS1	DTR1	RTS1	CTS3	—	RTS3	—	—
P6PCR	P67PCR	P66PCR	P65PCR	P64PCR	P63PCR	P62PCR	P61PCR	P60PCR
PINFNCR	—	—	—	—	—	SERIRQ OFF	LPCPD OFF	CLKRUN OFF
P4PCR	P47PCR	P46PCR	P45PCR	P44PCR	P43PCR	P42PCR	P41PCR	P40PCR
ICCR_3	ICE	IEIC	MST	TRS	ACKE	BBSY	IRIC	SCP
ICSR_3	ESTP	STOP	IRTR	AASX	AL	AAS	ADZ	ACKB
ICDR_3	bit 7	bit 6	bit 5	bit 4	bit 3	bit 2	bit 1	bit 0
SARX_3	SVAX6	SVAX5	SVAX4	SVAX3	SVAX2	SVAX1	SVAX0	FSX
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

DADRB_1	DA13	DA12	DA11	DA10	DA9	DA8	DA7	DA6
	DA5	DA4	DA3	DA2	DA1	DA0	CFS	REGS
DACNT_1	UC7	UC6	UC5	UC4	UC3	UC2	UC1	UC0
	UC8	UC9	UC10	UC11	UC12	UC13	—	REGS
CRCCR	DORCLR	—	—	—	—	LMS	G1	G0
CRCDIR	bit 7	bit 6	bit 5	bit 4	bit 3	bit 2	bit 1	bit 0
CRCDOR	bit 15	bit 14	bit 13	bit 12	bit 11	bit 10	bit 9	bit 8
	bit 7	bit 6	bit 5	bit 4	bit 3	bit 2	bit 1	bit 0
ICXR_0	STOPIM	HNDS	ICDRF	ICDRE	ALIE	ALSL	FNC1	FNC0
ICXR_1	STOPIM	HNDS	ICDRF	ICDRE	ALIE	ALSL	FNC1	FNC0
ICSMBCR	SMB5E	SMB4E	SMB3E	SMB2E	SMB1E	SMB0E	FSEL1	FSEL0
ICXR_2	STOPIM	HNDS	ICDRF	ICDRE	ALIE	ALSL	FNC1	FNC0
ICXR_3	STOPIM	HNDS	ICDRF	ICDRE	ALIE	ALSL	FNC1	FNC0
IICX3	—	—	—	—	TCSS	IICX5	IICX4	IICX3
ICXR_4	STOPIM	HNDS	ICDRF	ICDRE	ALIE	ALSL	FNC1	FNC0
ICXR_5	STOPIM	HNDS	ICDRF	ICDRE	ALIE	ALSL	FNC1	FNC0
KBCOMP	EVENTE	—	—	—	—	—	—	—
ICRD	ICRD7	ICRD6	—	—	—	—	ICRD1	—
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	IRQ4SC
ISCRL	IRQ3SCB	IRQ3SCA	IRQ2SCB	IRQ2SCA	IRQ1SCB	IRQ1SCA	IRQ0SCB	IRQ0SC

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	IRQ12SC
ISCR16L	IRQ11SCB	IRQ11SCA	IRQ10SCB	IRQ10SCA	IRQ9SCB	IRQ9SCA	IRQ8SCB	IRQ8SCA
ISSR16	ISS15	ISS14	ISS13	ISS12	ISS11	ISS10	ISS9	ISS8
ISSR	ISS7	ISS6	ISS5	ISS4	ISS3	ISS2	ISS1	ISS0
PTCNT0	SCPESEL1	SCPFSEL3	—	—	—	—	OBE	—
BCR2	—	—	—	—	ADFULLE	EXCKS	—	—
WSCR2	WMS10	WC11	WC10	—	—	—	—	—
PCSR	PWCKX1B	PWCKX1A	PWCKX0B	PWCKX0A	PWCKX1C	—	—	PWCKX0
SYSCR2	—	—	—	—	ADMXE	—	—	—
SBYCR	SSBY	STS2	STS1	STS0	DTSPEED	SCK2	SCK1	SCK0
LPWRCR	—	—	NESEL	EXCLE	—	PNCCS	PNCAH	—
MSTPCRH	MSTP15	MSTP14	MSTP13	MSTP12	MSTP11	MSTP10	MSTP9	MSTP8
MSTPCRL	MSTP7	MSTP6	MSTP5	MSTP4	MSTP3	MSTP2	MSTP1	MSTP0
ICCR_1	ICE	IEIC	MST	TRS	ACKE	BBSY	IRIC	SCP
ICSR_1	ESTP	STOP	IRTR	AASX	AL	AAS	ADZ	ACKB
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

	bit 7	bit 6	bit 5	bit 4	bit 3	bit 2	bit 1	bit 0
TCR	—	—	—	—	—	—	CKS1	CKS0
TOCR	—	OCRAMS	ICRS	OCRS	—	—	—	—
OCRAR	bit 15	bit 14	bit 13	bit 12	bit 11	bit 10	bit 9	bit 8
	bit 7	bit 6	bit 5	bit 4	bit 3	bit 2	bit 1	bit 0
OCRAF	bit 15	bit 14	bit 13	bit 12	bit 11	bit 10	bit 9	bit 8
	bit 7	bit 6	bit 5	bit 4	bit 3	bit 2	bit 1	bit 0
DADRA_0	DA13	DA12	DA11	DA10	DA9	DA8	DA7	DA6
	DA5	DA4	DA3	DA2	DA1	DA0	CFS	—
DACR_0	—	PWME	—	—	OEB	OEA	OS	CKS
DADRB_0	DA13	DA12	DA11	DA10	DA9	DA8	DA7	DA6
	DA5	DA4	DA3	DA2	DA1	DA0	CFS	REGS
DACNT_0	UC7	UC6	UC5	UC4	UC3	UC2	UC1	UC0
	UC8	UC9	UC10	UC11	UC12	UC13	—	REGS
TCSR_0	OVF	WT/$\overline{\text{IT}}$	TME	—	RST/$\overline{\text{NMI}}$	CKS2	CKS1	CKS0
TCNT_0	bit 7	bit 6	bit 5	bit 4	bit 3	bit 2	bit 1	bit 0
PAODR	PA7ODR	PA6ODR	PA5ODR	PA4ODR	PA3ODR	PA2ODR	PA1ODR	PA0OD
PAPIN	PA7PIN	PA6PIN	PA5PIN	PA4PIN	PA3PIN	PA2PIN	PA1PIN	PA0PIN
PADDR	PA7DDR	PA6DDR	PA5DDR	PA4DDR	PA3DDR	PA2DDR	PA1DDR	PA0DD
P1PCR	P17PCR	P16PCR	P15PCR	P14PCR	P13PCR	P12PCR	P11PCR	P10PC
P2PCR	P27PCR	P26PCR	P25PCR	P24PCR	P23PCR	P22PCR	P21PCR	P20PC
P3PCR	P37PCR	P36PCR	P35PCR	P34PCR	P33PCR	P32PCR	P31PCR	P30PC
P1DDR	P17DDR	P16DDR	P15DDR	P14DDR	P13DDR	P12DDR	P11DDR	P10DD
P2DDR	P27DDR	P26DDR	P25DDR	P24DDR	P23DDR	P22DDR	P21DDR	P20DD
P1DR	P17DR	P16DR	P15DR	P14DR	P13DR	P12DR	P11DR	P10DR

P5DR	P57DR	P56DR	P55DR	P54DR	P53DR	P52DR	P51DR	P50DR
P6DR	P67DR	P66DR	P65DR	P64DR	P63DR	P62DR	P61DR	P60DR
PBODR	PB7ODR	PB6ODR	PB5ODR	PB4ODR	PB3ODR	PB2ODR	PB1ODR	PB0ODR
PBPIN	PB7PIN	PB6PIN	PB5PIN	PB4PIN	PB3PIN	PB2PIN	PB1PIN	PB0PIN
P8DDR	P87DDR	P86DDR	P85DDR	P84DDR	P83DDR	P82DDR	P81DDR	P80DDR
P7PIN	P77PIN	P76PIN	P75PIN	P74PIN	P73PIN	P72PIN	P71PIN	P70PIN
PBDDR	PB7DDR	PB6DDR	PB5DDR	PB4DDR	PB3DDR	PB2DDR	PB1DDR	PB0DDR
P8DR	P87DR	P86DR	P85DR	P84DR	P83DR	P82DR	P81DR	P80DR
P9DDR	P97DDR	P96DDR	P95DDR	P94DDR	P93DDR	P92DDR	P91DDR	P90DDR
P9DR	P97DR	P96DR	P95DR	P94DR	P93DR	P92DR	P91DR	P90DR
IER	IRQ7E	IRQ6E	IRQ5E	IRQ4E	IRQ3E	IRQ2E	IRQ1E	IRQ0E
STCR	IICX2	IICX1	IICX0	—	FLSHE	—	ICKS1	ICKS0
SYSCR	CS256E	IOSE	INTM1	INTM0	XRST	NMIEG	—	RAME
MDCR	EXPE	—	—	—	—	MDS2	MDS1	—
BCR	—	ICIS	BRSTRM	BRSTS1	BRSTS0	—	IOS1	IOS0
WSCR	ABW256	AST256	ABW	AST	WMS1	WMS0	WC1	WC0
TCR_0	CMIEB	CMIEA	OVIE	—	—	CKS2	CKS1	CKS0
TCR_1	CMIEB	CMIEA	OVIE	—	—	CKS2	CKS1	CKS0
TCSR_0	CMFB	CMFA	OVF	ADTE	—	—	—	—
TCSR_1	CMFB	CMFA	OVF	—	—	—	—	—
TCORA_0	bit 7	bit 6	bit 5	bit 4	bit 3	bit 2	bit 1	bit 0
TCORA_1	bit 7	bit 6	bit 5	bit 4	bit 3	bit 2	bit 1	bit 0
TCORB_0	bit 7	bit 6	bit 5	bit 4	bit 3	bit 2	bit 1	bit 0
TCORB_1	bit 7	bit 6	bit 5	bit 4	bit 3	bit 2	bit 1	bit 0
TCNT_0	bit 7	bit 6	bit 5	bit 4	bit 3	bit 2	bit 1	bit 0

SMR_3	C/A (GM)	CHR (BLK)	PE	O/E (O/$\overline{E}$)	STOP (BCP1)	MP (BCP0)	CKS1 (CKS1)	CKS0 (CKS0)
BRR_3	bit 7	bit 6	bit 5	bit 4	bit 3	bit 2	bit 1	bit 0
SCR_3	TIE	RIE	TE	RE	MPIE	TIE	CKE1	CKE0
TDR_3	bit 7	bit 6	bit 5	bit 4	bit 3	bit 2	bit 1	bit 0
SSR_3*[3]	TDRE (TDRE)	RDRF (RDRF)	ORER (ORER)	FER (ERS)	PER (PER)	TEND (TEND)	MPB (MPB)	MPBT (MPBT)
RDR_3	bit 7	bit 6	bit 5	bit 4	bit 3	bit 2	bit 1	bit 0
SCMR_3	—	—	—	—	SDIR	SINV	—	SMIF
TCSR_1	OVF	WT/$\overline{IT}$	TME	PSS	RST/$\overline{NMI}$	CKS2	CKS1	CKS0
TCNT_1	bit 7	bit 6	bit 5	bit 4	bit 3	bit 2	bit 1	bit 0
TCR_X	CMIEB	CMIEA	OVIE	—	—	CKS2	CKS1	CKS0
TCSR_X	CMFB	CMFA	OVF	—	—	—	—	—
TCNT_X	bit 7	bit 6	bit 5	bit 4	bit 3	bit 2	bit 1	bit 0
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
TCR_Y	CMIEB	CMIEA	OVIE	—	—	CKS2	CKS1	CKS0
TCSR_Y	CMFB	CMFA	OVF	—	—	—	—	—
TCORA_Y	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
TCNT_Y	bit 7	bit 6	bit 5	bit 4	bit 3	bit 2	bit 1	bit 0
TCONRS	TMRX/Y	—	—	—	—	—	—	—

FFCR	Initialized	Initialized	—	—	—	—	Initialized
FLCR	Initialized	Initialized	—	—	—	—	Initialized
FMCR	Initialized	Initialized	—	—	—	—	Initialized
FLSR	Initialized	Initialized	—	—	—	—	Initialized
FMSR	Initialized	Initialized	—	—	—	—	Initialized
FSCR	Initialized	Initialized	—	—	—	—	Initialized
SCIFCR	Initialized	Initialized	—	—	—	—	Initialized
HICR4	Initialized	Initialized	—	—	—	—	Initialized
BTSR0	Initialized	Initialized	—	—	—	—	Initialized
BTSR1	Initialized	Initialized	—	—	—	—	Initialized
BTCSR0	Initialized	Initialized	—	—	—	—	Initialized
BTCSR1	Initialized	Initialized	—	—	—	—	Initialized
BTCR	Initialized	Initialized	—	—	—	—	Initialized
BTIMSR	Initialized	Initialized	—	—	—	—	Initialized
SMICFLG	Initialized	Initialized	—	—	—	—	Initialized
HICR5	Initialized	Initialized	—	—	—	—	Initialized
SMICCSR	—	—	—	—	—	—	—
SMICDTR	—	—	—	—	—	—	—
SMICIR0	Initialized	Initialized	—	—	—	—	Initialized
SMICIR1	Initialized	Initialized	—	—	—	—	Initialized
SIRQCR3	Initialized	Initialized	—	—	—	—	Initialized
TWR0MW	—	—	—	—	—	—	—
TWR0SW	—	—	—	—	—	—	—

TWR9	—	—	—	—	—	—	—
TWR10	—	—	—	—	—	—	—
TWR11	—	—	—	—	—	—	—
TWR12	—	—	—	—	—	—	—
TWR13	—	—	—	—	—	—	—
TWR14	—	—	—	—	—	—	—
TWR15	—	—	—	—	—	—	—
IDR3	—	—	—	—	—	—	—
ODR3	—	—	—	—	—	—	—
STR3	Initialized	Initialized	—	—	—	—	Initialized
SIRQCR4	Initialized	Initialized	—	—	—	—	Initialized
LADR3H	Initialized	Initialized	—	—	—	—	Initialized
LADR3L	Initialized	Initialized	—	—	—	—	Initialized
SIRQCR0	Initialized	Initialized	—	—	—	—	Initialized
SIRQCR1	Initialized	Initialized	—	—	—	—	Initialized
IDR1	—	—	—	—	—	—	—
ODR1	—	—	—	—	—	—	—
STR1	Initialized	Initialized	—	—	—	—	Initialized
SIRQCR5	Initialized	Initialized	—	—	—	—	Initialized
IDR2	—	—	—	—	—	—	—
ODR2	—	—	—	—	—	—	—
STR2	Initialized	Initialized	—	—	—	—	Initialized
HISEL	Initialized	Initialized	—	—	—	—	Initialized

LADR12H	Initialized	Initialized	—	—	—	—	Initialized
LADR12L	Initialized	Initialized	—	—	—	—	Initialized
SCIFADRH	Initialized	Initialized	—	—	—	—	Initialized
SCIFADRL	Initialized	Initialized	—	—	—	—	Initialized
SUBMSTPBH	Initialized	Initialized	—	—	—	—	Initialized
SUBMSTPBL	Initialized	Initialized	—	—	—	—	Initialized
ECS	Initialized	Initialized	—	—	—	—	Initialized
ECCR	Initialized	Initialized	—	—	—	—	Initialized
MSTPCRA	Initialized	Initialized	—	—	—	—	Initialized
P6NCE	Initialized	Initialized	—	—	—	—	Initialized
P6NCMC	Initialized	Initialized	—	—	—	—	Initialized
NCCS	Initialized	Initialized	—	—	—	—	Initialized
PEODR	Initialized	Initialized	—	—	—	—	Initialized
PFODR	—	—	—	—	—	—	Initialized
PEPIN	—	—	—	—	—	—	—
PEDDR	Initialized	Initialized	—	—	—	—	Initialized
PFPIN	—	—	—	—	—	—	—
PFDDR	Initialized	—	—	—	—	—	Initialized
PCODR	Initialized	Initialized	—	—	—	—	Initialized
PDODR	Initialized	Initialized	—	—	—	—	Initialized
PCPIN	—	—	—	—	—	—	—
PCDDR	Initialized	Initialized	—	—	—	—	Initialized
PDPIN	—	—	—	—	—	—	—
PDDDR	Initialized	Initialized	—	—	—	—	Initialized

ICDR_4	—	—	—	—	—	—	—
SARX_4	Initialized	Initialized	—	—	—	—	Initialized
ICMR_4	Initialized	Initialized	—	—	—	—	Initialized
SAR_4	Initialized	Initialized	—	—	—	—	Initialized
ICCR_5	Initialized	Initialized	—	—	—	—	Initialized
ICSR_5	Initialized	Initialized	—	—	—	—	Initialized
ICDR_5	—	—	—	—	—	—	—
SARX_5	Initialized	Initialized	—	—	—	—	Initialized
ICMR_5	Initialized	Initialized	—	—	—	—	Initialized
SAR_5	Initialized	Initialized	—	—	—	—	Initialized
SMR_1	Initialized	Initialized	—	—	—	—	Initialized
BRR_1	Initialized	Initialized	—	—	—	—	Initialized
SCR_1	Initialized	Initialized	—	—	—	—	Initialized
TDR_1	Initialized	Initialized	—	—	Initialized	Initialized	Initialized
SSR_1	Initialized	Initialized	—	—	Initialized	Initialized	Initialized
RDR_1	Initialized	Initialized	—	—	Initialized	Initialized	Initialized
SCMR_1	Initialized	Initialized	—	—	—	—	Initialized
ADDRA	Initialized	Initialized	—	—	Initialized	Initialized	Initialized
ADDRB	Initialized	Initialized	—	—	Initialized	Initialized	Initialized
ADDRC	Initialized	Initialized	—	—	Initialized	Initialized	Initialized
ADDRD	Initialized	Initialized	—	—	Initialized	Initialized	Initialized
ADDRE	Initialized	Initialized	—	—	Initialized	Initialized	Initialized
ADDRF	Initialized	Initialized	—	—	Initialized	Initialized	Initialized
ADDRG	Initialized	Initialized	—	—	Initialized	Initialized	Initialized

ICCR_3	Initialized	Initialized	—	—	—	—	Initialized
ICSR_3	Initialized	Initialized	—	—	—	—	Initialized
ICDR_3	—	—	—	—	—	—	—
SARX_3	Initialized	Initialized	—	—	—	—	Initialized
ICMR_3	Initialized	Initialized	—	—	—	—	Initialized
SAR_3	Initialized	Initialized	—	—	—	—	Initialized
ICCR_2	Initialized	Initialized	—	—	—	—	Initialized
ICSR_2	Initialized	Initialized	—	—	—	—	Initialized
ICDR_2	—	—	—	—	—	—	—
SARX_2	Initialized	Initialized	—	—	—	—	Initialized
ICMR_2	Initialized	Initialized	—	—	—	—	Initialized
SAR_2	Initialized	Initialized	—	—	—	—	Initialized
DADRA_1	Initialized	Initialized	—	—	Initialized	Initialized	Initialized
DACR_1	Initialized	Initialized	—	—	Initialized	Initialized	Initialized
DADRB_1	Initialized	Initialized	—	—	Initialized	Initialized	Initialized
DACNT_1	Initialized	Initialized	—	—	Initialized	Initialized	Initialized
CRCCR	Initialized	Initialized	—	—	—	—	Initialized
CRCDIR	Initialized	Initialized	—	—	—	—	Initialized
CRCDOR	Initialized	Initialized	—	—	—	—	Initialized
ICXR_0	Initialized	Initialized	—	—	—	—	Initialized
ICXR_1	Initialized	Initialized	—	—	—	—	Initialized
ICSMBCR	Initialized	Initialized	—	—	—	—	Initialized
ICXR_2	Initialized	Initialized	—	—	—	—	Initialized

ICRB	Initialized	Initialized	—	—	—	—	Initialized
ICRC	Initialized	Initialized	—	—	—	—	Initialized
ISR	Initialized	Initialized	—	—	—	—	Initialized
ISCRH	Initialized	Initialized	—	—	—	—	Initialized
ISCRL	Initialized	Initialized	—	—	—	—	Initialized
DTCERA	Initialized	Initialized	—	—	—	—	Initialized
DTCERB	Initialized	Initialized	—	—	—	—	Initialized
DTCERC	Initialized	Initialized	—	—	—	—	Initialized
DTCERD	Initialized	Initialized	—	—	—	—	Initialized
DTCERE	Initialized	Initialized	—	—	—	—	Initialized
DTVECR	Initialized	Initialized	—	—	—	—	Initialized
ABRKCR	Initialized	Initialized	—	—	—	—	Initialized
BARA	Initialized	Initialized	—	—	—	—	Initialized
BARB	Initialized	Initialized	—	—	—	—	Initialized
BARC	Initialized	Initialized	—	—	—	—	Initialized
IER16	Initialized	Initialized	—	—	—	—	Initialized
ISR16	Initialized	Initialized	—	—	—	—	Initialized
ISCR16H	Initialized	Initialized	—	—	—	—	Initialized
ISCR16L	Initialized	Initialized	—	—	—	—	Initialized
ISSR16	Initialized	Initialized	—	—	—	—	Initialized
ISSR	Initialized	Initialized	—	—	—	—	Initialized
PTCNT0	Initialized	Initialized	—	—	—	—	Initialized
BCR2	Initialized	Initialized	—	—	—	—	Initialized
WSCR2	Initialized	Initialized	—	—	—	—	Initialized

ICSR_1	Initialized	Initialized	—	—	—	—	Initialized
ICDR_1	—	—	—	—	—	—	—
SARX_1	Initialized	Initialized	—	—	—	—	Initialized
ICMR_1	Initialized	Initialized	—	—	—	—	Initialized
SAR_1	Initialized	Initialized	—	—	—	—	Initialized
TIER	Initialized	Initialized	—	—	—	—	Initialized
TCSR	Initialized	Initialized	—	—	—	—	Initialized
FRC	Initialized	Initialized	—	—	—	—	Initialized
OCRA	Initialized	Initialized	—	—	—	—	Initialized
OCRB	Initialized	Initialized	—	—	—	—	Initialized
TCR	Initialized	Initialized	—	—	—	—	Initialized
TOCR	Initialized	Initialized	—	—	—	—	Initialized
OCRAR	Initialized	Initialized	—	—	—	—	Initialized
OCRAF	Initialized	Initialized	—	—	—	—	Initialized
DADRA_0	Initialized	Initialized	—	—	Initialized	Initialized	Initialized
DACR_0	Initialized	Initialized	—	—	Initialized	Initialized	Initialized
DADRB_0	Initialized	Initialized	—	—	Initialized	Initialized	Initialized
DACNT_0	Initialized	Initialized	—	—	Initialized	Initialized	Initialized
TCSR_0	Initialized	Initialized	—	—	—	—	Initialized
TCNT_0	Initialized	Initialized	—	—	—	—	Initialized
PAODR	Initialized	Initialized	—	—	—	—	Initialized
PAPIN	—	—	—	—	—	—	—
PADDR	Initialized	Initialized	—	—	—	—	Initialized
P1PCR	Initialized	Initialized	—	—	—	—	Initialized

P4DDR	Initialized	—	—	—	—	—	Initialized
P3DR	Initialized	Initialized	—	—	—	—	Initialized
P4DR	Initialized	—	—	—	—	—	Initialized
P5DDR	Initialized	Initialized	—	—	—	—	Initialized
P6DDR	Initialized	Initialized	—	—	—	—	Initialized
P5DR	Initialized	Initialized	—	—	—	—	Initialized
P6DR	Initialized	Initialized	—	—	—	—	Initialized
PBODR	Initialized	Initialized	—	—	—	—	Initialized
PBPIN	—	—	—	—	—	—	—
P8DDR	Initialized	Initialized	—	—	—	—	Initialized
P7PIN	—	—	—	—	—	—	—
PBDDR	Initialized	Initialized	—	—	—	—	Initialized
P8DR	Initialized	Initialized	—	—	—	—	Initialized
P9DDR	Initialized	Initialized	—	—	—	—	Initialized
P9DR	Initialized	Initialized	—	—	—	—	Initialized
IER	Initialized	Initialized	—	—	—	—	Initialized
STCR	Initialized	Initialized	—	—	—	—	Initialized
SYSCR	Initialized	Initialized	—	—	—	—	Initialized
MDCR	Initialized	Initialized	—	—	—	—	Initialized
BCR	Initialized	Initialized	—	—	—	—	Initialized
WSCR	Initialized	Initialized	—	—	—	—	Initialized
TCR_0	Initialized	Initialized	—	—	—	—	Initialized
TCR_1	Initialized	Initialized	—	—	—	—	Initialized
TCSR_0	Initialized	Initialized	—	—	—	—	Initialized

ICCR_0	Initialized	Initialized	—	—	—	—	Initialized
ICSR_0	Initialized	Initialized	—	—	—	—	Initialized
ICDR_0	—	—	—	—	—	—	—
SARX_0	Initialized	Initialized	—	—	—	—	Initialized
ICMR_0	Initialized	Initialized	—	—	—	—	Initialized
SAR_0	Initialized	Initialized	—	—	—	—	Initialized
SMR_3	Initialized	Initialized	—	—	—	—	Initialized
BRR_3	Initialized	Initialized	—	—	—	—	Initialized
SCR_3	Initialized	Initialized	—	—	—	—	Initialized
TDR_3	Initialized	Initialized	—	—	Initialized	Initialized	Initialized
SSR_3	Initialized	Initialized	—	—	Initialized	Initialized	Initialized
RDR_3	Initialized	Initialized	—	—	Initialized	Initialized	Initialized
SCMR_3	Initialized	Initialized	—	—	—	—	Initialized
TCSR_1	Initialized	Initialized	—	—	—	—	Initialized
TCNT_1	Initialized	Initialized	—	—	—	—	Initialized
TCR_X	Initialized	Initialized	—	—	—	—	Initialized
TCSR_X	Initialized	Initialized	—	—	—	—	Initialized
TCNT_X	Initialized	Initialized	—	—	—	—	Initialized
TCORA_X	Initialized	Initialized	—	—	—	—	Initialized
TCORB_X	Initialized	Initialized	—	—	—	—	Initialized
TCR_Y	Initialized	Initialized	—	—	—	—	Initialized
TCSR_Y	Initialized	Initialized	—	—	—	—	Initialized
TCORA_Y	Initialized	Initialized	—	—	—	—	Initialized
TCORB_Y	Initialized	Initialized	—	—	—	—	Initialized

Input voltage (pins multiplexed with analog input)	(1)	V_{in}	–0.3 to AVCC + 0.3
Input voltage (pins multiplexed with IIC functions)	(2)	V_{in}	–0.3 to +6.5
Input voltage (pins other than (1) and (2) above)		V_{in}	–0.3 to VCC + 0.3
Reference power supply voltage		AVref	–0.3 to AVCC + 0.3
Analog power supply voltage		AVCC	–0.3 to +4.3
Analog input voltage		V_{AN}	–0.3 to AVCC + 0.3
Operating temperature		T_{opr}	–20 to +75 (regular specifications)
			–40 to +85 (wide temperature specifications)
Operating temperature (when flash memory is programmed or erased)		T_{opr}	0 to +75
Storage temperature		T_{stg}	–55 to +125

Caution: Permanent damage to this LSI may result if absolute maximum ratings are exce

Note: * Voltage applied to the VCC pin.
Make sure power is not applied to the VCL pin.

Item			Symbol	Min.	Typ.	Max.	Unit	Co
Schmitt trigger input voltage	EVENT15 to EVENT0, (Ex)DB7 to (Ex)DB0, $\overline{(Ex)IRQ15}$ to $\overline{(Ex)IRQ0}$, $\overline{ETRST}$, XTAL, EXCL, $\overline{ADTRG}$	(1)	V_T^-	VCC × 0.2	—	—	V	
			V_T^+	—	—	VCC × 0.7		
			$V_T^+ - V_T^-$	VCC × 0.05	—	—		
	SCL5 to SCL0, SDA5 to SDA0		V_T^-	VCC × 0.3	—	—		
			V_T^+	—	—	VCC × 0.7		
			$V_T^+ - V_T^-$	VCC × 0.05	—	—		
Input high voltage	$\overline{RES}$, $\overline{STBY}$, 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, Ports 80 to 83, C0 to C5, D6, D7			—	—	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 pins other than (1) and (3) above			–0.3	—	VCC × 0.2	
Output high voltage	SCL5 to SCL0, SDA5 to SDA0, $\overline{\text{CLKRUN}}$, GA20, $\overline{\text{PME}}$, $\overline{\text{LSMI}}$, LSC2*2	(4)	V_{OH}	—	—	—	
	Ports 80 to 83, C0 to C5, D6, D7*3			0.5	—	—	I H
	SERIRQ, LAD3 to LAD0			VCC × 0.9	—	—	I n
	Output pins other than (4) above			VCC –0.5	—	—	I H
				VCC –1.0	—	—	I
Output low voltage	SCL5 to SCL0, SDA5 to SDA0*2	(5)	V_{OL}	—	—	0.5	I
				—	—	0.4	I
	$\overline{\text{CLKRUN}}$, GA20, $\overline{\text{PME}}$, $\overline{\text{LSMI}}$, LSCI, SERIRQ, LAD3 to LAD0			—	—	VCC × 0.1	I
	Output pins other than (5) above			—	—	0.4	I
	HC7 to HC0			—	—	1.0	I

leakage current (off state)	Ports 8 to F						
Input pull-up MOS current	Ports 1 to 4, 6, A, D5 to D0	$-I_P$	20	—	300		V_{IN} = 0 V
Supply current*4	Normal operation	I_{CC}	—	45	60	mA	f = 34 MHz, high-spe All modules operatin
	Sleep mode		—	35	45		f = 34 MHz
	Standby mode*5		—	40	100	µA	Ta ≤ 50 °C
			—	—	250		50 °C < Ta
Analog power supply current	During A/D conversion	AICC	—	1.0	2.0	mA	
	A/D conversion standby		—	2.5	5.0	µA	
Reference power supply current	During A/D conversion	AI_{ref}	—	0.1	1.0	mA	
	A/D conversion standby		—	0.5	5.0	µA	
Input capacitance	All input pin	C_{in}	—	—	10	pF	V_{in} = 0 V, f = 1 MHz, T_a = 25 °C
RAM standby voltage		V_{RAM}	3.0	—	—	V	
VCC start voltage		VCC_{START}	—	0	0.8	V	
VCC rising edge		SVCC	—	—	20	ms/V	

Notes: 1. Do not leave the AVCC, AVref, and AVSS pins open even if the A/D converter is not use
Even if the A/D converter is not used, apply a value in the range from 3.0 V to 3.6 V to th
and AVref pins by connecting them to the power supply (VCC). The relationship betwee
two pins should be AVref ≤ AVCC.

2. An external pull-up resistor is necessary to provide high-level output from SCL5 to SCL0
SDA0 (ICE bit in ICCR is 1), $\overline{\text{CLKRUN}}$, GA20, $\overline{\text{PME}}$, $\overline{\text{LSMI}}$, and $\overline{\text{LSCI}}$.

3. Ports 80 to 83, C0 to C5, D6, and D7 are NMOS push-pull outputs.
High levels on ports 80 to 83, C0 to C5, D6, and D7 are driven by NMOS. An external pu
resistor is necessary to provide high-level output from these pins when they are used as

	HC7 to HC0		—	—	12
	Other output pins		—	—	1.6
Permissible output low current (total)	Total of HC7 to HC0	ΣI_{OL}	—	—	48
	Total of all output pins, including the above		—	—	90
Permissible output high current (per pin)	All output pins	$-I_{OH}$	—	—	2
Permissible output high current (total)	Total of all output pins	$\Sigma -I_{OH}$	—	—	60

Notes: 1. To protect LSI reliability, do not exceed the output current values in table 26.3.

2. When driving a Darlington transistor or LED, always insert a current-limiting resistor in t line, as show in figures 26.1 and 26.2.

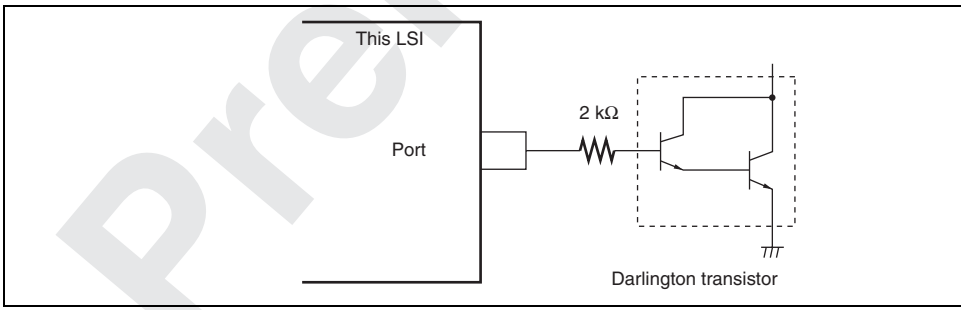


Figure 26.1 Darlington Transistor Drive Circuit (Example)

26.3 AC Characteristics

Figure 26.3 shows the test conditions for the AC characteristics.

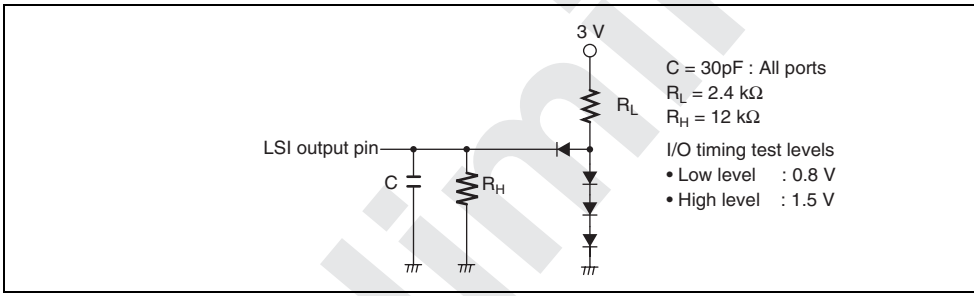


Figure 26.3 Output Load Circuit

26.3.1 Clock Timing

Table 26.4 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
26.6.

Reset oscillation stabilization (crystal)	t_{OSC1}	10	—	ms	Fig
Software standby oscillation stabilization time (crystal)	t_{OSC2}	8	—		Fig

Table 26.5 External Clock Input Conditions

Condition: VCC = 3.0 V to 3.6 V, VSS = 0 V, ϕ = 20 MHz to 34 MHz

Item	Symbol	Min.	Max.	Unit	Te Co
External clock input low level pulse width	t_{EXL}	58.8	—	ns	Fig
External clock input high level pulse width	t_{EXH}	58.8	—	ns	
External clock input rising time	t_{EXr}	—	5	ns	
External clock input falling time	t_{EXf}	—	5	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}	
External clock output stabilization delay time	t_{DEXT}*	500	—	µs	Fig

Note: * t_{DEXT} includes a $\overline{RES}$ pulse width (t_{RESW}).

Subclock input falling time	t_{EXCLf}	—	—	10	ns	
Clock low level pulse width	t_{CL}	0.4	—	0.6	t_{cyc}	Figu
Clock high level pulse width	t_{CH}	0.4	—	0.6	t_{cyc}	

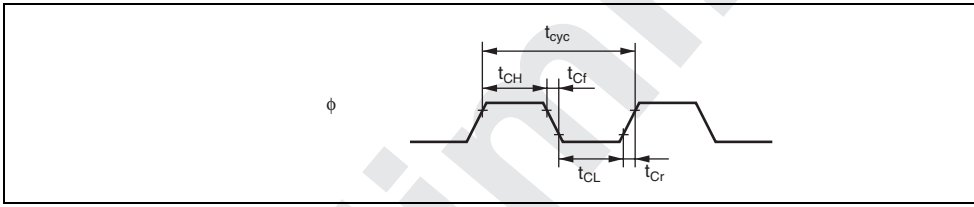


Figure 26.4 System Clock Timing

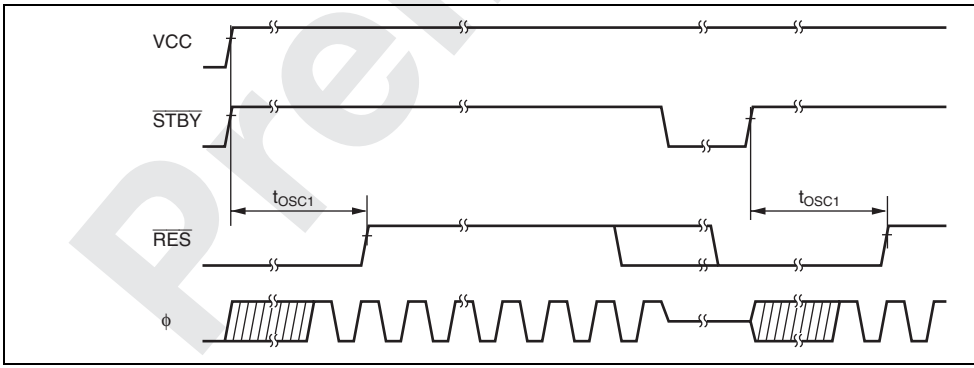


Figure 26.5 Oscillation Stabilization Timing

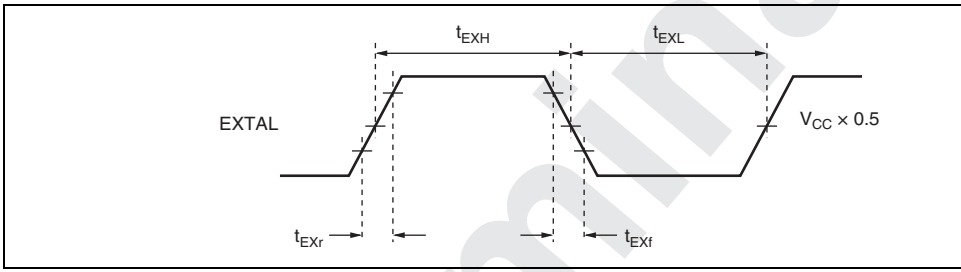
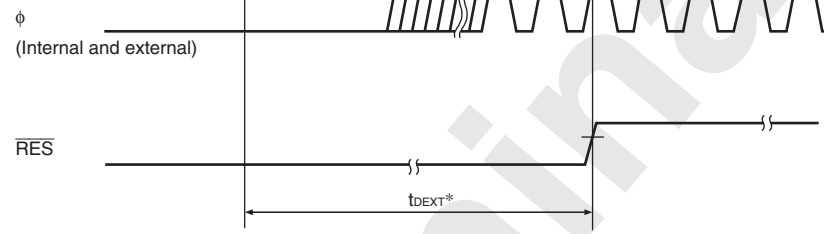


Figure 26.7 External Clock Input Timing



Note: The external clock output stabilization delay time (t_{DEXT}) includes a $\overline{RES}$ pulse width (t_{RESW}).

Figure 26.8 Timing of External Clock Output Stabilization Delay Time

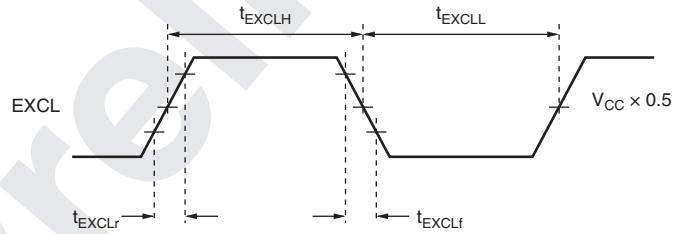


Figure 26.9 Subclock Input Timing

$\overline{RES}$ pulse width	t_{RESW}	20	—	t_{cyc}	
NMI setup time	t_{NMIS}	150	—	ns	Figure 26.11
NMI hold time	t_{NMIH}	10	—		
NMI pulse width (exiting software standby mode)	t_{NMIW}	200	—		
IRQ setup time ($\overline{IRQ15}$ to $\overline{IRQ0}$)	t_{IRQS}	150	—		
IRQ hold time ($\overline{IRQ15}$ to $\overline{IRQ0}$)	t_{IRQH}	10	—		
IRQ pulse width ($\overline{IRQ15}$ to $\overline{IRQ0}$) (exiting software standby mode)	t_{IRQW}	200	—		

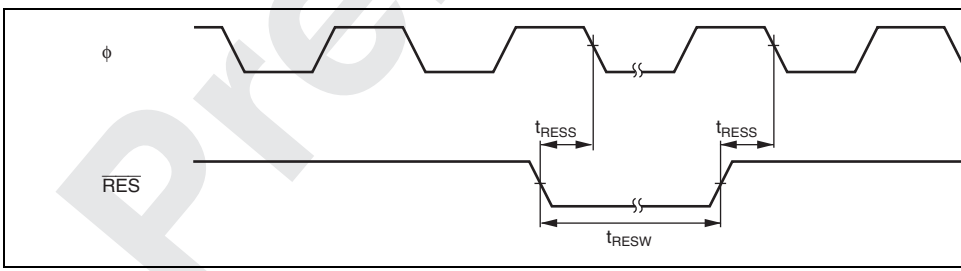


Figure 26.10 Reset Input Timing

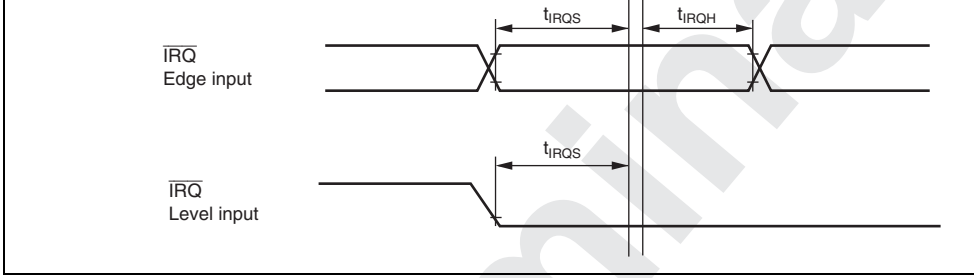
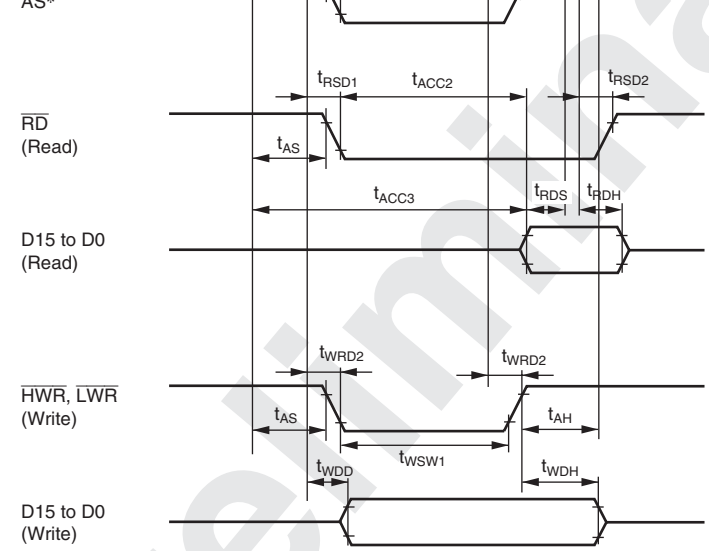


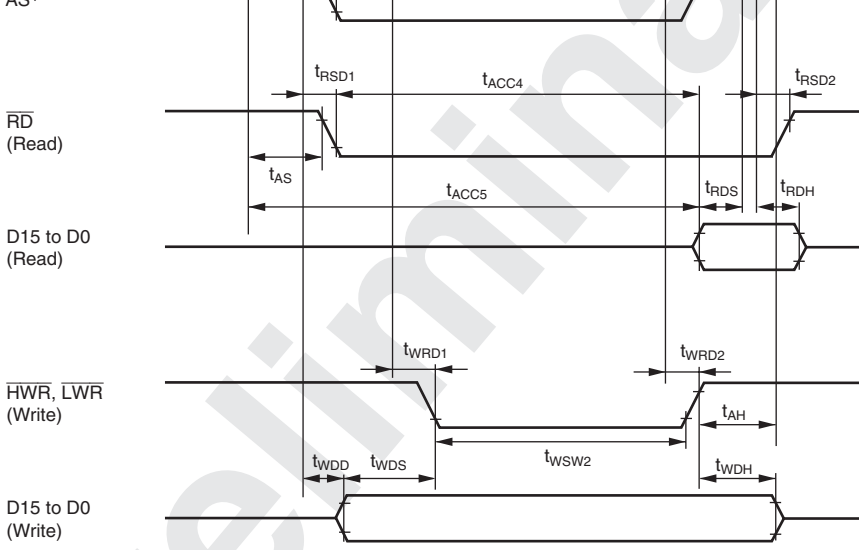
Figure 26.11 Interrupt Input Timing

Address setup time	t_{AS}	$0.5 \times t_{cyc} - 14.7$	—
Address hold time	t_{AH}	$0.5 \times t_{cyc} - 9.7$	—
$\overline{CS}$ delay time ($\overline{IOS}$, $\overline{CS256}$)	t_{CSD}	—	14.7
$\overline{AS}$ delay time	t_{ASD}	—	14.7
$\overline{HBE}$ delay time	t_{HBD}	—	t_{AD} +5.0
$\overline{LBE}$ delay time	t_{LBD}	—	t_{AD} +5.0
$\overline{RD}$ delay time 1	t_{RSD1}	—	14.7
$\overline{RD}$ delay time 2	t_{RSD2}	—	14.7
Read data setup time	t_{RDS}	14.7	—
Read data hold time	t_{RDH}	0	—
Read data access time 1	t_{ACC1}	—	$1.0 \times t_{cyc} - 29.4$
Read data access time 2	t_{ACC2}	—	$1.5 \times t_{cyc} - 24.7$
Read data access time 3	t_{ACC3}	—	$2.0 \times t_{cyc} - 29.4$
Read data access time 4	t_{ACC4}	—	$2.5 \times t_{cyc} - 24.7$
Read data access time 5	t_{ACC5}	—	$3.0 \times t_{cyc} - 29.4$
$\overline{WR}$ delay time 1	t_{WRD1}	—	14.7
$\overline{WR}$ delay time 2	t_{WRD2}	—	14.7
$\overline{WR}$ pulse width 1	t_{WSW1}	$1.0 \times t_{cyc} - 19.6$	—
$\overline{WR}$ pulse width 2	t_{WSW2}	$1.5 \times t_{cyc} - 19.6$	—
Write data delay time	t_{WDD}	—	24.7
Write data setup time	t_{WDS}	0	—
Write data hold time	t_{WDH}	$0.5 \times t_{cyc} - 5$	—
$\overline{WAIT}$ setup time	t_{WTS}	24.7	—
$\overline{WAIT}$ hold time	t_{WTH}	5	—



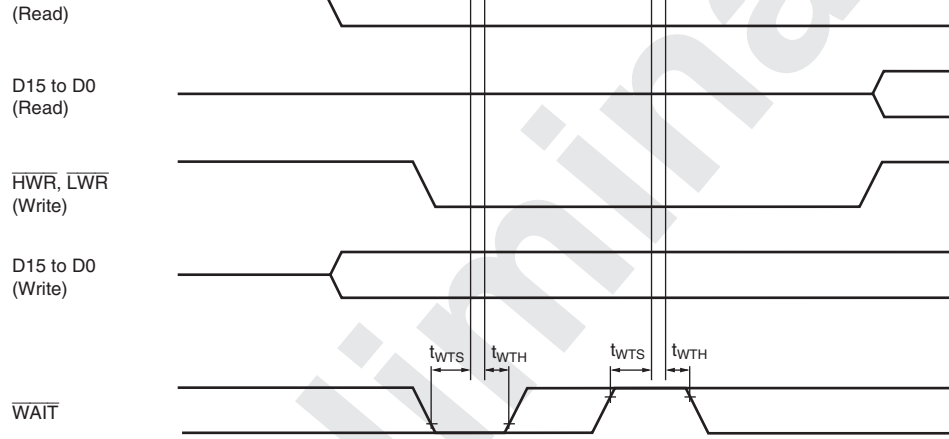
Note: * AS is multiplexed with IOS. Either the AS or IOS function can be selected by the IOSE bit of SYSC

Figure 26.12 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 26.13 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 26.14 Basic Bus Timing/3-State Access with One Wait State

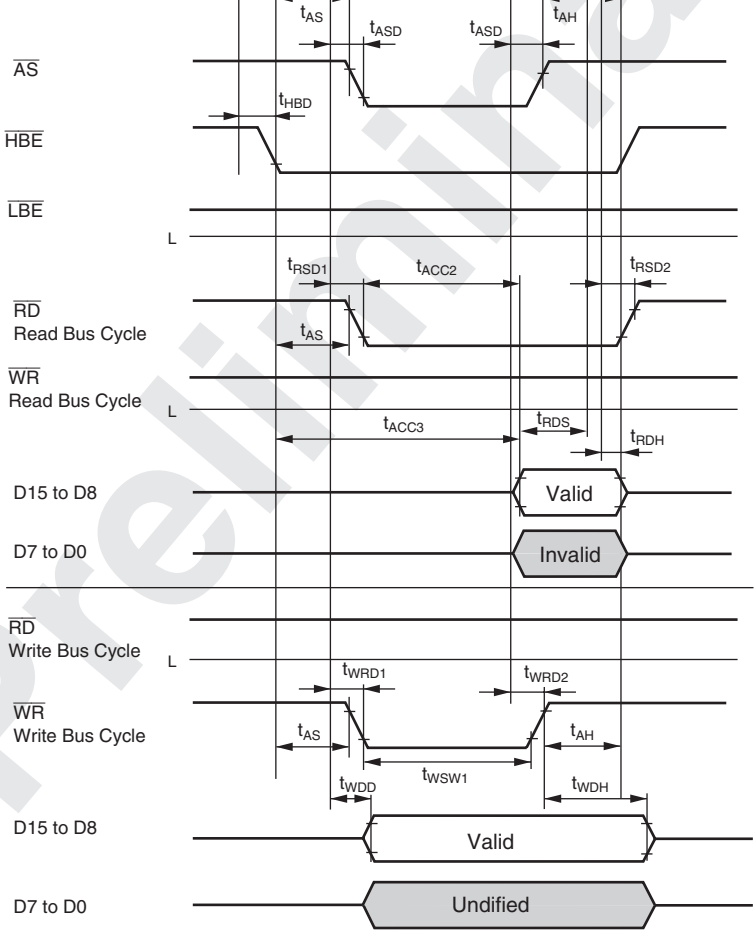


Figure 26.15 Even Byte Access (ADMXE = 0)

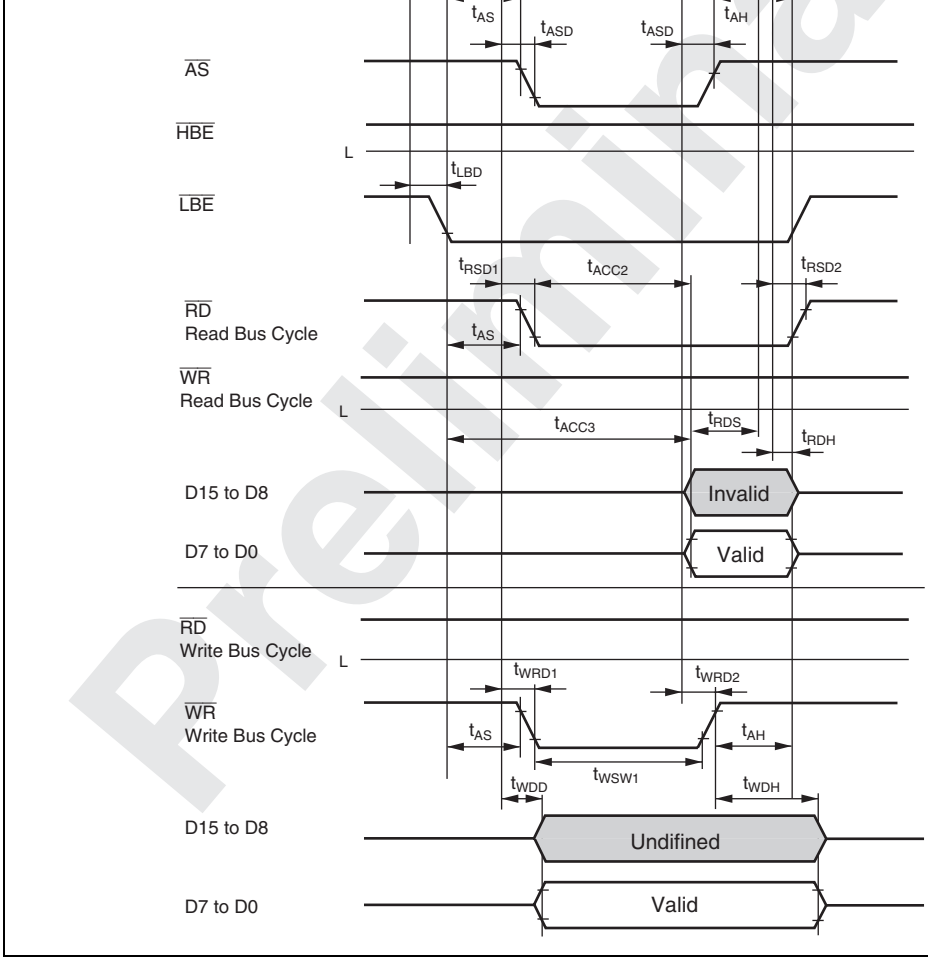


Figure 26.16 Odd Byte Access (ADMXE = 0)

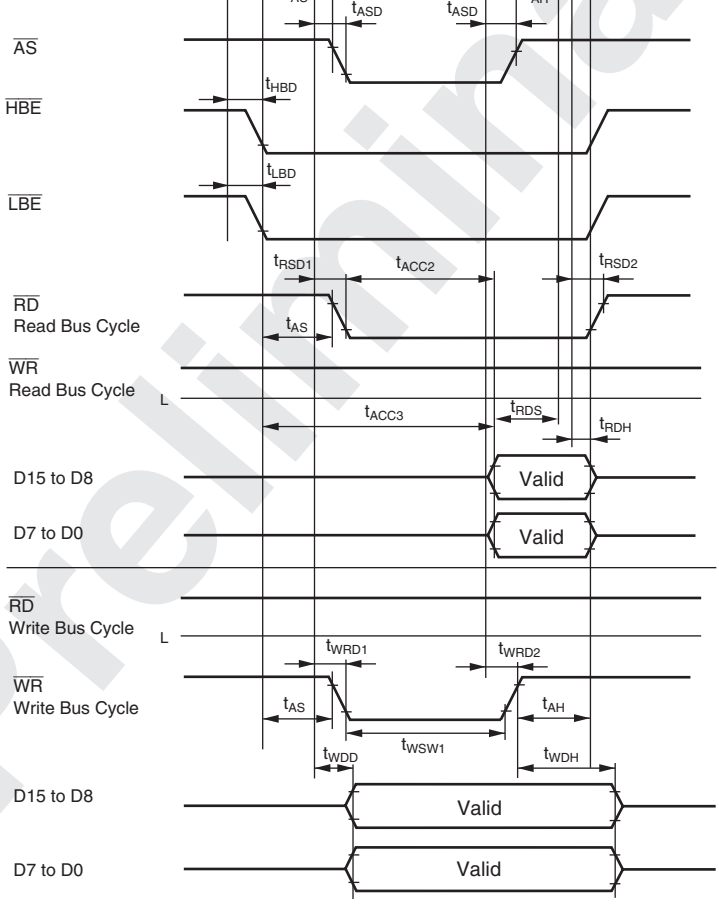
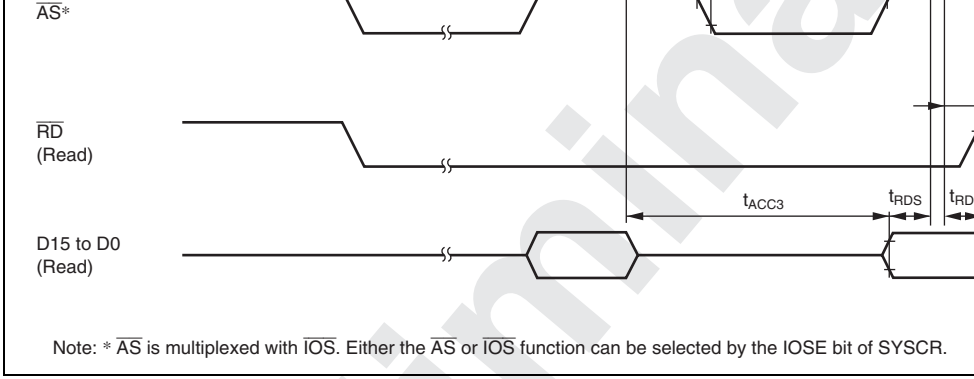
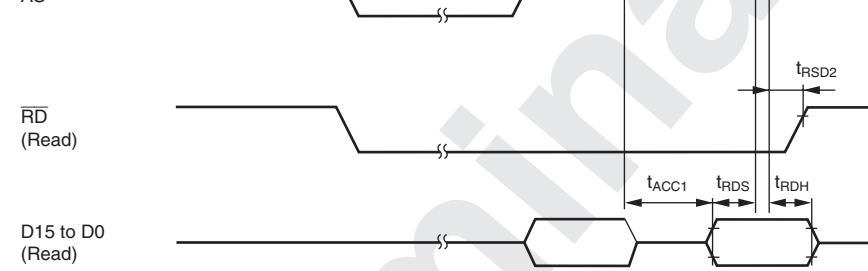


Figure 26.17 Word Access (ADMXE = 0)



Note: * AS is multiplexed with IOS. Either the AS or IOS function can be selected by the IOSE bit of SYSCR.

Figure 26.18 Burst ROM Access Timing/2-State Access



Note: * AS is multiplexed with IOS. Either the AS or IOS function can be selected by the IOSE bit of SYS

Figure 26.19 Burst ROM Access Timing/1-State Access

Address setup time 2	t_{AS2}	$0.5 \times t_{cyc} - 14.7$	—	26.21
Address hold time 2	t_{AH2}	$0.5 \times t_{cyc} - 9.7$	—	
$\overline{CS}$ delay time ($\overline{IOS}$, $\overline{CS256}$)	t_{CSD}	—	14.7	
$\overline{AH}$ delay time	t_{AHD}	—	14.7	
$\overline{RD}$ delay time 1	t_{RSD1}	—	14.7	
$\overline{RD}$ delay time 2	t_{RSD2}	—	14.7	
Read data setup time	t_{RDS}	14.7	—	
Read data hold time	t_{RDH}	0	—	
Read data access time 2	t_{ACC2}	—	$1.5 \times t_{cyc} - 24.4$	
Read data access time 4	t_{ACC4}	—	$2.5 \times t_{cyc} - 24.4$	
Read data access time 6	t_{ACC6}	—	$3.5 \times t_{cyc} - 24.4$	
Read data access time 7	t_{ACC7}	—	$4.5 \times t_{cyc} - 24.4$	
$\overline{WR}$ delay time 1	t_{WRD1}	—	14.7	
$\overline{WR}$ delay time 2	t_{WRD2}	—	14.7	
$\overline{WR}$ pulse width time 1	t_{WSW1}	$1.0 \times t_{cyc} - 19.6$	—	
$\overline{WR}$ pulse width time 2	t_{WSW2}	$1.5 \times t_{cyc} - 19.6$	—	
Write data delay time	t_{WDD}	—	24.4	
Write data setup time	t_{WDS}	0	—	
Write data hold time	t_{WDH}	$0.5 \times t_{cyc} - 5$	—	

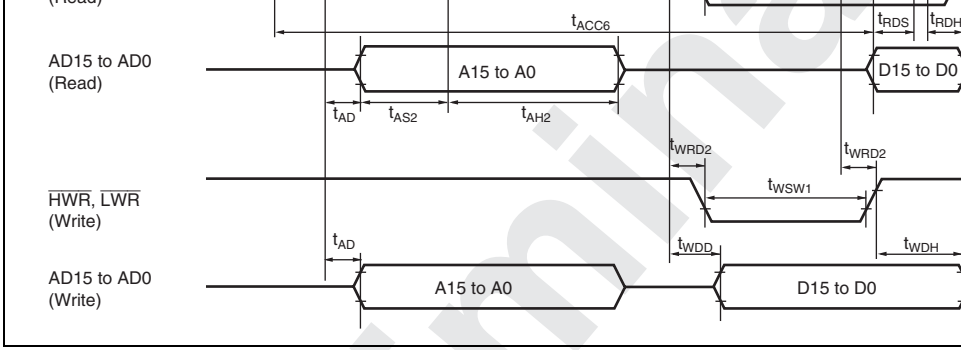


Figure 26.20 Multiplex Bus Timing/Data 2-State Access

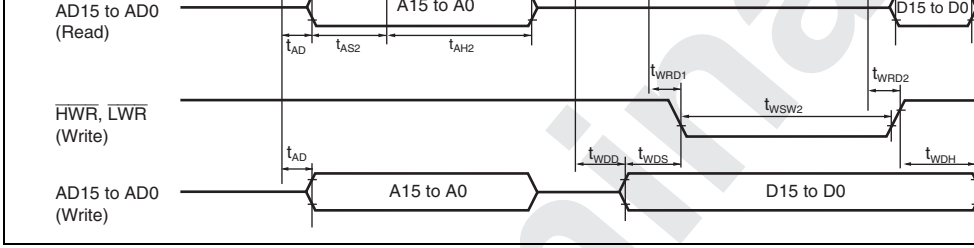


Figure 26.21 Multiplex Bus Timing/Data 3-State Access

I/O ports	Output data delay time		t_{PWD}	—	29.4	ns	Figure 26.2
	Input data setup time		t_{PRS}	19.6	—		
	Input data hold time		t_{PRH}	19.6	—		
PWMX	Timer output delay time		t_{PWOD}	—	29.4	ns	Figure 26.2
SCI	Input clock cycle	Asynchronous	t_{Scyc}	4	—	t_{cyc}	Figure 26.2
		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}	—	29.4	ns	Figure 26.2
	Receive data setup time (synchronous)		t_{RXS}	19.6	—		
	Receive data hold time (synchronous)		t_{RXH}	19.6	—		
A/D converter	Trigger input setup time		t_{TRGS}	19.6	—	ns	Figure 26.2
WDT	$\overline{\text{RESO}}$ output delay time		t_{RESD}	—	50	ns	Figure 26.2
	$\overline{\text{RESO}}$ output pulse width		t_{RESOW}	132	—	t_{cyc}	

Note: * Only the peripheral modules that can be used in subclock operation.

Figure 26.22 I/O Port Input/Output Timing

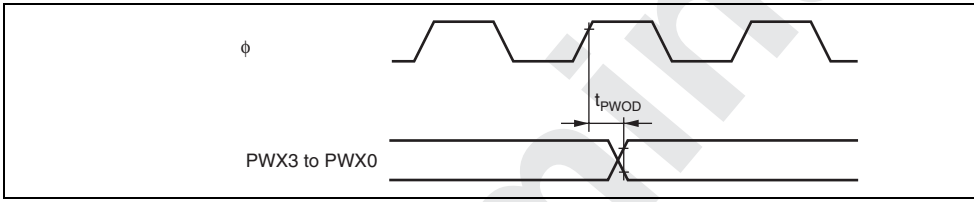


Figure 26.23 PWMX Output Timing

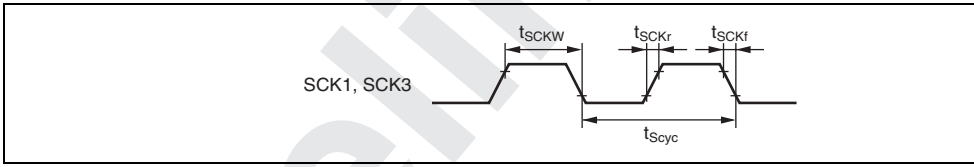


Figure 26.24 SCK Clock Input Timing

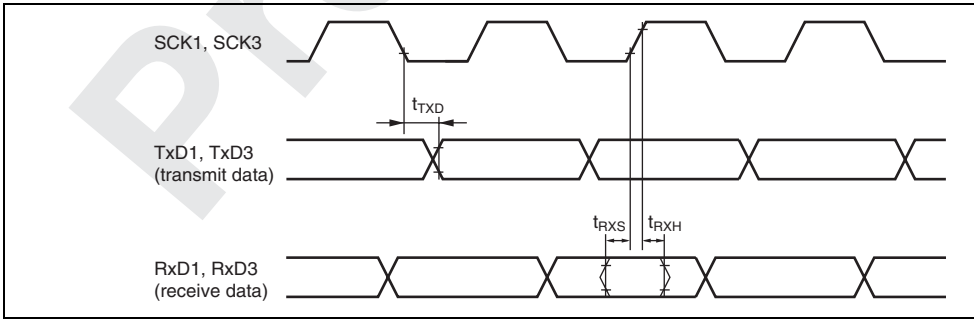


Figure 26.25 SCI Input/Output Timing (Clock Synchronous Mode)

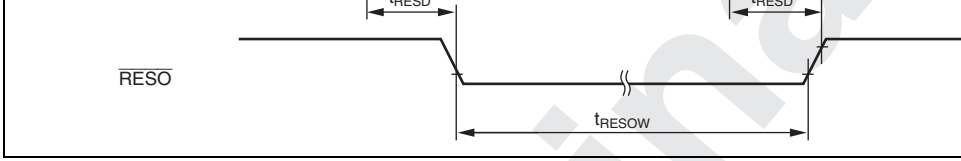
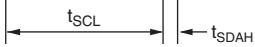


Figure 26.27 WDT Output Timing ($\overline{\text{RESO}}$)

SCL, SDA output fall time	t_{Of}	20 + 0.1 C_b	—	250	
SCL, SDA input spike pulse elimination time	t_{SP}	—	—	1	t_{cyc}
SDA input bus free time	t_{BUF}	5	—	—	
Start condition input hold time	t_{STAH}	3	—	—	
Repeated start condition input setup time	t_{STAS}	3	—	—	
Stop condition input setup time	t_{STOS}	3	—	—	
Data input setup time	t_{SDAS}	0.5	—	—	
Data input hold time	t_{SDAH}	0	—	—	ns
SCL, SDA capacitive load	C_b	—	—	400	pF

Note: * 17.5 t_{cyc} or 37.5 t_{cyc} can be set according to the clock selected for use by the IIC



Note: * S, P, and Sr indicate the following conditions:
S: Start condition
P: Stop condition
Sr: Repeated start condition

Figure 26.28 I²C Bus Interface Input/Output Timing

Table 26.12 LPC Module Timing

Conditions: VCC = 3.0 V to 3.6V, VSS = 0 V, ϕ = 20 MHz to 34 MHz

Item	Symbol	Min.	Typ.	Max.	Unit	Test Condition
Input clock cycle	t_{Lcyc}	30	—	—	ns	Figure 26.29
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	—	—		

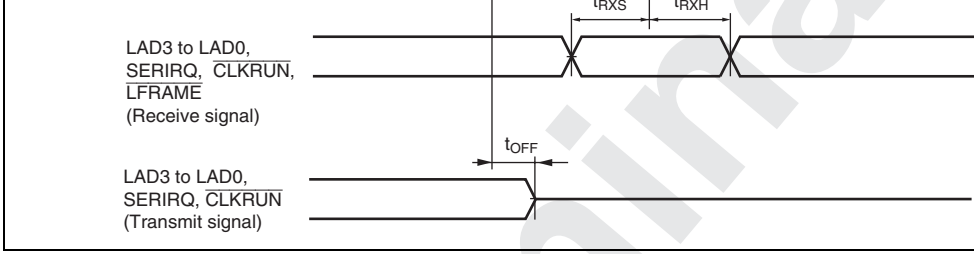


Figure 26.29 LPC Interface (LPC) Timing

ETRST pulse width	t_{TRSTW}	20	—	t_{cyc}	Figure 26.3
Reset hold transition pulse width	t_{RSTHW}	3	—		
ETMS setup time	t_{TMSS}	20	—	ns	Figure 26.3
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 $t_{cyc} \leq t_{TCKcyc}$

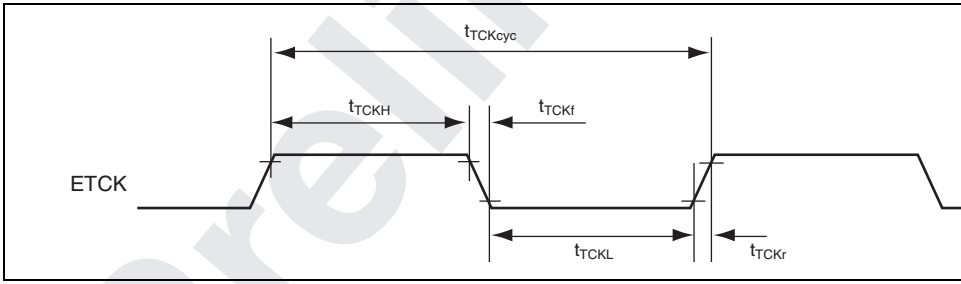


Figure 26.30 JTAG ETCK Timing

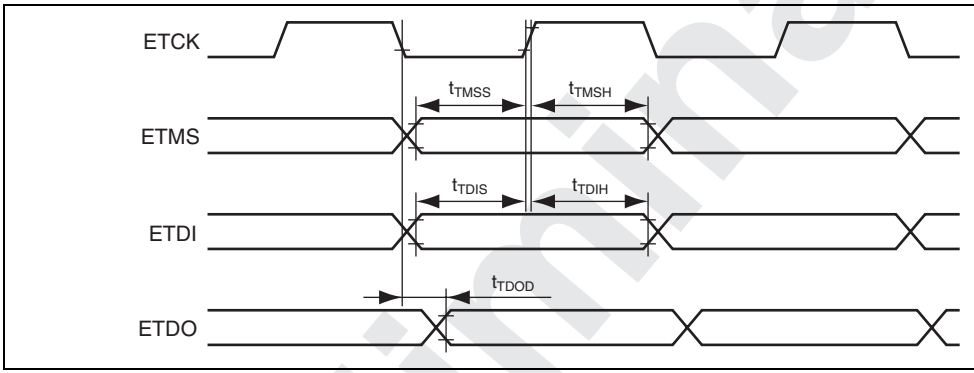


Figure 26.32 JTAG Input/Output Timing

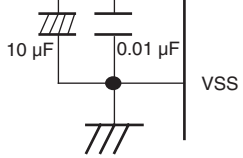
VSS = AVSS = 0 V, φ = 20 MHz to 34 MHz

	Condition A			Condition B			
Item	**Min.**	**Typ.**	**Max.**	**Min.**	**Typ.**	**Max.**	**Unit**
Resolution		10			10		Bits
Conversion time	—	—	4.0*[1]	—	—	4.7*[2]	µs
Analog input capacitance	—	—	20	—	—	20	pF
Permissible signal-source impedance	—	—	5	—	—	5	kΩ
Nonlinearity error	—	—	±7.0	—	—	±7.0	LSB
Offset error	—	—	±7.5	—	—	±7.5	
Full-scale error	—	—	±7.5	—	—	±7.5	
Quantization error	—	—	±0.5	—	—	±0.5	
Absolute accuracy	—	—	±8.0	—	—	±8.0	

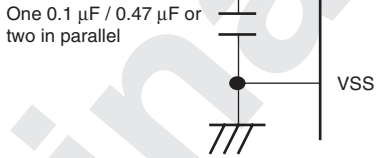
Notes: 1. Value when using the maximum operating frequency in single mode of 80 sta
2. Value when using the maximum operating frequency in single mode of 160 st

Item	Symbol	Min.	Typ.	Max.	Unit	Test Condi
Programming time*1*2*4	t_P	—	1	10	ms/128 bytes	
Erase time*1*2*4	t_E	—	40	130	ms/4-kbyte block	
		—	300	800	ms/32-kbyte block	
		—	600	1500	ms/64-kbyte block	
Programming time (total)*1*2*4	Σt_P	—	9.2	24	s/512 kbytes	Ta =
Erase time (total)*1*2*4	Σt_E	—	9.2	24	—	—
Programming and Erase time (total)*1*2*4	Σt_{PE}	—	18.4	48	—	—
Reprogramming count*5	N_{WEC}	100*3	1000	—	Times	
Data retention time*4	t_{DRP}	10	—	—	Years	

Notes: 1. Programming and erase time depends on the data.
2. Programming and erase time do not include data transfer time.
3. This value indicates the minimum number of which the flash memory are reprogrammed with all characteristics guaranteed. (The guaranteed value rang 1 to the minimum number.)
4. This value indicates the characteristics while the flash memory is reprogramme the specified range (including the minimum number).
5. Reprogramming count in each erase block.



It is recommended that a bypass capacitor be connected to the VCC pin. (The values are reference values.)
When connecting, place a bypass capacitor near the pin.



Do not connect Vcc power supply to the VCL
Always connect a capacitor for internal step-
stabilization.
Use one or two ceramic multilayer capacitor
(0.1 µF / 0.47 µF: connect in parallel when u
and place it (them) near the pin.

Figure 26.33 Connecting Capacitors to VCC and VCL Pins

A7 to A0	1 (DDR=1)			kept*	kept*	Address
Port 2	0 / 1 (DDR=0)	T	T	kept	kept	I/O port
A15 to A8	1 (DDR=1)			kept*	kept*	Address
Port 3	0	T	T	kept	kept	I/O port
D15 to D8	1			T	T	D15 to D
Port 4	0	T	T	kept	kept	I/O port
Port 5	X	T	T	kept	kept	I/O port
Port 6	0 / 1 (8 bits)	T	T	kept	kept	I/O port
D7 to D0	1 (16 bits)			T	T	D7 to D0
Port 7	X	T	T	T	T	Input por
Port 8	X	T	T	kept	kept	I/O port
Port 97	0	T	T	kept	kept	I/O port
$\overline{\text{WAIT}}$	1 (CS256E=0)			T	T	$\overline{\text{WAIT}}$
$\overline{\text{CS256}}$	1 (CS256E=1)			H	H	$\overline{\text{CS256}}$
Port 96	0	T	T	T	T	Input por
EXCL	1 (DDR=0)					EXCL
ϕ	1 (DDR=1)			H	ϕ output	ϕ
Port 95	0	T	T	kept	kept	I/O port
$\overline{\text{AS}}$, $\overline{\text{IOS}}$	1			H	H	$\overline{\text{AS}}$/$\overline{\text{IOS}}$
Port 94	0	T	T	kept	kept	I/O port
$\overline{\text{WR}}$, $\overline{\text{HWR}}$	1			H	H	$\overline{\text{WR}}$, $\overline{\text{HW}}$
Port 93	0	T	T	kept	kept	I/O port
$\overline{\text{RD}}$	1			H	H	$\overline{\text{RD}}$

A2	(address 18=1)					
A23 to A18	1 (address 18=0)			kept*	kept*	A23 to A1
Port A1, A0	0 / 1 (address 13=1)	T	T	kept	kept	I/O port
A17, A16	1 (address 13=0)			kept*	kept*	A17, A16
Port B	X	T	T	kept	kept	I/O port
Port C	X	T	T	kept	kept	I/O port
Port D	X	T	T	kept	kept	I/O port
Port E	X	T	T	kept	kept	I/O port
Port F	X	T	T	kept	kept	I/O port

Legend

H: High level

L: Low level

T: High impedance

x: Don’t care

kept: Input port pins are in the high-impedance state (when DDR = 0 and PCR = 1, the i
up MOS remains on).

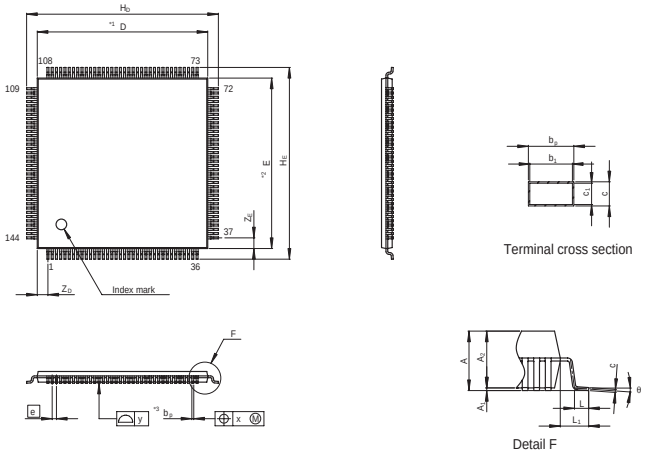
Output port pins retain their states.

Functions of some pins will be changed to the I/O port function, which is determine
DDR and DR, because the on-chip peripheral module associated with that pin func
initialized.

DDR: Data direction register

Note: * In the case of address output, the last address accessed is retained.

JEITA Package Code	RENESAS Code	Previous Code	MASS[Typ.]
P-TQFP144-16x16-0.40	PTQP0144LC-A	TFP-144/TFP-144V	0.6g



NOTE)
1. DIMENSIONS
DO NOT INCL
2. DIMENSION
INCLUDE TRI

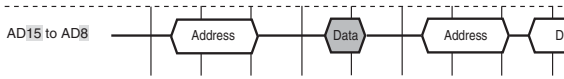
Reference Symbol	D
D	
E	
A_2	
H_D	1
H_E	1
A	
A_1	0
b_p	0
b_1	
c	0
c_1	
θ	
e	
x	
y	
Z_D	
Z_E	
L	
L_1	

Figure C.1 Package Dimensions (TQFP-144)

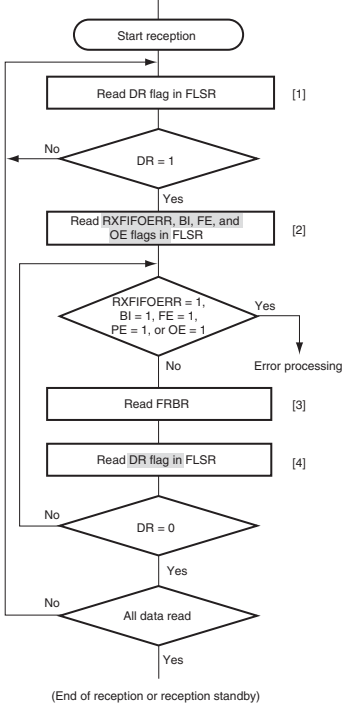
					Single-chip mode
4	0	0	0	—	Flash programming/
6	0	1	0	Emulation	On-chip emulation m

3.3.1 Mode 2	60	Description amended • Multiplex extended mode When 8-bit bus is specified, port 2 functions as t address output and data input/output regardless setting of the data direction register (DDR). Port used as a general port.
5.5 Interrupt Exception Handling Vector Table Table 5.3 Interrupt Sources, Vector Addresses, and Interrupt Priorities	83	Note added Note: Vector numbers not listed above are reserve system.
6.5.1 Data Size and Data Alignment (1) 8-Bit Access Space	119	Description amended The upper data bus (AD15 to AD8) is used in addre multiplex extended mode.
6.5.2 Valid Strobes Table 6.13 Data Buses Used and Valid Strobes	121	Table amended

Area	Access Size	Read/ Write	Address	Valid Strobe	Upper Data Bus (D15 to D8/ AD15 to AD8)
8-bit access space	Byte	Read	—	$\overline{RD}$	Valid
		Write	—	$\overline{HWR}$	
8-bit access space (in address-data multiplex extended mode)	Byte	Read	—	$\overline{RD}$	Valid
		Write	—	$\overline{HWR}$	

Figure 6.17 Bus Timing for 8-Bit, 2-State Access Space	135	Figure amended 
(2) 8-Bit, 3-State Data Access Space		Description amended … When an 8-bit access space is accessed, the upp (AD15 to AD8) of the data bus is used.
Figure 6.18 Bus Timing for 8-Bit, 3-State Access Space		Figure amended 
Section 7 Data Transfer Controller (DTC)	151 to 178	Description amended normal mode → normal transfer mode repeat mode → repeat transfer mode

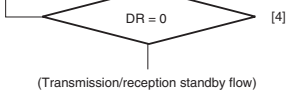
		CRAL is decremented by 1 every time data is transf the contents of CRAH are transferred when the cou reaches H'00. The number of times data is transferr when CRAH = CRAL = H'01, 255 when CRAH = CR and 256 when CRAH = CRAL = H'00. In block transfer mode CRA is divided in two, with th eight bits designated as CRAH and the lowest eight CRAL. CRAH holds the value for the block size, and functions as an 8-bit block size counter (1 to 256). C decremented by 1 every time data is transferred, an contents of CRAH are transferred when the counter reaches H'00. The block size is one byte (or one wo CRAH = CRAL = H'01, 255 bytes (or 255 words) wh = CRAL = H'FF, and 256 bytes (or 256 words) when CRAL = H'00.
7.5 Location of Register Information and DTC Vector Table Figure 7.4 Correspondence between DTC Vector Address and Register Information	166	Newly added
10.1 Features • Special functions provided by automatic addition function	271	Description deleted



[2] Read the RXFIFOERR, BI, FE, PE
FLSR to ensure that no error has
error has occurred, perform error p
the OUT2 bit in FMCR and the EL
set to 1, a receive line status inter

[3] Read the receive data in FRBR.

[4] Check the DR flag in FLSR. When
cleared to 0 and all data has been
is complete.



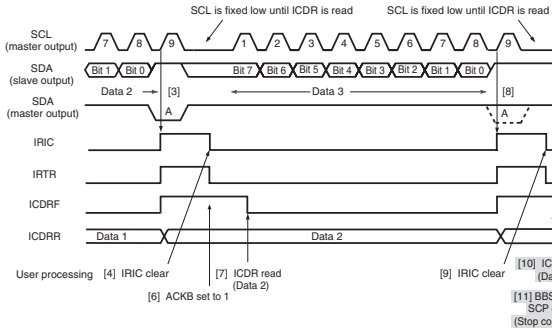
17.4.4 Master Receive Operation

Figure 17.12 Stop Condition Issuance Timing Example in Master Receive Mode

(MLS = WAIT = 0, HNDS = 1)

490

Figure amended



17.6 Usage Notes

Table 17.13 I^2C Bus Timing (with Maximum Influence of t_{Sr}/t_{Sf})

518

Table amended

			Time Indication (at Maximum Transfer Rate) [ns]			
Item	t_{cyc} Indication		t_{Sr}/t_{Sf} Influence (Max.)	I^2C Bus Specification (Min.)	ϕ = 20 MHz	ϕ = 2
t_{SCLLO}	0.5 t_{SCLO} (-t_{Sf})	Standard mode	-250	4700	4750	4230
		High-speed mode	-250	1300	950*1	870*
t_{BUFO}	0.5 t_{SCLO} -1 t_{cyc} (-t_{Sr})	Standard mode	-1000	4700	3950*1	3440
		High-speed mode	-300	1300	850*1	780*

					1: LPCPD Pin state is high level
2	PME	Undefined	R	—	0: PME Pin state is low level 1: PME Pin state is high level
1	LSMI	Undefined	R	—	0: LSMI Pin state is low level 1: LSMI Pin state is high level
0	LSCI	Undefined	R	—	0: LSCI Pin state is low level 1: LSCI Pin state is high level

24.2 Mode Transitions and LSI States Figure 24.1 Mode Transition Diagram	777	Note amended * NMI, IRQ0 to IRQ15
25.2 Register Bits	798	Table amended

Register Abbreviation	Bit 7	Bit 6	Bit 5	Bit 4	Bit 3	Bit 2	Bit 1
SUBMSTPBH	SMSTPB15	SMSTPB14	SMSTPB13	SMSTPB12	SMSTPB11	SMSTPB10	SMSTPB9
SUBMSTPBL	SMSTPB7	SMSTPB6	SMSTPB5	SMSTPB4	SMSTPB3	SMSTPB2	SMSTPB1

26.2 DC Characteristics Table 26.2 DC Characteristics (1)	818	Table amended

Item	
Input high voltage	RES, STBY, NMI, FWE, MD2, MD1, MD0
	EXTAL
	Port 7
	SCL5 to SCL0, SDA5 to SDA0, Ports 80 to 83, C0 to C5, D6, D7

B. Product Lineup	855	Table amended	

Product	Type	Type Code	Mark Code	P
H8S/2164	F-ZTAT version (regular specifications)	R4F2164	F2164VTE34V	1 (T
H8S/2164	F-ZTAT version (wide temperature specifications)	R4F2164	F2164VTE34DV	1 (T

A

B

C

D

F

G

H

L

M

W

Renesas 16-Bit Single-Chip Microcomputer
Hardware Manual
H8S/2164 Group

Publication Date:	Rev.1.00, March 17, 2008
	Rev.2.00, September 28, 2009
Published by:	Sales Strategic Planning Div.
	Renesas Technology Corp.
Edited by:	Customer Support Department
	Global Strategic Communication Div.
	Renesas Solutions Corp.

RENESAS SALES OFFICES

Refer to "**http://www.renesas.com/en/network**" for the latest and detailed information.

Renesas Technology America, Inc.
450 Holger Way, San Jose, CA 95134-1368, U.S.A
Tel: <1> (408) 382-7500, Fax: <1> (408) 382-7501

Renesas Technology Europe Limited
Dukes Meadow, Millboard Road, Bourne End, Buckinghamshire, SL8 5FH, U.K.
Tel: <44> (1628) 585-100, Fax: <44> (1628) 585-900

Renesas Technology (Shanghai) Co., Ltd.
Unit 204, 205, AZIACenter, No.1233 Lujiazui Ring Rd, Pudong District, Shanghai, China 200120
Tel: <86> (21) 5877-1818, Fax: <86> (21) 6887-7858/7898

Renesas Technology Hong Kong Ltd.
7th Floor, North Tower, World Finance Centre, Harbour City, Canton Road, Tsimshatsui, Kowloon, Hong Kong
Tel: <852> 2265-6688, Fax: <852> 2377-3473

Renesas Technology Taiwan Co., Ltd.
10th Floor, No.99, Fushing North Road, Taipei, Taiwan
Tel: <886> (2) 2715-2888, Fax: <886> (2) 3518-3399

Renesas Technology Singapore Pte. Ltd.
1 Harbour Front Avenue, #06-10, Keppel Bay Tower, Singapore 098632
Tel: <65> 6213-0200, Fax: <65> 6278-8001

Renesas Technology Korea Co., Ltd.
Kukje Center Bldg. 18th Fl., 191, 2-ka, Hangang-ro, Yongsan-ku, Seoul 140-702, Korea
Tel: <82> (2) 796-3115, Fax: <82> (2) 796-2145

Renesas Technology Malaysia Sdn. Bhd
Unit 906, Block B, Menara Amcorp, Amcorp Trade Centre, No.18, Jln Persiaran Barat, 46050 Petaling Jaya, Selangor Darul Ehsan, Ma
Tel: <603> 7955-9390, Fax: <603> 7955-9510

H8S/2164 Group
Hardware Manual

Renesas Electronics Corporation
1753, Shimonumabe, Nakahara-ku, Kawasaki-shi, Kanagawa 211-8668 Japan

REJ09B0429-0200

X-ON Electronics

Largest Supplier of Electrical and Electronic Components

Click to view similar products for 16-bit Microcontrollers - MCU ***category:***

Click to view products by Renesas ***manufacturer:***

Other Similar products are found below :

M30302FCPFP#U3 MB90F036APMC-GSE1 MB90F428GCPFR-GSE1 MB96F683RBPMC-GSAE1 R5F10MMGDFB#30 R5F111PGGFB#30 R5F117BCGNA#20 DF3026XBL25V DF36014GFTV DF36014GFXV DF36034GFPV R5F11B7EANA#U0 R5F21172DSP#U0 MB90092PF-G-BNDE1 MB90F335APMC1-G-SPE1 MB90F345CAPFR-GSE1 MB90F568PMCR-GE1 MB96F395RSAPMC-GSE2 DF36024GFXV UPD78F1018F1-BA4-A MB96F018RBPMC-GSE1 MB90F867ASPFR-GE1 M30290FCHP#U3A DF2239FA20IV R5F117BCGFP#30 LC88F58B0AU-SQFPH MB90F548GPF-GE1 MB90214PF-GT-310-BND-AE1 MB90F342CESPQC-GSE2 MB90F428GAPF-GSE1 ML620Q504H-NNNTBWBX S912ZVH128F2VLL UPD78F1500AGK-GAK-AX HD64F3337SF16V MB90F428GCPF-GSE1 MB90F342ESPMC-G-JNE1 MB90022PF-GS-358E1 MB96F395RWAPMC-GSE2 MB96395RSAPMC-GS-110E2 MB90F883CSPMC-GE1 S912ZVHY64F1CLL S912ZVHY64F1VLQ ST10F280 MB96F338RSAPMCR-GK5E2 CY90096PF-G-002-BND-ERE1 ML62Q1569-NNNGAZ0AX ML62Q1739-NNNGAZ0AX ML62Q1749-NNNGAZ0AX ML62Q1579-NNNGAZ0AX ML62Q1559-NNNGAZ0AX