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.

RENESAS

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/2153 Group

Hardware Manual

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

H8S/2153 R4F2153

Renesas Electronics
www.renesas.com

Rev.3.00 2009.09

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 pr
   wait until the target clock signal is stable.

5. Differences between Products

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

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

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

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

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

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

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

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

11. Index

Objective: This manual was written to explain the hardware functions and electrical characteristics of the H8S/2153 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 11 8-Bit Timer (TMR)

Section 12 Watchdog Timer (WDT)

Section 13 Serial Communication Interface (SCI)

Section 15 I^2C Bus Interface (IIC)

Section 16 LPC Interface (LPC)

Section 17 A/D Converter

Section 24 Electrical Characteristics

Appendix

Section 15 I^2C Bus Interface (IIC)

Section 17 A/D Converter

Section 19 Flash Memory

Section 24 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 clocked synchronous serial communication interface (SCI)
  CRC operation circuit (CRC)
  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	R4F2153	512 Kbytes	40 Kbytes	

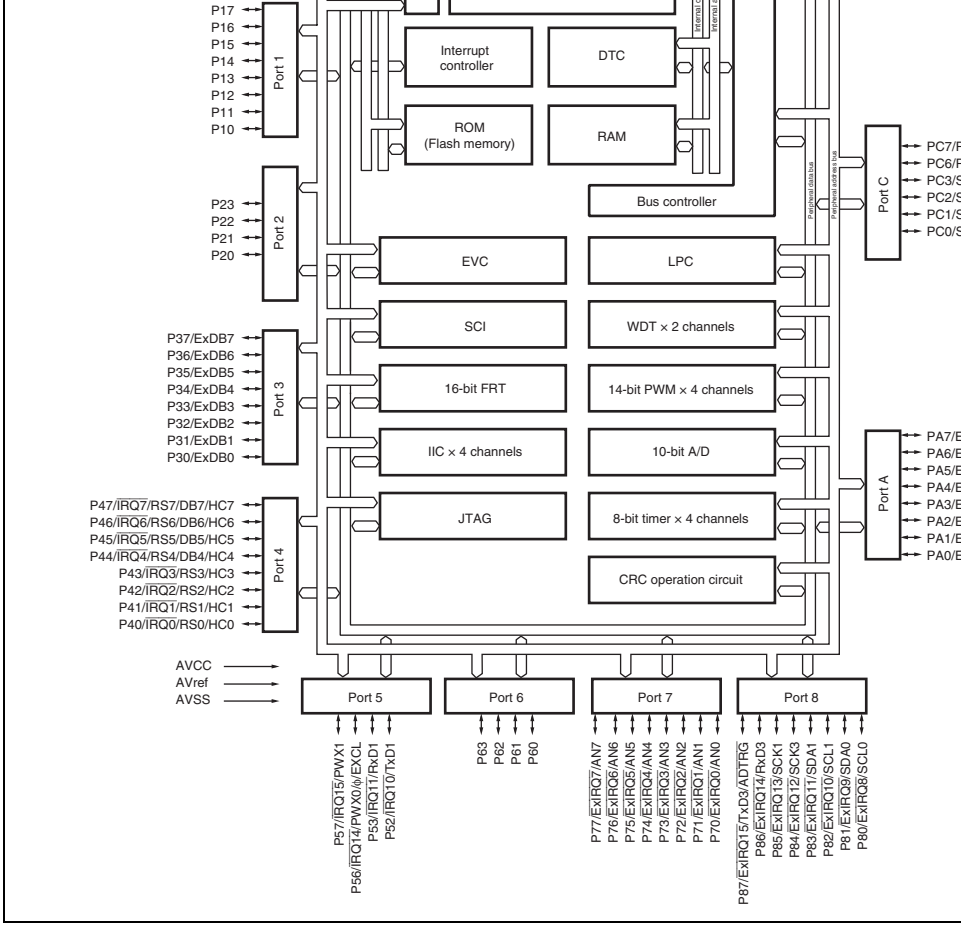


Figure 1.1 Internal Block Diagram

9	P30	P10	NC	NC	P22	ETDO	VCC	P60	NC	P72	P71
8	P33	P32	P31	VSS	P20	$\overline{\text{ETRST}}$	P63	P77	NC	P70	PE0
7	P36	NC	P35	P34				AVSS	PE1	PE2	PE3
6	P40	P41	P37	P42				PE4	PE7	PE5	PE6
5	P43	P52	P53	P44				P82	P81	NC	P80
4	FWE	VSS	NC	P47	NMI	PC3	PA6	VSS	P85	P84	P83
3	$\overline{\text{RESO}}$	XTAL	NC	VSS	$\overline{\text{STBY}}$	$\overline{\text{MD2}}$	PC0	NC	NC	P87	P86
2	EXTAL	P45	P56	$\overline{\text{RES}}$	NC	PC6	PC1	PA5	PA4	PA1	PA0
1	VCC	P46	P57	MD1	VCL	PC7	PC2	PA7	VCC	PA3	PA2
	A	B	C	D	E	F	G	H	J	K	L

H8S/2153Group
PLBG0112GA-A
BP-112
(Top view)

:NC pin

Figure 1.2 Pin Assignment (BP-112)

C2	P56/IRQ14/PWX0/ϕ/EXCL	NC
C1	P57/$\overline{\text{IRQ15}}$/PWX1	NC
D3	VSS	VSS
D2	$\overline{\text{RES}}$	$\overline{\text{RES}}$
D1	MD1	VSS
E4	NMI	FA9
E3	$\overline{\text{STBY}}$	VCC
E1	VCL	VCL
E2	NC	NC
F3	$\overline{\text{MD2}}$	VCC
F1	PC7/PWX3	$\overline{\text{WE}}$
F2	PC6/PWX2	NC
F4	PC3/SDA3	NC
G1	PC2/SCL3	NC
G2	PC1/SDA2	NC
G3	PC0/SCL2	NC
H1	PA7/EVENT7	VCC
G4	PA6/EVENT6	VCC
H2	PA5/EVENT5	VSS
J1	VCC	VCC
H3	NC	NC
J2	PA4/EVENT4	NC

L3	P86/$\overline{\text{ExIRQ14}}$/RxD3	NC
J4	P85/$\overline{\text{ExIRQ13}}$/SCK1	NC
K4	P84/$\overline{\text{ExIRQ12}}$/SCK3	NC
L4	P83/$\overline{\text{ExIRQ11}}$/SDA1	NC
H5	P82/$\overline{\text{ExIRQ10}}$/SCL1	NC
J5	P81/$\overline{\text{ExIRQ9}}$/SDA0	NC
L5	P80/$\overline{\text{ExIRQ8}}$/SCL0	NC
K5	NC	NC
J6	PE7/SERIRQ	NC
L6	PE6/LCLK	NC
K6	PE5/$\overline{\text{LRESET}}$	NC
H6	PE4/$\overline{\text{LFRAME}}$	NC
L7	PE3/LAD3	NC
K7	PE2/LAD2	NC
J7	PE1/LAD1	NC
L8	PE0/LAD0	NC
H7	AVSS	VSS
K8	P70/$\overline{\text{ExIRQ0}}$/AN0	NC
L9	P71/$\overline{\text{ExIRQ1}}$/AN1	NC
J8	NC	NC
K9	P72/$\overline{\text{ExIRQ2}}$/AN2	NC
L10	P73/$\overline{\text{ExIRQ3}}$/AN3	NC

H9	P60	NC
H10	P61	NC
H11	P62	NC
G8	P63	NC
G9	VCC	VCC
G11	ETMS	NC
G10	NC	NC
F9	ETDO	NC
F11	ETDI	NC
F10	ETCK	NC
F8	$\overline{\text{ETRST}}$	$\overline{\text{RES}}$
E11	VSS	NC
E10	P23	FA11
E9	P22	FA10
D11	P21	OE
E8	P20	FA8
D10	P17	FA7
C11	P16	FA6
D9	NC	NC
C10	P15	FA5
B11	P14	FA4
C9	NC	NC

B8	P32/ExDB2	FO2
A8	P33/ExDB3	FO3
D7	P34/ExDB4	FO4
C7	P35/ExDB5	FO5
A7	P36/ExDB6	FO6
B7	NC	NC
C6	P37/ExDB7	FO7
A6	P40/$\overline{\text{IRQ0}}$/RS0/HC0	NC
B6	P41/$\overline{\text{IRQ1}}$/RS1/HC1	NC
D6	P42/$\overline{\text{IRQ2}}$/RS2/HC2	NC
A5	P43/$\overline{\text{IRQ3}}$/RS3/HC3	NC
B5	P52/$\overline{\text{IRQ10}}$/TxD1	VCC
C5	P53/$\overline{\text{IRQ11}}$/RxD1	VSS
A4	FWE	FWE
D5	P44/$\overline{\text{IRQ4}}$/RS4/DB4/HC4	FA12
B4	VSS	VSS
A3	$\overline{\text{RESO}}$	NC
C4	NC	NC
B3	XTAL	XTAL
A2	EXTAL	EXTAL
C3	NC	NC

				external capacitor (that is located near stabilize internal step-down power.
	VSS	D3, H4, E11, D8, B4	Input	Ground pins. Connect all these pins to power supply (0V).
Clock	XTAL	B3	Input	For connection to a crystal resonator. A clock can be supplied from the EXTAL example of crystal resonator connectio section 21, Clock Pulse Generator.
	EXTAL	A2	Input	
	ϕ	C2	Output	Supplies the system clock to external d
	EXCL	C2	Input	32.768-kHz external clock for sub clock supplied.
Operating mode control	$\overline{MD2}$	F3	Input	These pins set the operating mode. Inp these pins should not be changed durin operation.
	MD1	D1		
System control	$\overline{RES}$	D2	Input	Reset pin. When this pin is low, the chip
	$\overline{RESO}$	A3	Output	Outputs a reset signal to an external de
	$\overline{STBY}$	E3	Input	When this pin is low, a transition is mad hardware standby mode.
	FWE	A4	Input	Pin for use by flash memory.

		J5, L5, H8, K11, K10, L11, L10, K9, L9, K8		
Boundary scan	$\overline{\text{ETRST}}$	F8	Input	Boundary scan interface pins
	ETMS	G11	Input	
	ETDO	F9	Output	
	ETDI	F11	Input	
	ETCK	F10	Input	
14-bit PWM timer (PWMX)	PWX0 PWX1 PWX2 PWX3	C2 C1 F2 F1	Output	PWM D/A pulse output pins
Serial communi-cation interface (SCI_1 and SCI_3)	TxD1, TxD3	B5, K3	Output	Transmit data output pins
	RxD1, RxD3	C5, L3	Input	Receive data input pins
	SCK1, SCK3	J4, K4	Input/ Output	Clock input/output pins.
I^2C bus interface (IIC)	SCL0, SCL1, SCL2, SCL3	L5, H5, G3, G1	Input/ Output	IIC clock input/output pins. These pins ca bus directly with the NMOS open drain o
	SDA0, SDA1, SDA2, SDA3,	J5, L4, G2, F4	Input/ Output	IIC data input/output pins. These pins ca bus directly with the NMOS open drain o

				and D/A converter. When the A/D conv D/A converter are not used, this pin sho connected to the system power supply
	AVSS	H7	Input	Ground pins for the A/D converter and converter. These pins should be conne system power supply (0 V).
	$\overline{\text{ADTRG}}$	K3	Input	External trigger input pin to start A/D co
LPC interface (LPC)	LAD3 to LAD0	L7, K7, J7, L8	Input/ Output	Transfer cycle type/address/data I/O pi
	$\overline{\text{LFRAME}}$	H6	Input	Input pin indicating transfer cycle start a termination
	$\overline{\text{LRESET}}$	K6	Input	LPC reset pin. When this pin is low, a r is entered.
	LCLK	L6	Input	PCI clock input pin
	SERIRQ	J6	Input/ Output	LPC serialized host interrupt request si
Event counter	EVENT7 to EVENT0	H1, G4, H2, J2, K1, L1, K2, L2	Input	Event counter input pins.
Retain state output pins	RS7 to RS0	D4, B1, B2, D5, A5, D6, B6, A6	Output	The outputs on these pins are only initi system reset.
Debounced input pins	DB7 to DB4	D4, B1, B2, D5	Input	Pins with a noise eliminating function.
	ExDB7 to ExDB0	C6, A7, C7, D7, A8, B8, C8, A9		

P37 to P30	C6, A7, C7, D7, A8, B8, C8, A9	Input/ Output	8-bit input/output pins
P47 to P40	D4, B1, B2, D5, A5, D6, B6, A6	Input/ Output	8-bit input/output pins
P57, P56, P53, P52	C1, C2, C5, B5	Input/ Output	4-bit input/output pins
P63 to P60	G8, H11, H10, H9	Input/ Output	4-bit input/output pins
P77 to P70	H8, K11, K10, L11, L10, K9, L9, K8	Input	8-bit input pins
P87 to P80	K3, L3, J4, K4, L4, H5, J5, L5	Input/ Output	8-bit input/output pins
PA7 to PA0	H1, G4, H2, J2, K1, L1, K2, L2	Input/ Output	8-bit input/output pins
PC7, PC6, PC3 to PC0	F1, F2, F4, G1, G2, G3	Input/ Output	6-bit input/output pins
PE7 to PE0	J6, L6, K6, H6, L7, K7, J7, L8	Input/ Output	8-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.

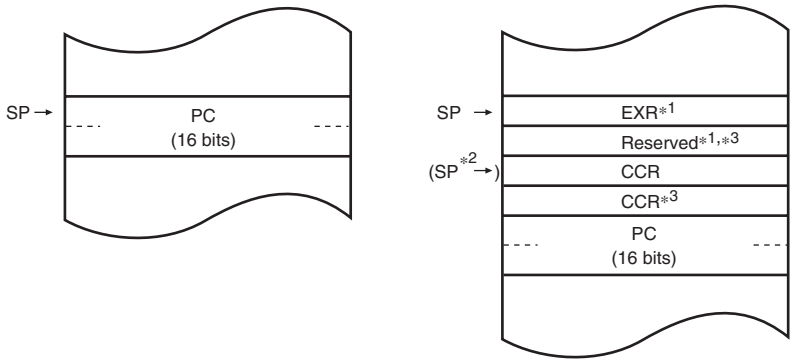
  The memory indirect addressing mode (@@aa:8) employed in the JMP and JSR instr
  uses an 8-bit absolute address included in the instruction code to specify a memory op
  that contains a branch address. In normal mode the operand is a 16-bit word operand,
  providing a 16-bit branch address. Branch addresses 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)



(a) Subroutine Branch (b) Exception Handling

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

Figure 2.2 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.

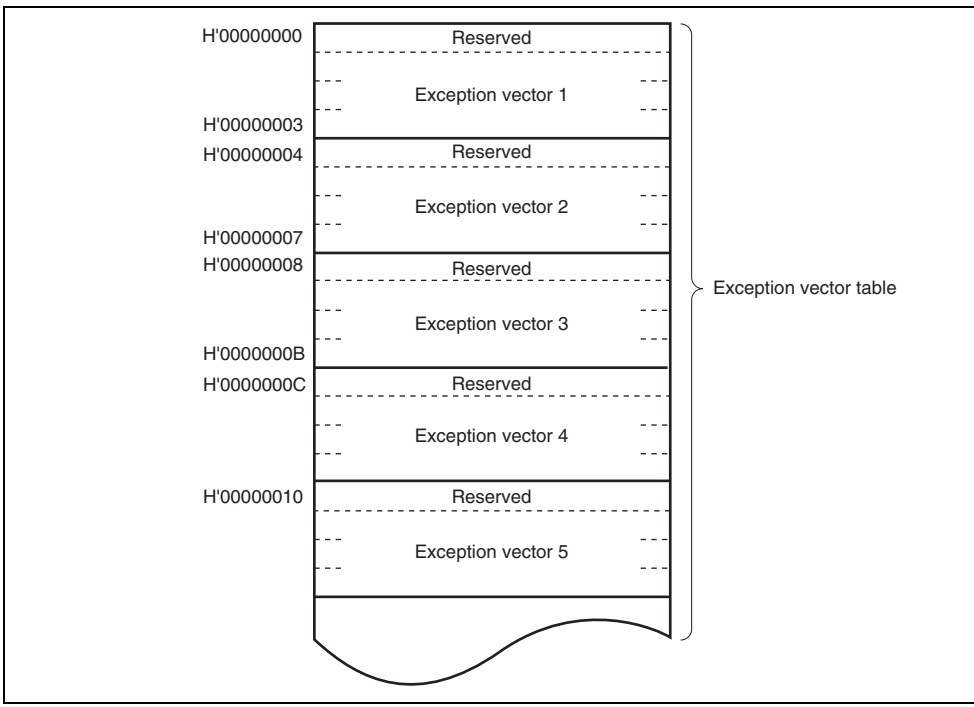
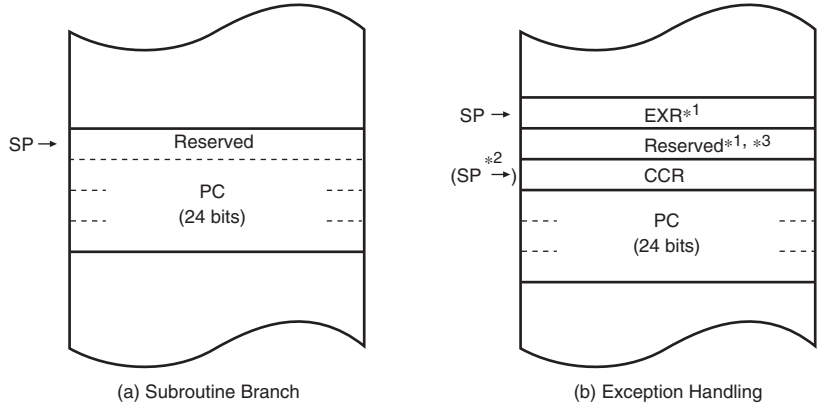


Figure 2.3 Exception Vector Table (Advanced Mode)

EXR is not pushed onto the stack in interrupt control mode 0. For details, see section
Exception Handling.



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

Figure 2.4 Stack Structure in Advanced Mode

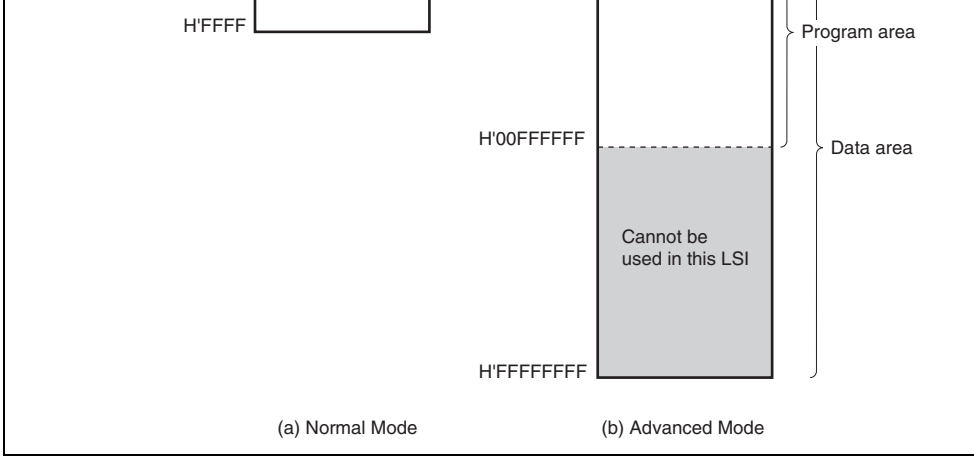
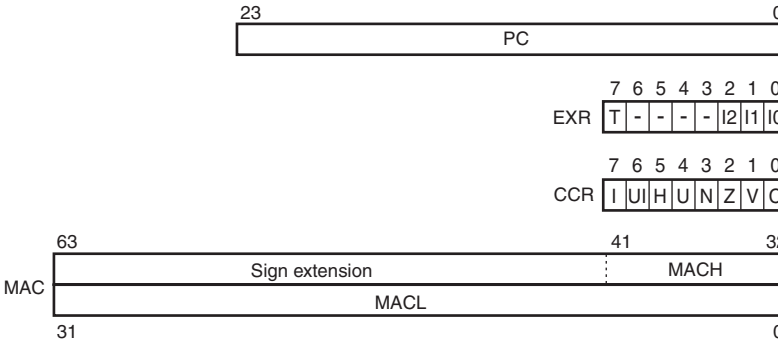


Figure 2.5 Memory Map

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

Control Registers (CR)



[Legend]

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

Figure 2.6 CPU Registers

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

The usage of each register can be selected independently.

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

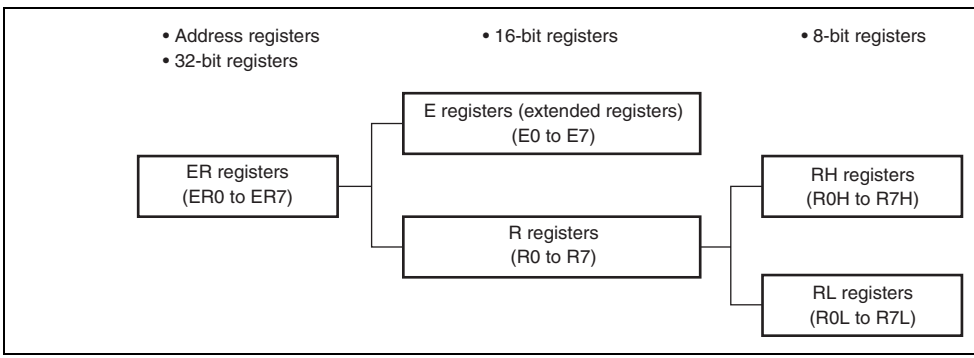


Figure 2.7 Usage of General Registers

Figure 2.8 Stack

2.4.2 Program Counter (PC)

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

2.4.3 Extended Control Register (EXR)

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

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

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

- Shift and rotate instructions, to indicate a c

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.

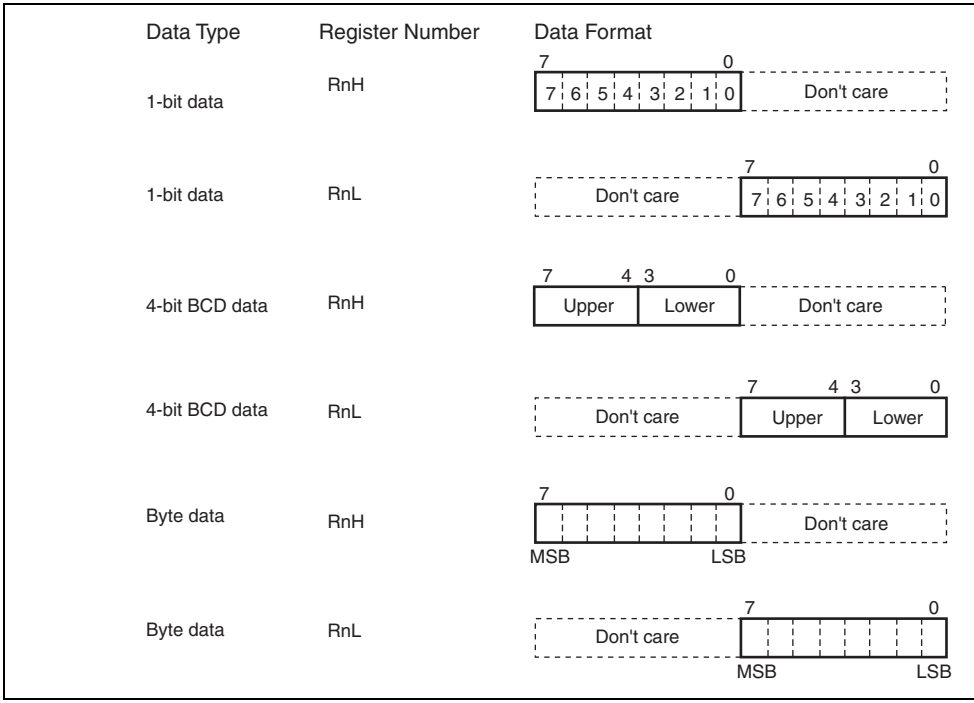
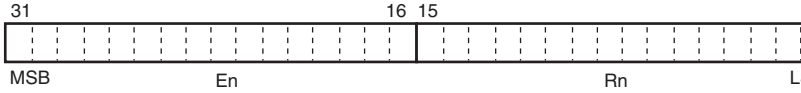


Figure 2.9 General Register Data Formats (1)

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)

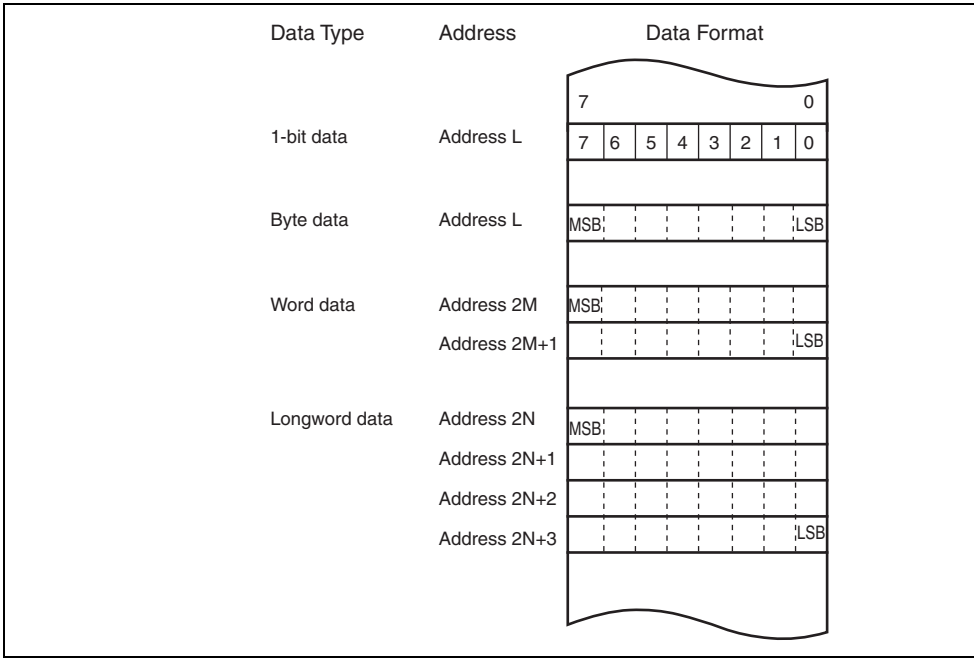


Figure 2.10 Memory Data Formats

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

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

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

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

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

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

Table 2.6 Shift Instructions

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

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

		~(<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 gene 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 r 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	$N \oplus V = 0$
BLT	Less than	$N \oplus V = 1$
BGT	Greater than	$Z \vee (N \oplus V) = 0$
BLE	Less or equal	$Z \vee (N \oplus V) = 1$

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

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

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

```
else next;
```

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

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

Some instructions have two operation fields.

- Register Field

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

- Effective Address Extension

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

- Condition Field

  Specifies the branching condition of Bcc instructions.

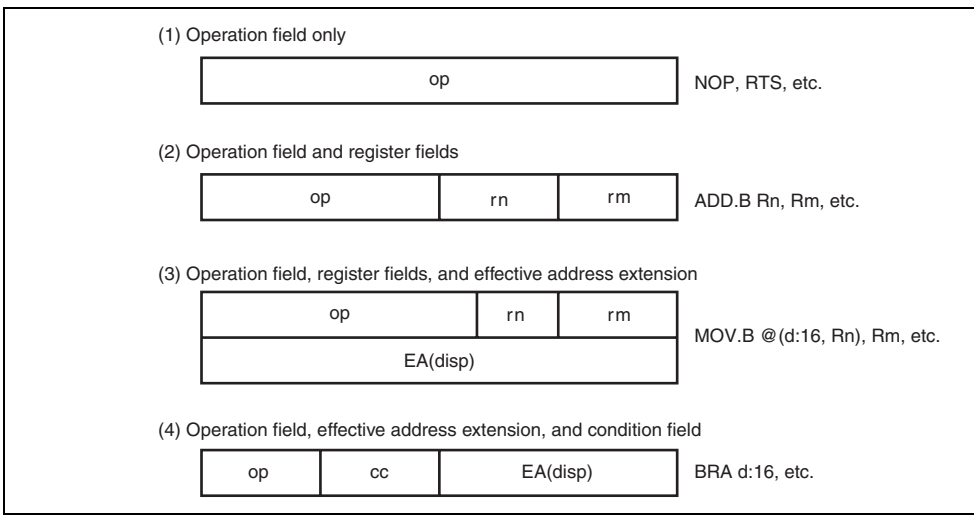


Figure 2.11 Instruction Formats (Examples)

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

2.7.1 Register Direct—Rn

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

2.7.2 Register Indirect—@ERn

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

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

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

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 o
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.

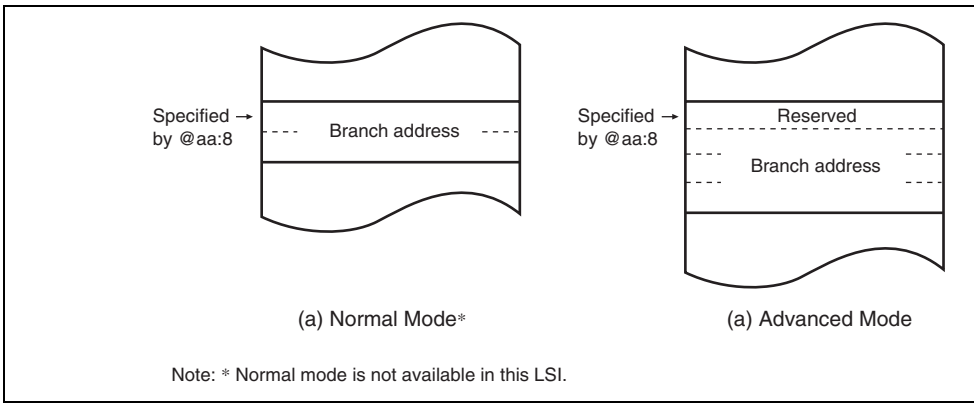
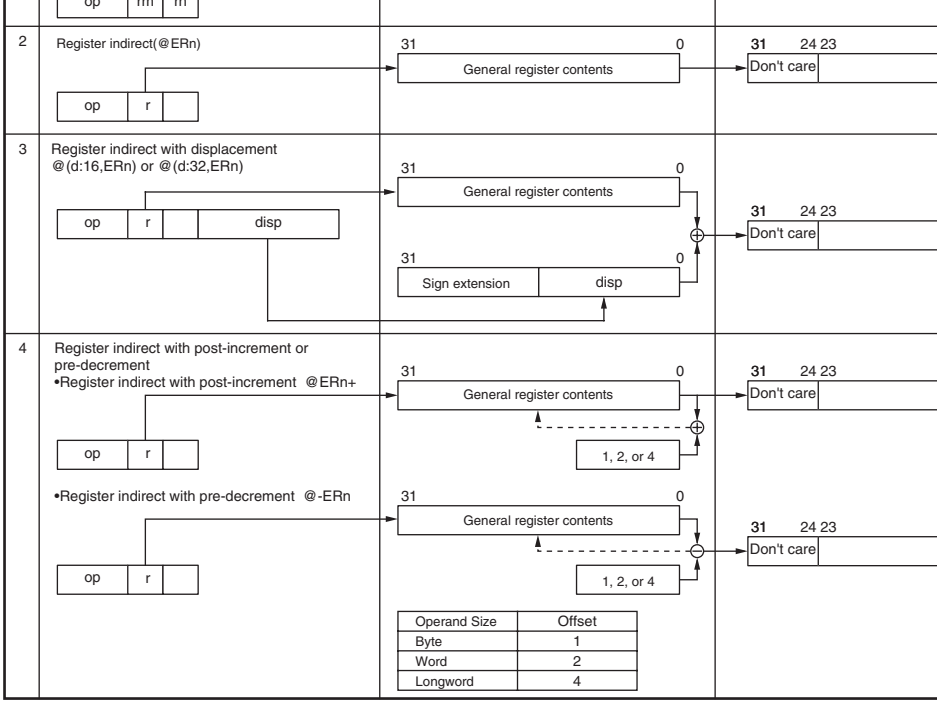
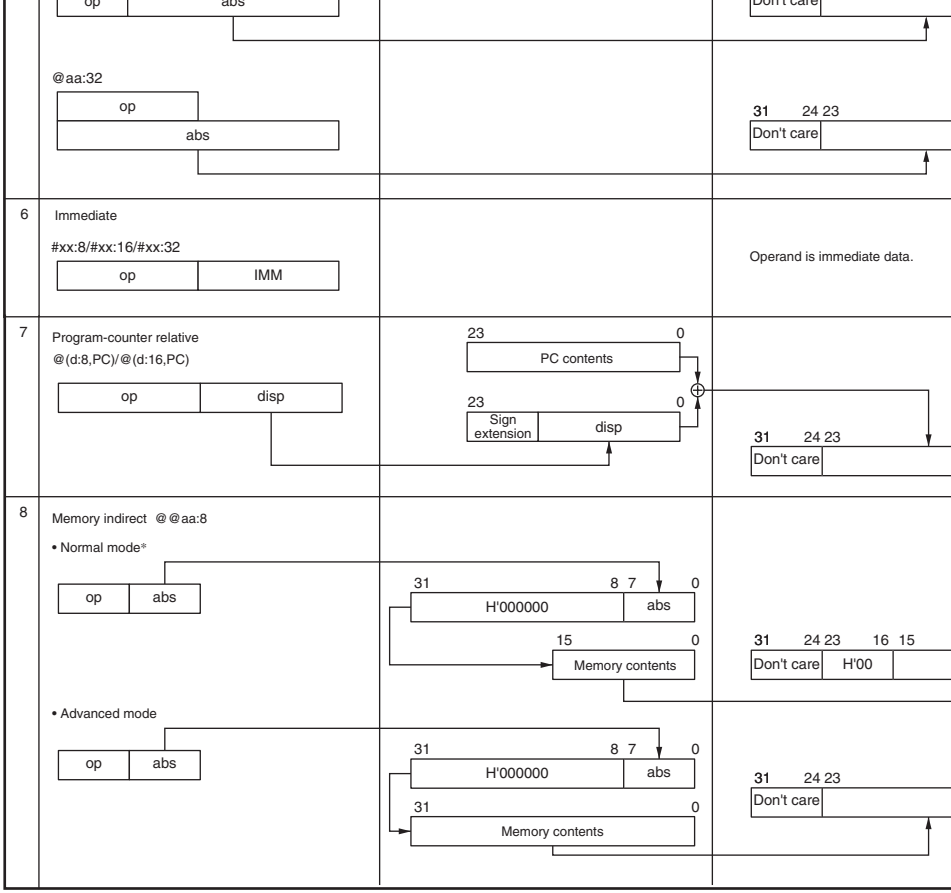


Figure 2.12 Branch Address Specification in Memory Indirect Mode


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



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

- Exception-Handling State

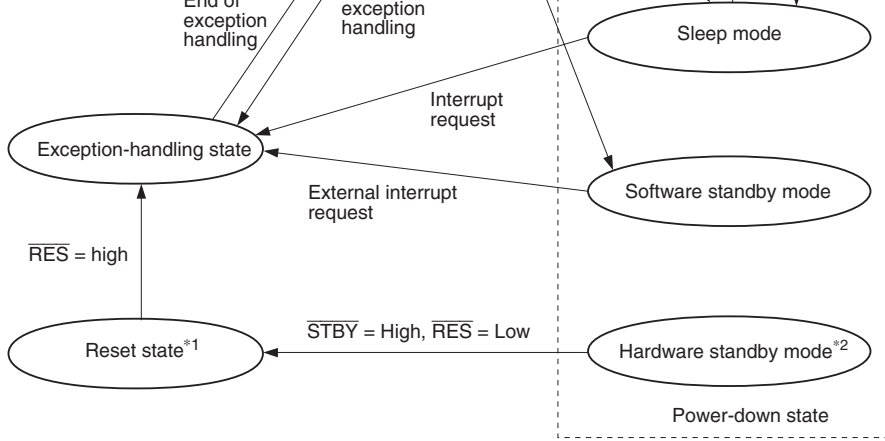
  The exception-handling state is a transient state that occurs when the CPU alters the processing flow due to an exception source, such as a reset, trace, interrupt, or trap in The CPU fetches a start address (vector) from the exception vector table and branche address. For further details, refer to section 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 22, Power-Down Modes.



Notes: 1. From any state except hardware standby mode, a transition to the reset state occurs wheneve
goes low. A transition can also be made to the reset state when the watchdog timer overflows
2. From any state, a transition to hardware standby mode occurs when STBY goes low.

Figure 2.13 State Transitions

beforehand.

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

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

Bit	Bit Name	Initial Value	R/W	Description
7	—	0	R/W	Reserved The initial value should not be changed.
6 to 3	—	All 0	R	Reserved The initial value should not be changed.
2	MDS2	—*	R	Mode Select 2 and 1
1	MDS1	—*	R	These bits indicate the input levels at mode pi and MD1) (the current operating mode). Bits M and MDS1 correspond to $\overline{\text{MD2}}$, MD1, and MD respectively. MDS2 and MDS1 are read-only they cannot be written to. The mode pin ($\overline{\text{MD2}}$ MD1) input levels are latched into these bits w MDCR is read. These latches are canceled by
0	—	0	R	Reserved The initial value should not be changed.

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

				see section 5.6, Interrupt Control Modes and Operation. 00: Interrupt control mode 0 01: Interrupt control mode 1 10: Setting prohibited 11: Setting prohibited
3	XRST	1	R	External Reset This bit indicates the reset source. A reset is by an external reset input, or when the watch overflows. 0: A reset is caused when the watchdog time overflows. 1: A reset is caused by an external reset.
2	NMIEG	0	R/W	NMI Edge Select Selects the valid edge of the NMI interrupt in 0: An interrupt is requested at the falling edg input 1: An interrupt is requested at the rising edge input
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 RAM initialized when the reset state is released. 0: On-chip RAM is disabled 1: On-chip RAM is enabled

				15.3. The IICXn bit controls IIC_n. (n = 0 to
4	—	0	R/W	Reserved The initial value should not be changed.
3	FLSHE	0	R/W	Flash Memory Control Register Enable Enables or disables CPU access for flash m registers (FCCS, FPCS, FECS, FKEY, FMA FTDAR), control registers of power-down sta (SBYCR, LPWRCR, MSTPCRH, MSTPCRL control register of on-chip peripheral module (PCSR). 0: Area from H'FFFE88 to H'FFFE8F is res Control registers of power-down states a chip peripheral modules are accessed in from H'FFFF80 to H'FFFF87. 1: Control registers of flash memory are ac an area from H'FFFE88 to H'FFFE8F. Area from H'FFFF80 to H'FFFF87 is res
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 ti counter (TCNT) and a count condition togetl bits CKS2 to CKS0 in the timer control regis (TCR). For details, see section 11.2.4, Time Register (TCR).

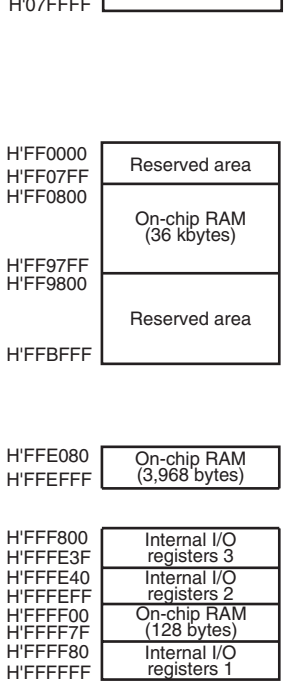


Figure 3.1 Address Map

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.
Low	Trap instruction	Started by execution of a trap (TRAPA) instruction instruction exception handling requests are accep times in program execution state.

Reserved for system use		1	H'000004 to H'000007
		|	|
		3	H'00000C to H'00000F
Illegal instruction exception		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	H'000030 to H'000033
		|	|
		15	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

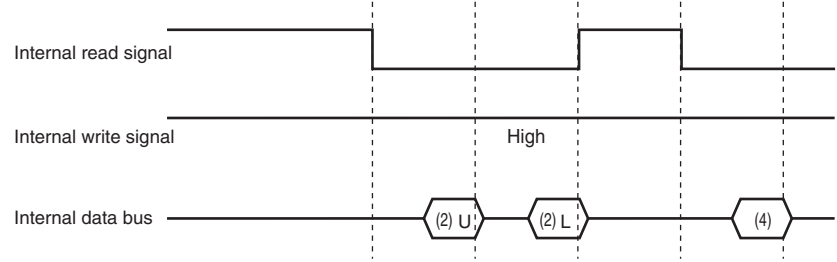
External interrupt	IRQ8	56	H'0000E0 to H'0000E3
	IRQ9	57	H'0000E4 to H'0000E7
	IRQ10	58	H'0000E8 to H'0000EB
	IRQ11	59	H'0000EC to H'0000EF
	IRQ12	60	H'0000F0 to H'0000F3
	IRQ13	61	H'0000F4 to H'0000F7
	IRQ14	62	H'0000F8 to H'0000FB
	IRQ15	63	H'0000FC to H'0000FF
Internal interrupt*		64	H'000100 to H'000103
		|	|
		107	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, and
SUBMSTPB) are initialized, and all modules except the DTC operate in module stop m
Therefore, the registers of on-chip peripheral modules cannot be read from or written to
from and write to these registers, clear module stop mode.

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

4.5 Trap Instruction Exception Handling

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

Trap instruction exception handling is conducted as follows:

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

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

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

Table 4.3 Status of CCR after Trap Instruction Exception Handling

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

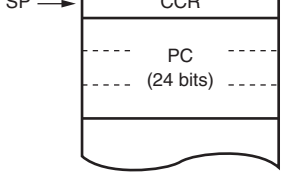


Figure 4.2 Stack Status after Exception Handling

Use the following instructions to restore registers:

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

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

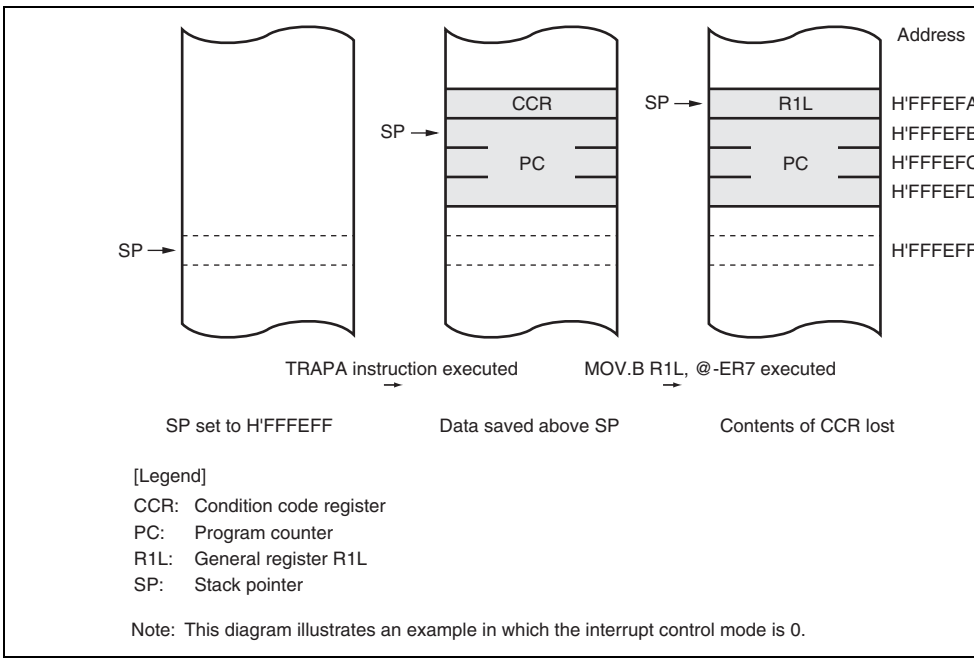


Figure 4.3 Operation When SP Value is Odd

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

- Three-level interrupt mask control

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

- Independent vector addresses

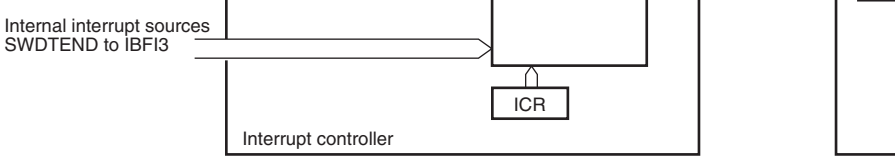
  All interrupt sources are assigned independent vector addresses, making it unnecessa
  source to be identified in the interrupt handling routine.

- Twenty-nine 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 $\overline{\text{IRQ}}$n (n = 15, 14, 11, 10, and 7 to 0) and $\overline{\text{ExIRQ}}$m (m =

- DTC control

  The DTC can be activated by an interrupt request.



[Legend]
ICR: Interrupt control register
ISCR: IRQ sense control register
IER: IRQ enable register
ISR: IRQ status register
SYSCR: System control register

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
$\overline{\text{IRQ15}}$, $\overline{\text{IRQ14}}$, $\overline{\text{IRQ11}}$ $\overline{\text{IRQ10}}$, $\overline{\text{IRQ7}}$ 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, 14, 11, 10, and 7 to 0) interrupt can be selec

- 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	—
0	ICRn0	WDT_1	—	—	—

[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	
3	IRQ13SCB	0	R/W	01: Interrupt request generated at falling edge of ExIRQn input
2	IRQ13SCA	0	R/W	10: Interrupt request generated at rising edge of I ExIRQn input
1	IRQ12SCB	0	R/W	11: Interrupt request generated at both falling and edges of IRQn* or ExIRQn input
0	IRQ12SCA	0	R/W	(n = 15 to 12) Note: * IRQn here only stands for IRQ15 and IR

Note: For n = 13 or 12, only ExIRQ can be selected.

- 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	
5	IRQ10SCB	0	R/W	00: Interrupt request generated at low level of IRQ ExIRQn input
4	IRQ10SCA	0	R/W	
3	IRQ9SCB	0	R/W	01: Interrupt request generated at falling edge of ExIRQn input
2	IRQ9SCA	0	R/W	10: Interrupt request generated at rising edge of I ExIRQn input
1	IRQ8SCB	0	R/W	11: Interrupt request generated at both falling and edges of IRQn* or ExIRQn input
0	IRQ8SCA	0	R/W	(n = 11 to 8) Note: * IRQn here only stands for IRQ11 and IR

Note: For n = 9 or 8, only ExIRQ can be selected.

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)

- ISCRL

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

- 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 $\overline{\text{IRQ15}}$, $\overline{\text{IRQ14}}$, $\overline{\text{IRQ11}}$ a

- 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 $\overline{\text{ExIRQ}}$ high • When IRQn interrupt exception handling is e when falling-edge, rising-edge, or both-edge is set (n = 7 to 0)

edge on the NMI pin.

IRQ15 to IRQ0 Interrupts: Interrupts IRQ15 to IRQ0 are requested by an input signal a
$\overline{\text{IRQ15}}$, $\overline{\text{IRQ14}}$, $\overline{\text{IRQ11}}$, $\overline{\text{IRQ10}}$, $\overline{\text{IRQ7}}$ to $\overline{\text{IRQ0}}$ or pins $\overline{\text{ExIRQ15}}$ to $\overline{\text{ExIRQ0}}$. Interrupts IR
IRQ0 have the following 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 $\overline{\text{IRQ15}}$, $\overline{\text{IRQ14}}$, $\overline{\text{IRQ11}}$, $\overline{\text{IRQ10}}$, $\overline{\text{IRQ7}}$ to $\overline{\text{IRQ}}$
  $\overline{\text{ExIRQ15}}$ 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 modul
- 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 IRQ3	18 19	H'000048 H'00004C	ICRA5	
	IRQ4 IRQ5	20 21	H'000050 H'000054	ICRA4	
	IRQ6 IRQ7	22 23	H'000058 H'00005C	ICRA3	
DTC	SWDTEND (Software activation data transfer end)	24	H'000060	ICRA2	
WDT_0	WOVI0 (Interval timer)	25	H'000064	ICRA1	
WDT_1	WOVI1 (Interval timer)	26	H'000068	ICRA0	
—	Address break	27	H'00006C	—	
A/D converter	ADI (A/D conversion end)	28	H'000070	ICRB7	
EVC	EVENTI	29	H'000074	—	
TMR_X	CMIAX (Compare match A) CMIBX (Compare match B) OVIX (Overflow)	44 45 46	H'0000B0 H'0000B4 H'0000B8	ICRB4	
FRT	OCIA (Output compare A) OCIB (Output compare B) FOVI (Overflow)	52 53 54	H'0000D0 H'0000D4 H'0000D8	ICRB6	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	
IIC_0	IICI0	94	H'000178	ICRC4	
IIC_1	IICI1	98	H'000188	ICRC3	
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	L

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

Mode	INTM1	INTM0	Registers	Mask Bits	Description
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.

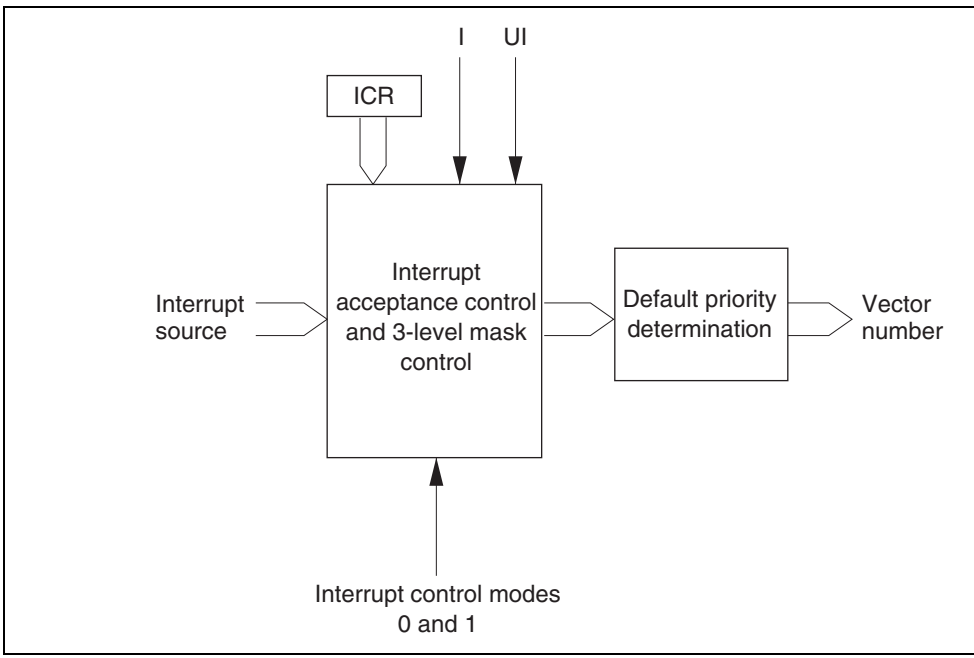


Figure 5.3 Block Diagram of Interrupt Control Operation

			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. The EVENTI interrupt is enabled or disabled by the
4. When the CPU accepts an interrupt request, it starts interrupt exception handling after execution of the current instruction has been completed.
5. The PC and CCR are saved to the stack area by interrupt exception handling. The PC the stack shows the address of the first instruction to be executed after returning from interrupt handling routine.
6. Next, the I bit in CCR is set to 1. This masks all interrupts except for NMI and addres interrupts.
7. The CPU generates a vector address for the accepted interrupt and starts execution of interrupt handling routine at the address indicated by the contents of the vector addres vector table.

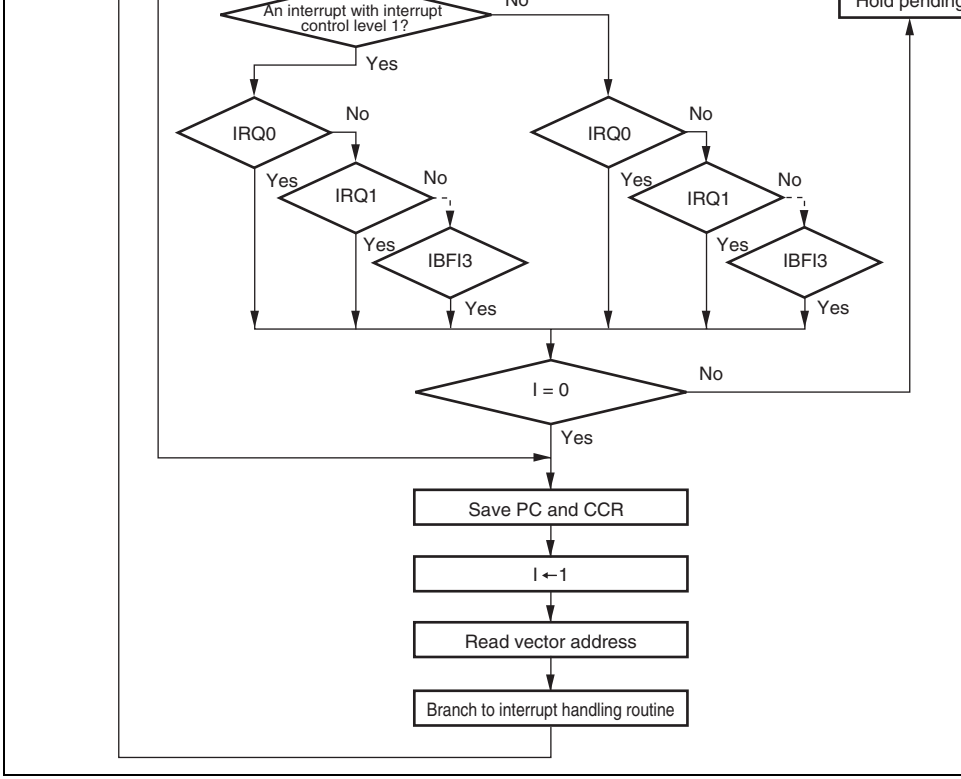


Figure 5.4 Flowchart of Procedure up to Interrupt Acceptance in Interrupt Contr

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.

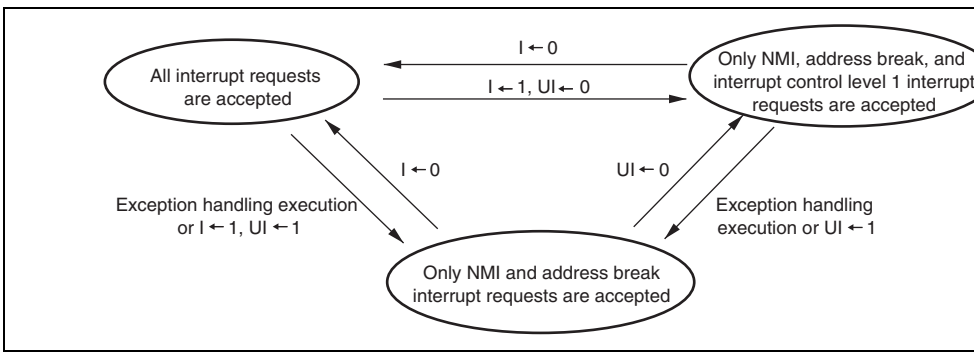


Figure 5.5 State Transition in Interrupt Control Mode 1

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 fron 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.

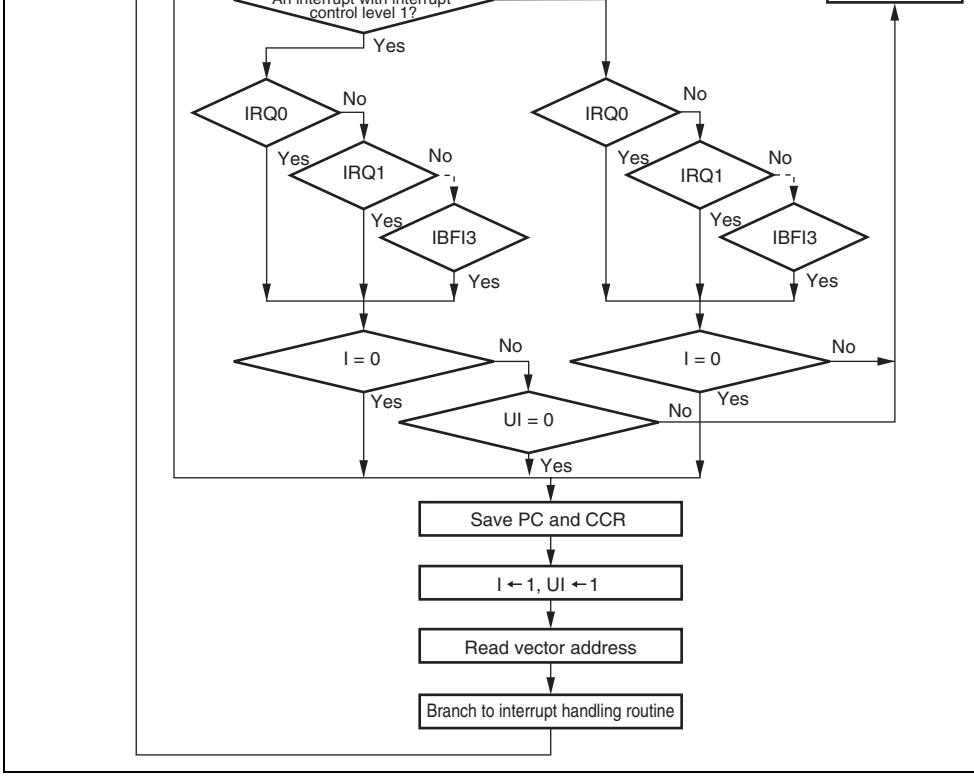


Figure 5.6 Flowchart of Procedure Up to Interrupt Acceptance in Interrup
Control Mode 1

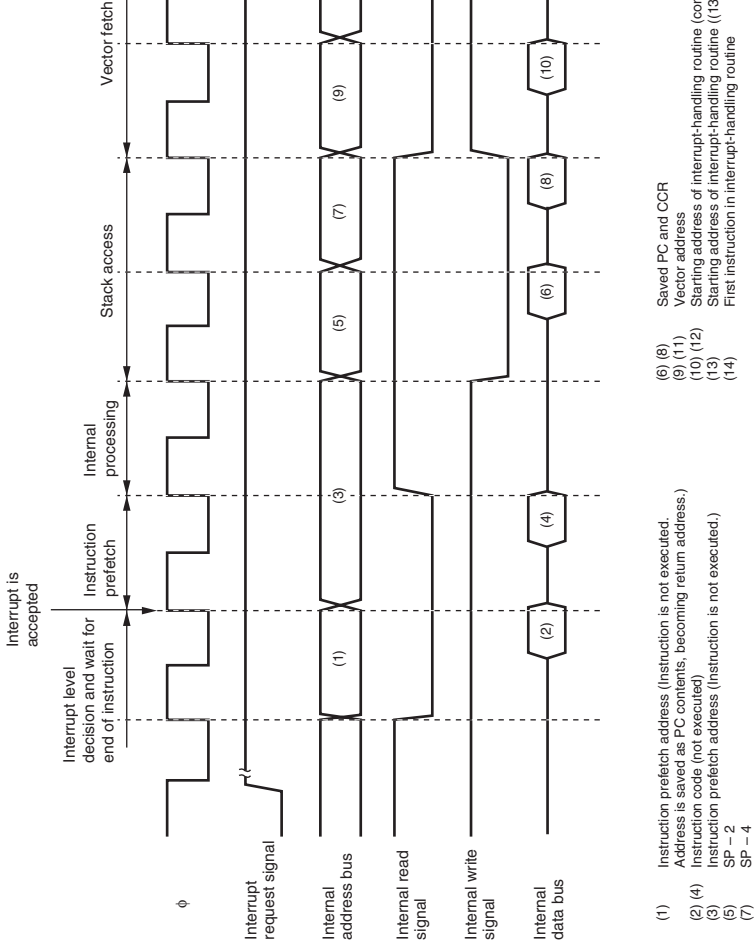


Figure 5.7 Interrupt Exception Handling

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

Symbol	Internal Memory
Instruction fetch S_I	1
Branch address read S_J	
Stack manipulation S_K	

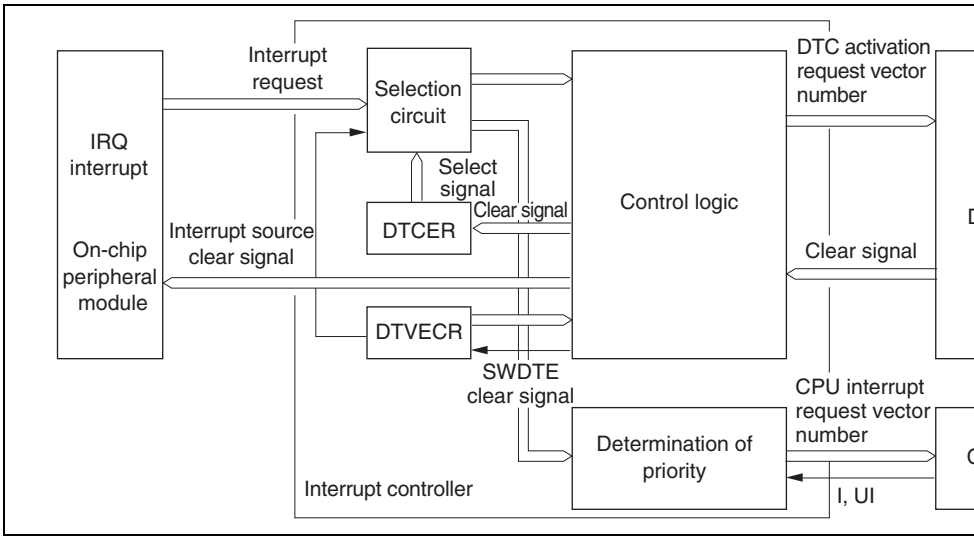


Figure 5.8 Interrupt Control for DTC

The interrupt controller has three main functions in DTC control.

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

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

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

[Legend]

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

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

×: The relevant interrupt cannot be used.

*: Don't care

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.

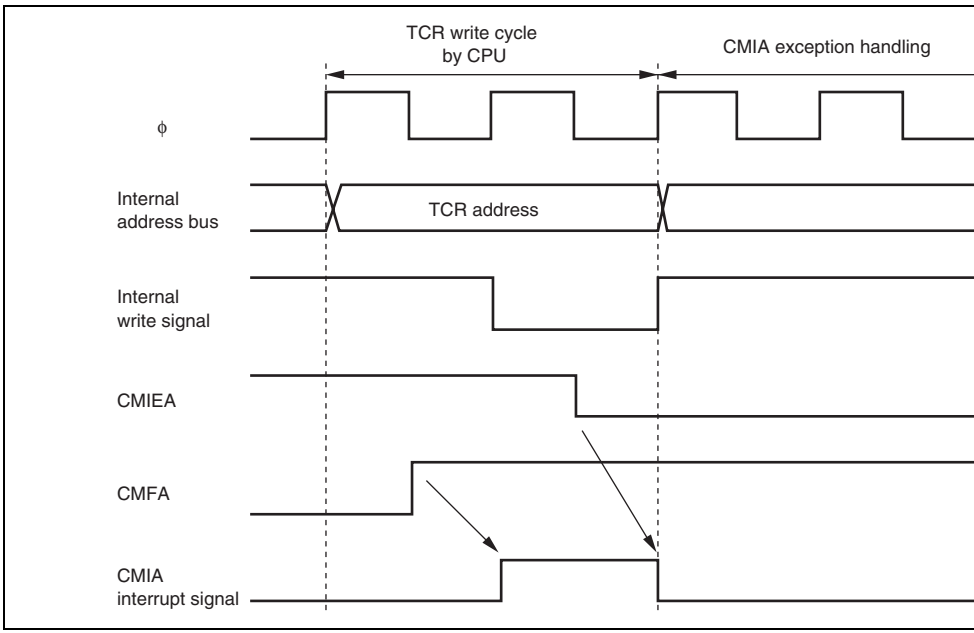


Figure 5.9 Conflict between Interrupt Generation and Disabling

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).

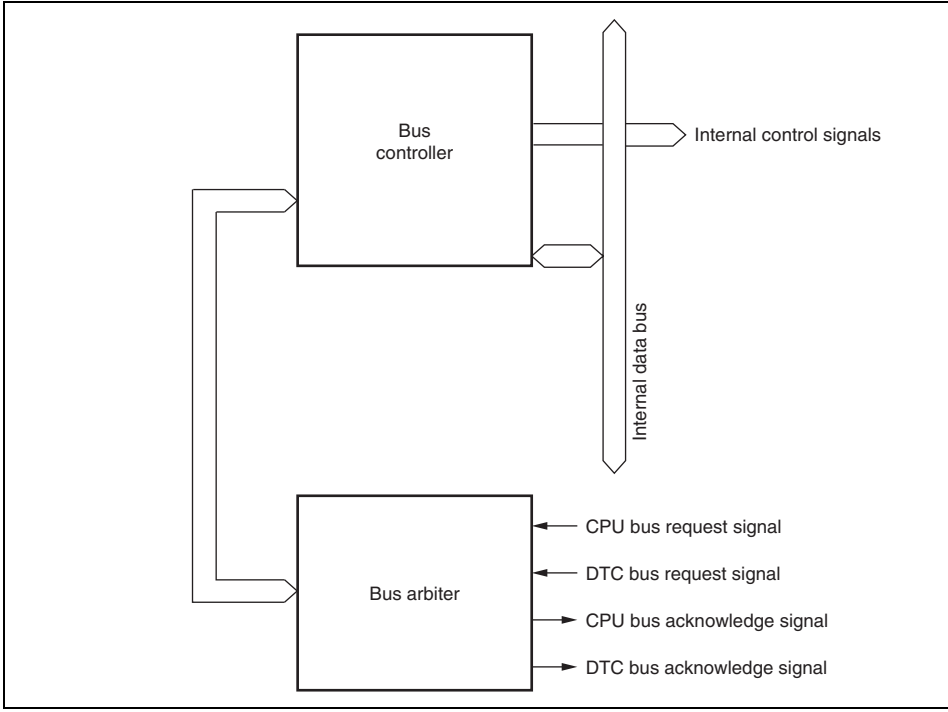


Figure 6.1 Block Diagram of Bus Controller

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

(High) DTC > CPU (Low)

6.2.3 Bus Mastership Transfer Timing

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

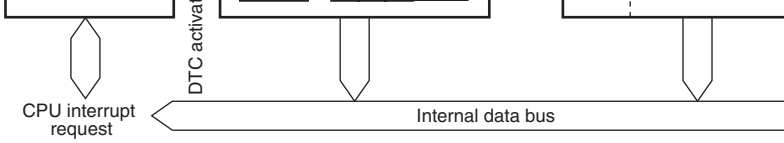
(1) CPU

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

1. Bus mastership is transferred at a break between bus cycles. However, if bus cycle is
   in discrete operations, as in the case of a 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.
2. If the CPU is in sleep mode, it transfers the bus mastership immediately.

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 in
source occurs, the DTC reads a set of register information that is stored in on-chip RAM
corresponding DTC registers and transfers data. After the data transfer, it writes a set of
register information back to on-chip RAM.

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

				(by +1 when Sz = 0, by +2 when Sz = 1) 11: SAR is decremented after a transfer (by –1 when Sz = 0, by –2 when Sz = 1)
5 4	DM1 DM0	Undefined	—	Destination Address Mode 1 and 0 These bits specify a DAR operation after a dat transfer. 0x: DAR is fixed 10: DAR is incremented after a transfer (by +1 when Sz = 0, by +2 when Sz = 1) 11: DAR is decremented after a transfer (by –1 when Sz = 0, by –2 when Sz = 1)
3 2	MD1 MD0	Undefined	—	DTC Mode 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: x 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. 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 func
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 decreme
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 r of transfers have not ended • When 0 is written to the DISEL bit after a s activated data transfer end interrupt (SWDT request has been sent to the CPU. This bit will not be cleared when the DISEL bit data transfer has ended or when the specified of 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 so activation. The vector address is expressed as H'0400 + (number × 2). For example, when DTVEC6 to D = H'10, the vector address is H'0420. When the SWDTE bit is 0, these bits can be written to.

				These bits are always read as 0 and cannot b modified.
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 coun 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 b modified.
3	—	0	R/W	Reserved The initial value should not be changed.

				100: EVENT0 to EVENT4 are used
				101: EVENT0 to EVENT5 are used
				110: EVENT0 to EVENT6 are used
				111: EVENT0 to EVENT7 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 monito 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 8	—	All 0	R	Reserved
7 to 0	E7 to E0	0	R	Event Monitor 7 to 0
				These bits indicate processed/unprocessed sta the events that are input to EVENT7 to EVENT
				0: The corresponding event is already process
				1: The corresponding event is not yet processe

	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) + 14: Result of EVENT 7 count

The corresponding flag to ECS input pin is set to 1 when the event pins that are specifie ECSB2 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	B'00010
					1	0	0	B'00100
				1	0	0	0	B'00110
			1	0	0	0	0	B'01000
		1	0	0	0	0	0	B'01010
	1	0	0	0	0	0	0	B'01100
1	0	0	0	0	0	0	0	B'01110

7.3.1 Event Counter Handling Priority

Counting of EVENT0 to EVENT7 is processed in the priority shown as below.

High Low

EVENT0 > EVENT1 · · · · · · · · · · EVENT6 > EVENT7

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 acti
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.

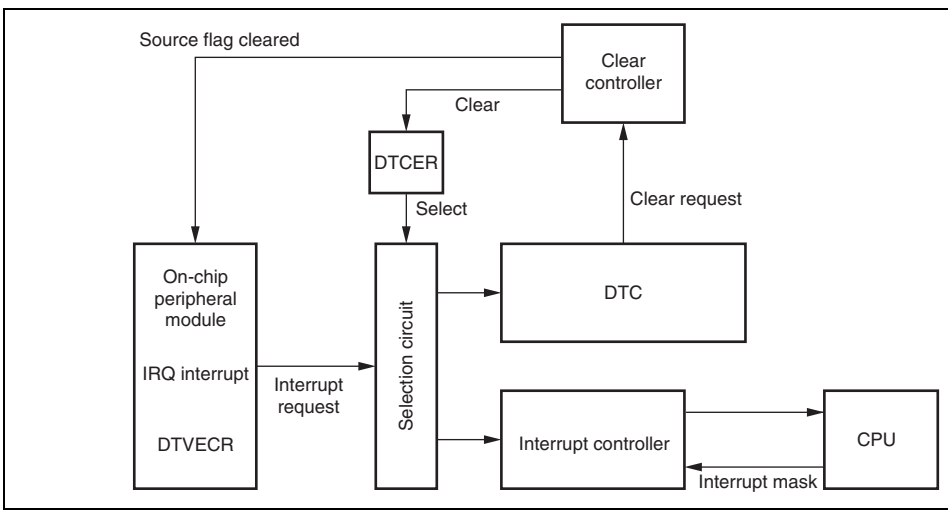


Figure 7.2 Block Diagram of DTC Activation Source Control

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 th information start address.

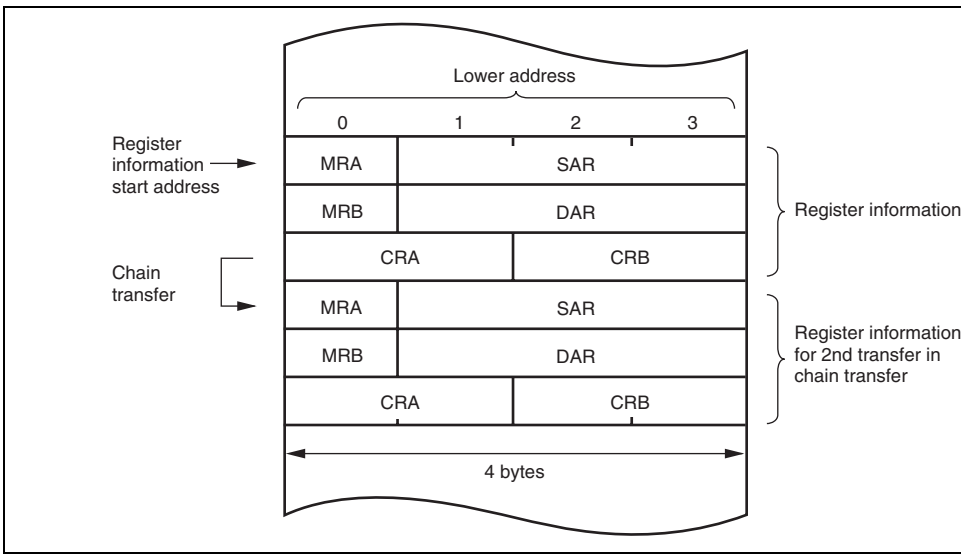


Figure 7.3 DTC Register Information Location in Address Space

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

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 s
always be 0.

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

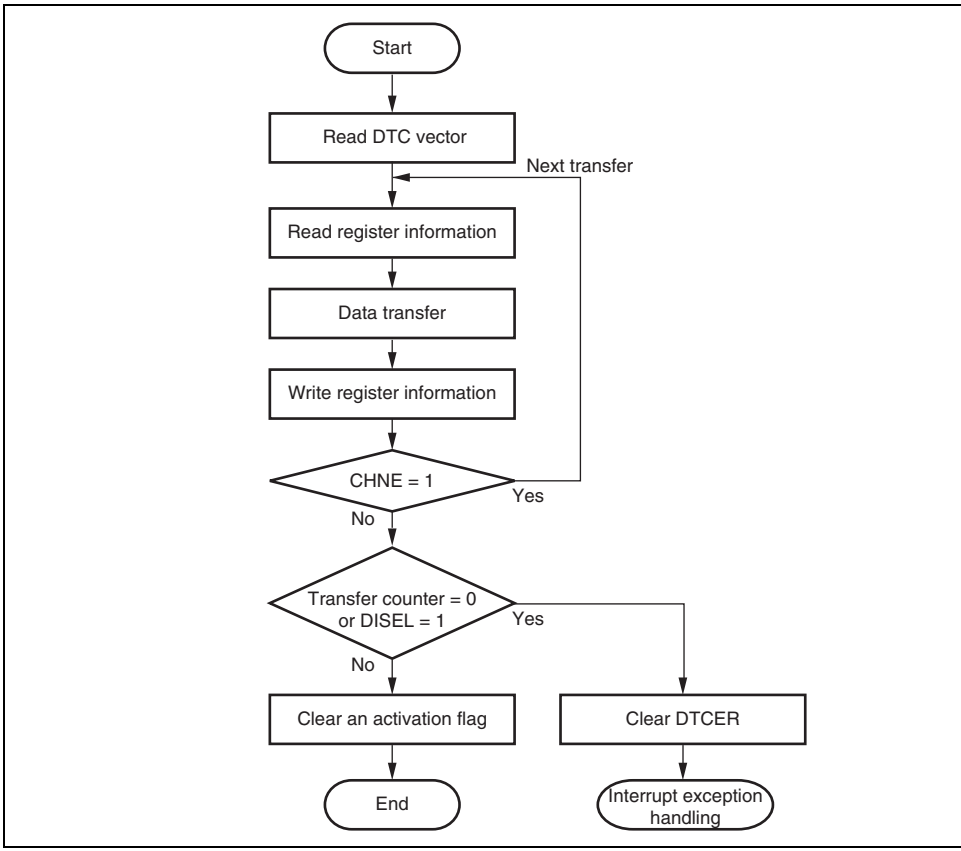


Figure 7.5 DTC Operation Flowchart

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

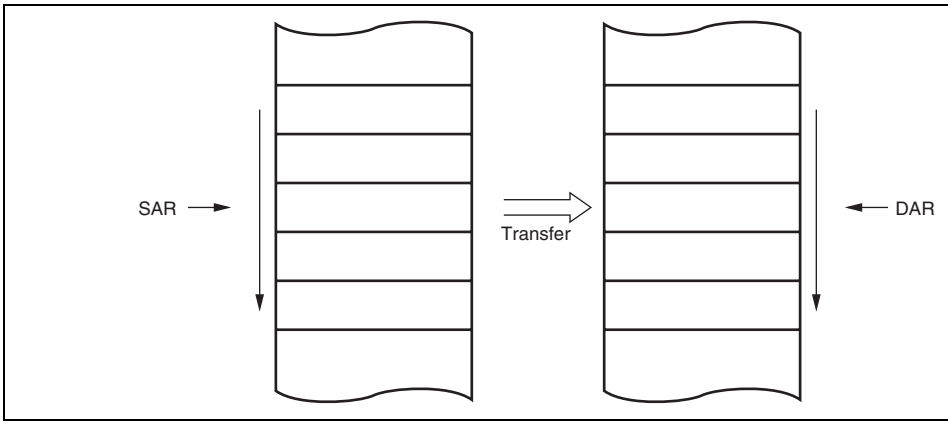


Figure 7.6 Memory Mapping in Normal Transfer Mode

Name	Abbreviation	Function
DTC source address register	SAR	Transfer source address
DTC destination address register	DAR	Transfer destination address
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

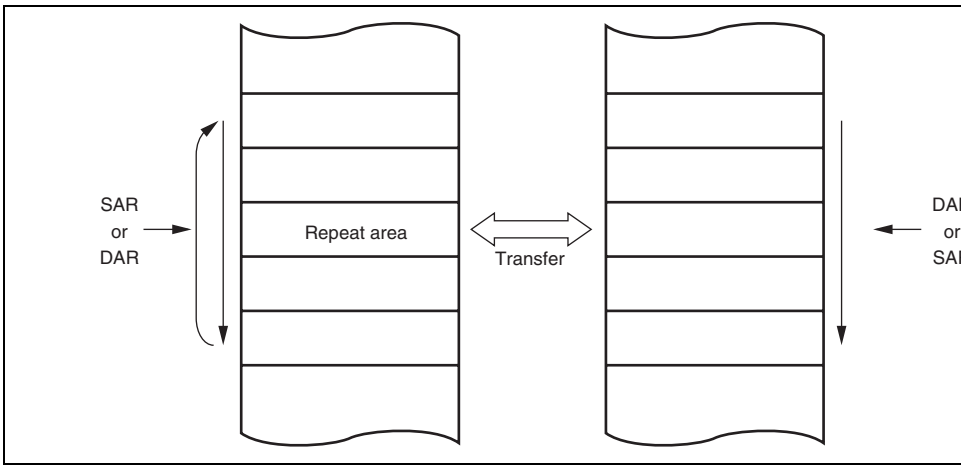


Figure 7.7 Memory Mapping in Repeat Transfer Mode

Name	Abbreviation	Function
DTC source address register	SAR	Transfer source address
DTC destination address register	DAR	Transfer destination addres
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

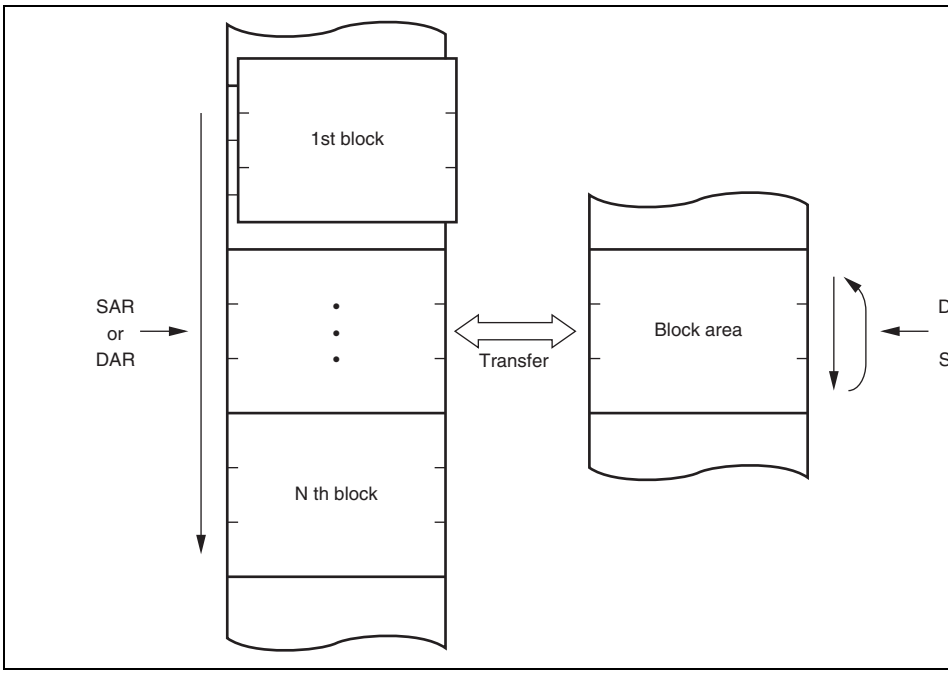


Figure 7.8 Memory Mapping in Block Transfer Mode

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

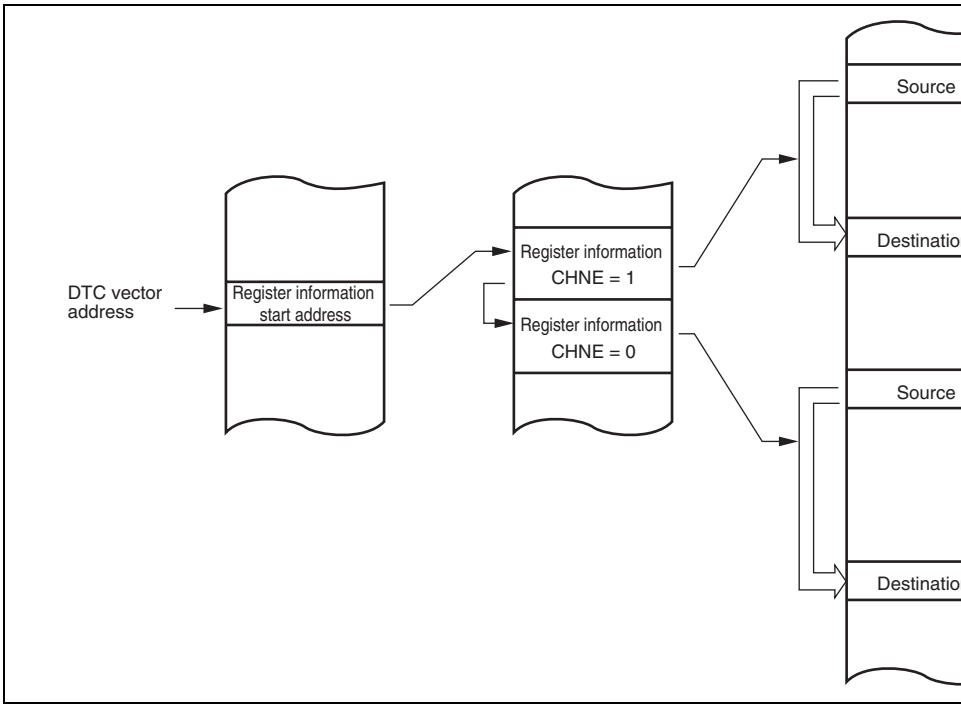


Figure 7.9 Chain Transfer Operation

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

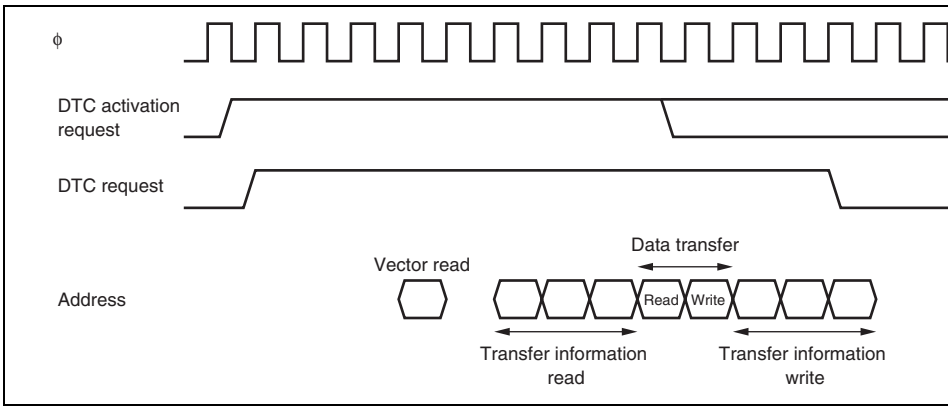


**Figure 7.10 DTC Operation Timing (Example in Normal Transfer Mode or R
Transfer Mode)**

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

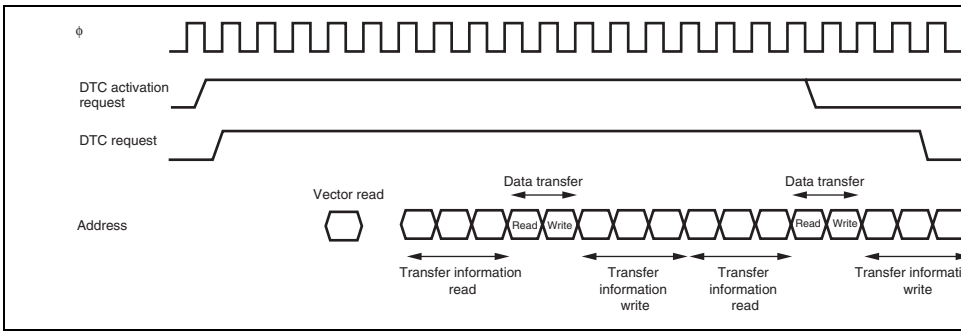


Figure 7.12 DTC Operation Timing (Example of Chain Transfer)

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 the
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-Ch Regis
Bus width		32	16	16	8	
Access states		1	1	1	2	
Execution status	Vector read S_I	—	—	1	—	
	Register information read/write S_J	1	—	—	—	
	Byte data read S_K	1	1	1	2	
	Word data read S_K	1	1	1	4	
	Byte data write	1	1	1	2	
	Word data write S_L	1	1	1	4	
	Internal operation S_M	1	1	1	1	

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

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

For example, when the DTC vector address table is located in on-chip ROM, normal tra mode is set, and data is transferred from on-chip ROM to an internal I/O register, then t

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 trans
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 RAM
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 interrupt
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 transf
been completed, the SWDTE bit is held at 1 and a CPU interrupt is requested.

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

2. Set the start address of the register information at the DTC vector address.
3. Set the corresponding bit in DTCER to 1.
4. Set the SCI to the appropriate receive mode. Set the RIE bit in SCR to 1 to enable th reception complete (RXI) interrupt. Since the generation of a receive error during the reception operation will disable subsequent reception, the CPU should be enabled to receive error interrupts.
5. Each time the reception of one byte of data has been completed on the SCI, the RDR SSR is set to 1, an RXI interrupt is generated, and the DTC is activated. The receive transferred from RDR to RAM by the DTC. DAR is incremented and CRA is decren The RDRF flag is automatically cleared to 0.
6. When CRA becomes 0 after 128 data transfers have been completed, the RDRF flag 1, the DTCE bit is cleared to 0, and an RXI interrupt request is sent to the CPU. The 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 transfer by software.
4. Write 1 to the SWDTE bit and the vector number (H'60) to DTVECR. The write data
5. Read DTVECR again and check that it is set to the vector number (H'60). If it is not, t indicates that the write failed. This is presumably because an interrupt occurred betwe 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 sho 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 DT
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 re
writing. Multiple DTC activation sources can be set at one time (only at the initial settin
masking all interrupts and writing data after executing a dummy read on the relevant reg

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 v
reads from or writes to the respective registers, and they cannot be cleared by the DISEI

MOSs.

Port 3 pins and pins 47 to 44 have built-in de-bouncers (DBn) that eliminate noises in th
signals.

Port 4 are designed for retain state outputs (RSn), which retain the output values on the p
if a reset is generated when 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. Ports 8, and C0 to C3 are NMOS push-
output.

Table 8.1 Port Functions

Port	Description	Single-Chip Mode	I/O S
Port 1	General I/O port	P17	Built
		P16	pull-
		P15	
		P14	
		P13	
		P12	
		P11	
		P10	
Port 2	General I/O port	P23	Built
		P22	pull-
		P21	
		P20	

Port 4	General I/O port also functioning as interrupt input, retain state output, and de-bounce input	P47/IRQ7/RS7/DB7/HC7 P46/$\overline{\text{IRQ6}}$/RS6/DB6/HC6 P45/$\overline{\text{IRQ5}}$/RS5/DB5/HC5 P44/$\overline{\text{IRQ4}}$/RS4/DB4/HC4 P43/$\overline{\text{IRQ3}}$/RS3/HC3 P42/$\overline{\text{IRQ2}}$/RS2/HC2 P41/$\overline{\text{IRQ1}}$/RS1/HC1 P40/$\overline{\text{IRQ0}}$/RS0/HC0	Built-i pull-u (sink 12 m/
Port 5	General I/O port also functioning as interrupt input, system clock output, PWMX output, and SCI_1 input/output	P57/$\overline{\text{IRQ15}}$/PWX1 P56/$\overline{\text{IRQ14}}$/PWX0/ϕ/EXCL P53/$\overline{\text{IRQ11}}$/RxD1 P52/$\overline{\text{IRQ10}}$/TxD1	
Port 6	General I/O port	P63 P62 P61 P60	Built-i pull-u
Port 7	General I/O port also functioning as A/D converter analog input and interrupt input	P77/$\overline{\text{ExIRQ7}}$/AN7 P76/$\overline{\text{ExIRQ6}}$/AN6 P75/$\overline{\text{ExIRQ5}}$/AN5 P74/$\overline{\text{ExIRQ4}}$/AN4 P73/$\overline{\text{ExIRQ3}}$/AN3 P72/$\overline{\text{ExIRQ2}}$/AN2 P71/$\overline{\text{EXIRQ1}}$/AN1 P70/$\overline{\text{EXIRQ0}}$/AN0	

Port A	General I/O port also functioning as DTC event counter input	PA7/EVENT7		Built-in pull-
		PA6/EVENT6		
		PA5/EVENT5		
		PA4/EVENT4		
		PA3/EVENT3		
		PA2/EVENT2		
		PA1/EVENT1		
		PA0/EVENT0		
Port C	General I/O port also functioning as PWMX output, and IIC_2 and IIC_3 input/output	PC7/PWX3		NMO pull (PC
		PC6/PWX2		
		PC3/SDA3		
		PC2/SCL3		
		PC1/SDA2		
		PC0/SCL2		
Port E	General I/O port also functioning as LPC input/output	PE7/SERIRQ		
		PE6/LCLK		
		PE5/$\overline{\text{LRESET}}$		
		PE4/$\overline{\text{LFRAME}}$		
		PE3/LAD3		
		PE2/LAD2		
		PE1/LAD1		
		PE0/LAD0		

Bit	Bit Name	Initial Value	R/W	Description
7	P17DDR	0	W	The corresponding port 1 pins function as ad output port when the P1DDR bits are set to 1 input port when cleared to 0.
6	P16DDR	0	W	
5	P15DDR	0	W	
4	P14DDR	0	W	
3	P13DDR	0	W	
2	P12DDR	0	W	
1	P11DDR	0	W	
0	P10DDR	0	W	

8.1.2 Port 1 Data Register (P1DR)

P1DR stores output data for the port 1 pins.

Bit	Bit Name	Initial Value	R/W	Description
7	P17DR	0	R/W	P1DR stores output data for the port 1 pins th used as the general output port. If a port 1 read is performed while the P1DDR set to 1, the P1DR values are read. If a port 1 performed while the P1DDR bits are cleared pin states are read.
6	P16DR	0	R/W	
5	P15DR	0	R/W	
4	P14DR	0	R/W	
3	P13DR	0	R/W	
2	P12DR	0	R/W	
1	P11DR	0	R/W	
0	P10DR	0	R/W	

2	P12PCR	0	R/W
1	P11PCR	0	R/W
0	P10PCR	0	R/W

8.1.4 Port 1 Input Pull-Up MOS

Port 1 has a built-in input pull-up MOS that can be controlled by software. This input pu
MOS can be used regardless of the operating mode. Table 8.2 summarizes the input pull
states.

Table 8.2 Port 1 Input Pull-Up MOS States

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

[Legend]
Off: Always off.
On/Off: On when P1DDR = 0 and P1PCR = 1; otherwise off.

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

Bit	Bit Name	Initial Value	R/W	Description
7	—	0	W	Reserved
6	—	0	W	
5	—	0	W	
4	—	0	W	
3	P23DDR	0	W	The corresponding port 2 pins function as output port when the P2DDR bits are set t as input port when cleared to 0.
2	P22DDR	0	W	
1	P21DDR	0	W	
0	P20DDR	0	W	

8.2.2 Port 2 Data Register (P2DR)

P2DR stores output data for the port 2 pins.

Bit	Bit Name	Initial Value	R/W	Description
7	—	0	R/W	Reserved
6	—	0	R/W	
5	—	0	R/W	
4	—	0	R/W	
3	P23DR	0	R/W	P2DR stores output data for the port 2 pin used as the general output port. If a port 2 read is performed while the P2D are set to 1, the P2DR values are read. If read is performed while the P2DDR bits a cleared to 0, the pin states are read.
2	P22DR	0	R/W	
1	P21DR	0	R/W	
0	P20DR	0	R/W	

2	P22PCR	0	R/W	input pull-up MOS is turned on when a P is set to 1.
1	P21PCR	0	R/W	
0	P20PCR	0	R/W	

8.2.4 Port 2 Input Pull-Up MOS

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

Table 8.3 Port 2 Input Pull-Up MOS States

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

[Legend]
Off: Always off.
On/Off: On when P2DDR = 0 and P2PCR = 1; otherwise off.

- Noise cancel cycle setting register (NCCS)

8.3.1 Port 3 Data Direction Register (P3DDR)

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

Bit	Bit Name	Initial Value	R/W	Description
7	P37DDR	0	W	The corresponding port 3 pins function as outp when the P3DDR bits are set to 1, and as inpu when cleared to 0.
6	P36DDR	0	W	
5	P35DDR	0	W	
4	P34DDR	0	W	
3	P33DDR	0	W	
2	P32DDR	0	W	
1	P31DDR	0	W	
0	P30DDR	0	W	

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

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	When the pins are in input state, the correspo input pull-up MOS is turned on when a P3PCI to 1.
6	P36PCR	0	R/W	
5	P35PCR	0	R/W	
4	P34PCR	0	R/W	
3	P33PCR	0	R/W	
2	P32PCR	0	R/W	
1	P31PCR	0	R/W	
0	P30PCR	0	R/W	

2	P32NCE	0	R/W
1	P31NCE	0	R/W
0	P30NCE	0	R/W

8.3.5 Noise Canceler Mode Control Register (P3NCMC)

P3NCMC controls whether 1 or 0 is expected for the input signal to port 3 in bit units.

Bit	Bit Name	Initial Value	R/W	Description
7	P37NCMC	1	R/W	1 expected: 1 is stored in the port data register is input stably 0 expected: 0 is stored in the port data register is input stably
6	P36NCMC	1	R/W	
5	P35NCMC	1	R/W	
4	P34NCMC	1	R/W	
3	P33NCMC	1	R/W	
2	P32NCMC	1	R/W	
1	P31NCMC	1	R/W	
0	P30NCMC	1	R/W	

000:	80 ns	ϕ/2
001:	1.28 μs	ϕ/32
010:	20.48 μs	ϕ/512
011:	327.7 μs	ϕ/8192
100:	1.31 ms	ϕ/32768
101:	2.62 ms	ϕ/65536
110:	5.24 ms	ϕ/131072
111:	10.49 ms	ϕ/262144

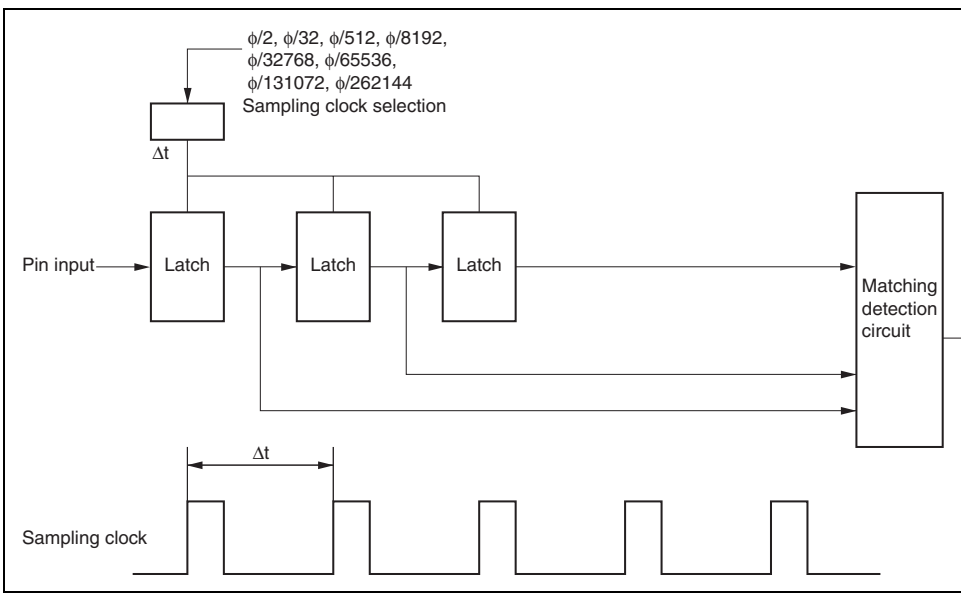


Figure 8.1 Noise Canceler Circuit

(n = 7 to 0)

Figure 8.2 Noise Canceler Operation

8.3.7 Pin Functions

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

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

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

8.3.8 Port 3 Input Pull-Up MOS

Port 3 has a built-in input pull-up MOS that can be controlled by software. This input pul
MOS can be used in single-chip mode. Table 8.4 summarizes the input pull-up MOS stat

Table 8.4 Port 3 Input Pull-Up MOS States

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

[Legend]
Off: Always off.
On/Off: On when input state and P3PCR = 1; otherwise off.

- Noise cancel cycle setting register (NCCS)

8.4.1 Port 4 Data Direction Register (P4DDR)

The individual bits of P4DDR specify input or output for the pins of port 4.

These bits are initialized only by a system reset, and retain their values even when an int
signal is generated by the WDT.

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

5	P45DR	0	R/W	set to 1, the P4DR values are read. If a port 4
4	P44DR	0	R/W	performed while the P4DDR bits are cleared to
3	P43DR	0	R/W	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 input state, the correspon
6	P46PCR	0	R/W	input pull-up MOS is turned on when a P4PCR
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	

8.4.5 Noise Canceler Mode Control Register (P4NCMC)

P4NCMC controls whether 1 or 0 is expected for the input signal to port 4 in bit units.

Bit	Bit Name	Initial Value	R/W	Description
7	P47NCMC	1	R/W	1 expected: 1 is stored in the port data registe is input stably 0 expected: 0 is stored in the port data registe is input stably
6	P46NCMC	1	R/W	
5	P45NCMC	1	R/W	
4	P44NCMC	1	R/W	
3 to 0	—	All 1	R/W	Reserved

000:	80 ns	ϕ/2
001:	1.28 μs	ϕ/32
010:	20.48 μs	ϕ/512
011:	327.7 μs	ϕ/8192
100:	1.31 ms	ϕ/32768
101:	2.62 ms	ϕ/65536
110:	5.24 ms	ϕ/131072
111:	10.49 ms	ϕ/262144

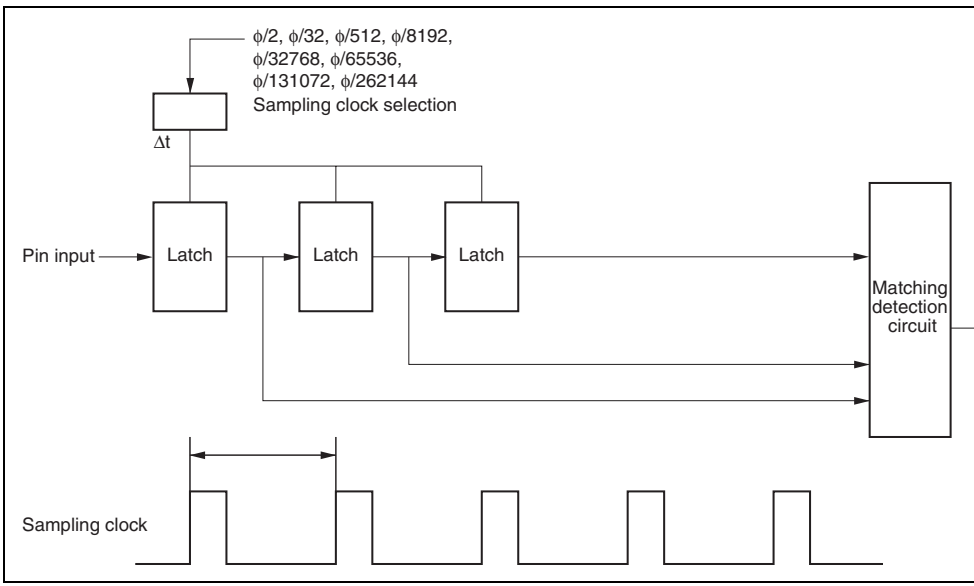


Figure 8.3 Noise Canceler Circuit

(n = 7 to 4)

Figure 8.4 Noise Canceler Operation

8.4.7 Pin Functions

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

- P47 to P44

The pin function is switched as shown below according to the P4nDDR bit and P4nNCE

When the ISSn bit in ISSR is cleared to 0 and the IRQnE bit in IER of the interrupt cont
set to 1, this pin can be used as the IRQn input pin. To use this pin as the IRQn input pin
P4nDDR bit to 0.

P4nDDR	0		1
P4nNCE	0	1	x
Pin function	P4n input pin	DBn input	P4n outpu
	$\overline{\text{IRQn}}$ input pin	$\overline{\text{IRQn}}$ input pin (with the noise canceler)	

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

n = 3 to 0

8.4.8 Port 4 Input Pull-Up MOS

Port 4 has a built-in input pull-up MOS that can be controlled by software. This input pul
MOS can be used in single-chip mode. Table 8.5 summarizes the input pull-up MOS stat

Table 8.5 Port 4 Input Pull-Up MOS States

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

[Legend]
Off: Always off.
On/Off: On when input state and P4PCR = 1: otherwise off.

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

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 port 5 pins function port when the P5DDR bits are set to 1, and port when cleared to 0.
6	P56DDR	0	W	The corresponding port 5 pin functions as th clock output pin (ϕ) when this bit is set to 1, the general I/O port when cleared to 0.
5, 4	—	All 0	W	Reserved
3	P53DDR	0	W	If port 5 pins are specified for use as the ge port, the corresponding port 5 pins function port when the P5DDR bits are set to 1, and port when cleared to 0.
2	P52DDR	0	W	
1, 0	—	All 0	W	Reserved

5, 4	—	All 0	R/W	Reserved
3	P53DR	0	R/W	See the description of bits 7 and 6.
2	P52DR	0	R/W	
1, 0	—	All 0	R/W	Reserved

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

8.5.3 Pin Functions

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

- 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

Pin function	P56 input pin	EXCL input pin	ϕ output pin	PWX0 o
	$\overline{\text{IRQ14}}$ 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 of SCI_1, the SMIF bit in SCMR, and the P53DDR bit.

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

RE	x		0	1
SMIF	0		1	
P53DDR	0	1	x	x
Pin function	P53 input pin	P53 output pin	RxD1 input pin	RxD1 input
	$\overline{\text{IRQ11}}$ input pin			

[Legend]

x: Don't care

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

[Legend]

x: Don't care

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

Bit	Bit Name	Initial Value	R/W	Description
7	—	0	W	Reserved
6	—	0	W	
5	—	0	W	
4	—	0	W	
3	P63DDR	0	W	If port 6 pins are specified for use as the ge port, the corresponding port 6 pins function port when the P6DDR bits are set to 1, and ports when cleared to 0.
2	P62DDR	0	W	
1	P61DDR	0	W	
0	P60DDR	0	W	

2	P62DR	0	R/W	used as the general output port.
1	P61DR	0	R/W	If a port 6 read is performed while the P6DDR set to 1, the P6DR values are read. If a port 6 performed while the P6DDR bits are cleared pin states are read.
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 to 4	—	All 0	W	Reserved
3	P63PCR	0	W	When the pins are in input state, the correspo input pull-up MOS is turned on when a P6PC set to 1.
2	P62PCR	0	W	
1	P61PCR	0	W	
0	P60PCR	0	W	

Off: Always off.

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

Bit	Bit Name	Initial Value	R/W	Description
7	P77PIN	Undefined*	R	When a P7PIN read is performed, the pin sta always read.
6	P76PIN	Undefined*	R	
5	P75PIN	Undefined*	R	
4	P74PIN	Undefined*	R	
3	P73PIN	Undefined*	R	
2	P72PIN	Undefined*	R	
1	P71PIN	Undefined*	R	
0	P70PIN	Undefined*	R	

Note: * The initial value is determined in accordance with the pin states of P77 to P70.

not set these bits to other values than those shown in the following table.

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

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

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

[Legend]

x: Don't care

Pin function	AN3 input pin	P73 input pin	ExIRQ3 input pin	AN3 input pin	P73 input pin	i

[Legend]

x: Don't care

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

  The pin function is switched as shown below according to the combination of the CH bits in ADCSR of the A/D converter, the SCANE bit in ADCR, and the ISS4 bit in IS interrupt controller. Do not set these bits to other values than those shown in the follo 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	$\overline{\text{E}}$ i

[Legend]

x: Don't care

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 CH bits in ADCSR of the A/D converter, the SCANE and SCANS bits in ADCR, and th in ISSR of the interrupt controller. Do not set these bits to other values than those sh 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 input 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 CH
  bits in ADCSR of the A/D converter, the SCANE and SCANS bits in ADCR, and the
  in ISSR of the interrupt controller. Do not set these bits to other values than those sho
  following table.

SCANE	0			1			
SCANS	x			0			
CH2 to CH0	B'000	Other than B'000		B'000 to B'011	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	A

[Legend]

x: Don't care

0	x	1	1	0	—	ON	—	—	—	—	—
0	x	1	1	1	ON	—	—	—	—	—	—
1	0	0	0	0	—	—	—	—	—	—	—
1	0	0	0	1	—	—	—	—	—	—	ON
1	0	0	1	0	—	—	—	—	—	ON	ON
1	0	0	1	1	—	—	—	—	ON	ON	ON
1	0	1	0	0	—	—	—	ON	—	—	—
1	0	1	0	1	—	—	ON	ON	—	—	—
1	0	1	1	0	—	ON	ON	ON	—	—	—
1	0	1	1	1	ON	ON	ON	ON	—	—	—
1	1	0	0	0	—	—	—	—	—	—	—
1	1	0	0	1	—	—	—	—	—	—	ON
1	1	0	1	0	—	—	—	—	—	ON	ON
1	1	0	1	1	—	—	—	—	ON	ON	ON
1	1	1	0	0	—	—	—	ON	ON	ON	ON
1	1	1	0	1	—	—	ON	ON	ON	ON	ON
1	1	1	1	0	—	ON	ON	ON	ON	ON	ON
1	1	1	1	1	ON	ON	ON	ON	ON	ON	ON

[Legend]

x: Don't care

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

Bit	Bit Name	Initial Value	R/W	Description
7	P87DDR	0	W	If port 8 pins are specified for use as the gen port, the corresponding port 8 pins function a port when the P8DDR bits are set to 1, and a port when cleared to 0.
6	P86DDR	0	W	
5	P85DDR	0	W	
4	P84DDR	0	W	
3	P83DDR	0	W	
2	P82DDR	0	W	
1	P81DDR	0	W	
0	P80DDR	0	W	

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

8.8.3 Pin Functions

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

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

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

  When the TRGS1 and EXTRGS bits are both set to 1 and the TRGS0 bit is cleared 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		
SMIF	0	1	0	1	0
TE	0	x	0	x	1
Pin function	P87 input pin		P87 input pin		TxD3 out
	$\overline{\text{ExIRQ15}}$ input pin/ $\overline{\text{ADTRG}}$ input pin				

[Legend]

x: Don't care

	$\overline{\text{ExIRQ14}}$ input pin		pin	

- P85/$\overline{\text{ExIRQ13}}$/SCK1

  The pin function is switched as shown below according to the combination of the C/$\overline{\text{A}}$
  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/$\overline{\text{A}}$	0			1	
CKE0	0		1	x	
P85DDR	0	1	x	x	
Pin function	P85 input pin $\overline{\text{ExIRQ13}}$ input pin	P85 output pin	SCK1 output pin	SCK1 output pin	SCI

[Legend]

x: Don't care

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

[Legend]

x: Don't care

- P83/$\overline{\text{ExIRQ11}}$/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 the ISS11 bit in ISSR16 is set to 1, this pin can be used as the $\overline{\text{ExIRQ11}}$ input
  use this pin as the $\overline{\text{ExIRQ11}}$ input pin, clear the P83DDR bit to 0.

  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, and direct bus drive is possible

ICE	0		1
P83DDR	0	1	x
Pin function	P83 input pin	P83 output pin	SDA1 input/o
	$\overline{\text{ExIRQ11}}$ input pin		

[Legend]

x: Don't care

Pin function	P82 input pin	P82 output pin	SCL1 input/ou
	$\overline{\text{ExIRQ10}}$ input pin		

[Legend]

x: Don't care

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

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

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

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

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

[Legend]

x: Don't care

Pin function	P80 input pin	P80 output pin	SCL0 input/o
	$\overline{\text{ExIRQ8}}$ input pin		

[Legend]

x: Don't care

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

Bit	Bit Name	Initial Value	R/W	Description
7	PA7DDR	0	W	The corresponding port A pins function as ou when the PADDR bits are set to 1, and as inp when cleared to 0. This register is assigned to the same address of PAPIN. When this address is read, the por states are returned.
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

8.9.3 Port A Input Data Register (PAPIN)

PAPIN indicates the pin states.

Bit	Bit Name	Initial Value	R/W	Description
7	PA7PIN	Undefined*	R	When a PAPIN read is performed, the pin s always read. This register is assigned to the same addres of PADDR. When this register is written to, written to PADDR and the port A setting is t changed.
6	PA6PIN	Undefined*	R	
5	PA5PIN	Undefined*	R	
4	PA4PIN	Undefined*	R	
3	PA3PIN	Undefined*	R	
2	PA2PIN	Undefined*	R	
1	PA1PIN	Undefined*	R	
0	PA0PIN	Undefined*	R	

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

When this pin is used as EVENT input pin according to bits ECSB3 to ECSB0 in EC
data transfer controller settings, clear the PAnDDR bit to 0. Though this pin has been
EVENT input pin, to use as the PAn output pin, set the PAnDDR bit to 1.

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

[Legend]
n = 7 to 0

x: Don't care

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

Table 8.8 summarizes the input pull-up MOS states.

Table 8.8 PortA Input Pull-Up MOS States

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

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

8.10.1 Port C Data Direction Register (PCDDR)

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

Bit	Bit Name	Initial Value	R/W	Description
7	PC7DDR	0	W	The corresponding port C pins function as ou when the PCDDR bits are set to 1, and as inp when cleared to 0. This register is assigned to the same address of PCPIN. When this address is read, the por states are returned. Bits 5 and 4 are reserved.
6	PC6DDR	0	W	
5	—	0	W	
4	—	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

8.10.3 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 state is re
6	PC6PIN	Undefined*	R	This register is assigned to the same addres of PCDDR. When this register is written to, written to PCDDR and the port C setting is t changed.
5	—	Undefined	R	
4	—	Undefined	R	
3	PC3PIN	Undefined*	R	Bits 5 and 4 are reserved.
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, PC6, to PC0 pins.

Pin function	PC7 input pin	PC7 output pin	PWX3 outpu

[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 14-bit PWMX and the PC6DDR bit.

OEA	0		1
PC6DDR	0	1	x
Pin function	PC6 input pin	PC6 output pin	PWX2 outpu

[Legend]
x: Don't care

- PC3/SDA3

  The pin function is switched as shown below according to the combination of the ICE
  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/out

[Legend]
x: Don't care

- PC1/SDA2

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

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

[Legend]
x: Don't care

- PC0/SCL2

  The pin function is switched as shown below according to the combination of the IC
  ICCR of the IIC_2 and the PC0DDR bit.

ICE	0		1
PC0DDR	0	1	x
Pin function	PC0 input pin	PC0 output pin	SCL2 input/o

[Legend]
x: Don't care

PEDDR is used to specify the input/output attribute of each pin of port E.

Bit	Bit Name	Initial Value	R/W	Description
7	PE7DDR	0	W	The corresponding port E pins function as ou when the PEDDR bits are set to 1, and as inp when cleared to 0. This register is assigned to the same address of PEPIN. When this address is read, the por states are returned.
6	PE6DDR	0	W	
5	PE5DDR	0	W	
4	PE4DDR	0	W	
3	PE3DDR	0	W	
2	PE2DDR	0	W	
1	PE1DDR	0	W	
0	PE0DDR	0	W	

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

8.11.3 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	Pin states can be read by performing a reac this register. This register is assigned to the same addres of PEDDR. When this register is written to, written to PEDDR and the port E setting is t changed.
6	PE6PIN	Undefined*	R	
5	PE5PIN	Undefined*	R	
4	PE4PIN	Undefined*	R	
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 pi 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 the LPC enabled/disabled a
  PE6DDR bit.

LPC	Disabled		Enablec
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 the LPC enabled/disabled a
  PE5DDR bit.

LPC	Disabled		Enablec
PE5DDR	0	1	x
Pin function	PE5 input pin	PE5 output pin	$\overline{\text{LRESET}}$ inpu

[Legend]

x: Don't care

- PE3/LAD3

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

- PE0/LAD0

  The pin function is switched as shown below according to the LPC enabled/disabled a
  PE0DDR bit.

LPC	Disabled		Enablec
PE0DDR	0	1	x
Pin function	PE0 input pin	PE0 output pin	LAD0 input/out

[Legend]

x: Don't care

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

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

- ISSR16

Bit	Bit Name	Initial Value	R/W	Description
15	ISS15	0	R/W	0: P57/$\overline{\text{IRQ15}}$ is selected 1: P87/$\overline{\text{ExIRQ15}}$ is selected
14	ISS14	0	R/W	0: P56/$\overline{\text{IRQ14}}$ is selected 1: P86/$\overline{\text{ExIRQ14}}$ is selected
13	ISS13	0	R/W	P85/$\overline{\text{ExIRQ13}}$ is always selected
12	ISS12	0	R/W	P84/$\overline{\text{ExIRQ12}}$ is always 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	P81/$\overline{\text{ExIRQ9}}$ is always selected
8	ISS8	0	R/W	P80/$\overline{\text{ExIRQ8}}$ is always 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

- Two base cycle settings

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

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

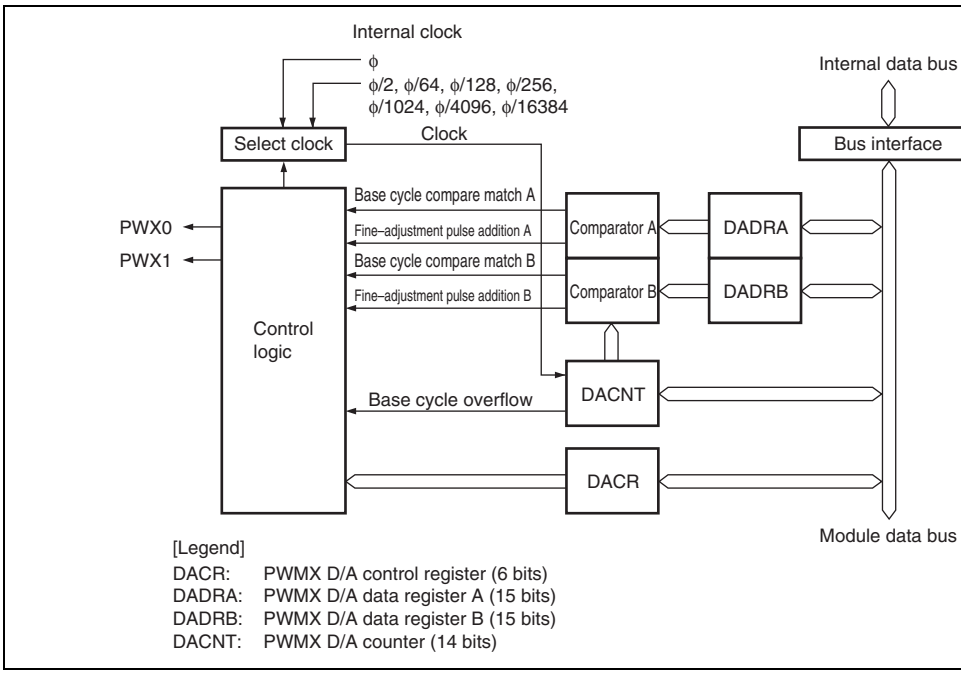


Figure 9.1 PWMX (D/A) Block Diagram

PWMX output pin 3	PWX3	Output	PWM timer pulse output of PWMX_1 ch

9.3 Register Descriptions

The PWMX (D/A) module has the following registers. For details on the module stop con register, see section 22.1.3, 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 bi which registers can be accessed. When chan register to be accessed, set this bit in advanc 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 contin compared with the DADR value to determine th cycle of the output waveform, and to decide wh output a fine-adjustment pulse equal in width to resolution. To enable this operation, this regist be set within a range that depends on the CFS DADR value is outside this range, the PWM ou held constant. A channel can be operated with 12-bit precisio fixing DA0 and DA1 to 0. The two data bits are compared with UC12 and UC13 of DACNT.
1	CFS	1	R/W	Carrier Frequency Select 0: Base cycle = resolution (T) × 64 The range of DA13 to DA0: H'0100 to H'3FF 1: Base cycle = resolution (T) × 256 The range of DA13 to DA0: H'0040 to H'3FF
0	—	1	R	Reserved This bit is always read as 1 and cannot be mod

				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) chann 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) chann 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
```

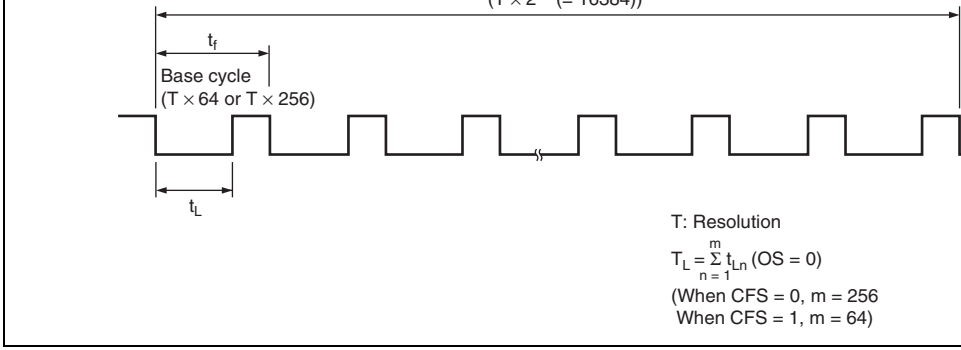


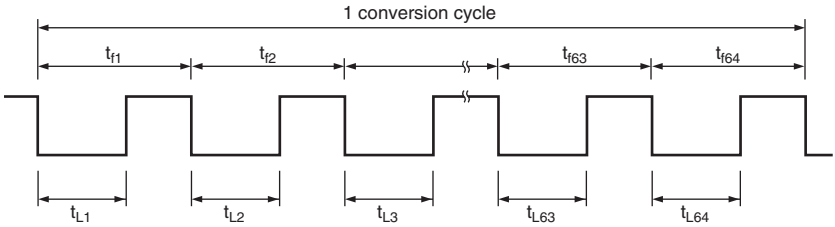
Figure 9.2 PWMX (D/A) Operation

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.

						97.7 kHz		DA13 to 0 = H'0000 to H'003F	12			0	0		
								(Data value) × T							
				(ϕ)				DA13 to 0 = H'0040 to H'3FFF	10	0	0	0	0		
0	0	0	1	0.08	0	5.12 μs/	1.311ms	Always low/high output	14						
						195.3 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	20.48 μs/	1.311 ms	Always low/high output	14						
						48.8 kHz		DA13 to 0 = H'0000 to H'003F	12			0	0		
								(Data value) × T							
				(ϕ/2)				DA13 to 0 = H'0040 to H'3FFF	10	0	0	0	0		
0	0	1	1	2.56	0	163.8 μs/	41.943 ms	Always low/high output	14						
						6.1 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	655.4 μs/	41.943 ms	Always low/high output	14						
						1.5 kHz		DA13 to 0 = H'0000 to H'003F	12			0	0		
								(Data value) × T							
				(ϕ/64)				DA13 to 0 = H'0040 to H'3FFF	10	0	0	0	0		
0	1	0	1	5.12	0	327.7 μs/	83.886 ms	Always low/high output	14						
						3.1 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	1310.7 μs/	83.886 ms	Always low/high output	14						
						0.8 kHz		DA13 to 0 = H'0000 to H'003F	12			0	0		
								(Data value) × T							
				(ϕ/128)				DA13 to 0 = H'0040 to H'3FFF	10	0	0	0	0		

				(ϕ/256)				DA13 to 0 = H'0040 to H'3FFF	10	0	0	0	0
1	0	0	1	40.96	0	2.62 ms/	671.09 ms	Always low/high output	14				
						381.5 Hz		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	10.49 ms/	671.09 ms	Always low/high output	14				
						95.4 Hz		DA13 to 0 = H'0000 to H'003F	12			0	0
								(Data value) × T					
				(ϕ/1024)				DA13 to 0 = H'0040 to H'3FFF	10	0	0	0	0
1	0	1	1	163.84	0	10.49 ms/	2.684 s	Always low/high output	14				
						95.4 Hz		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	41.94 ms/	2.684 s	Always low/high output	14				
						23.8 Hz		DA13 to 0 = H'0000 to H'003F	12			0	0
								(Data value) × T					
				(ϕ/4096)				DA13 to 0 = H'0040 to H'3FFF	10	0	0	0	0
1	1	0	1	655.36	0	41.94 ms/	10.737 s	Always low/high output	14				
						23.8 Hz		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	167.77 ms/	10.737 s	Always low/high output	14				
						6.0 Hz		DA13 to 0 = H'0000 to H'003F	12			0	0
								(Data value) × T					
				(ϕ/16384)				DA13 to 0 = H'0040 to H'3FFF	10	0	0	0	0
1	1	1	1	Setting prohibited	—	—	—	—	—	—	—	—	—

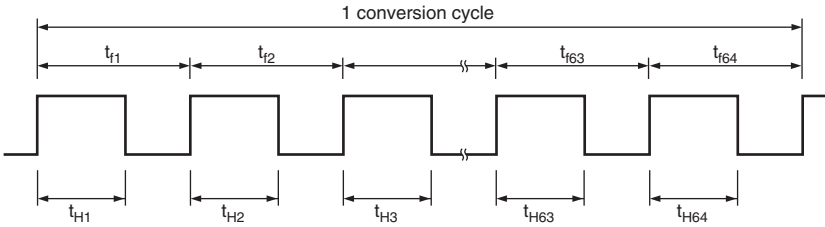
Note: * Indicates the conversion cycle when specific DA3 to DA0 bits are fixed.



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

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

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



$t_{f1} = t_{f2} = t_{f3} = \cdots = t_{f63} = t_{f64} = T \times 256$
$t_{H1} + t_{H2} + t_{H3} + \cdots + t_{H63} + t_{H64} = T_H$

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

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.

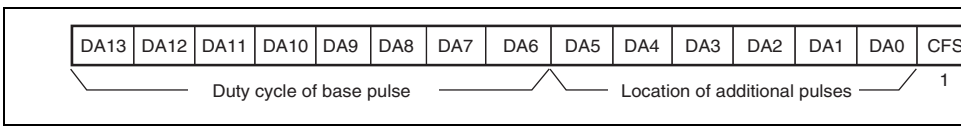


Figure 9.5 D/A Data Register Configuration when CFS = 1

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).

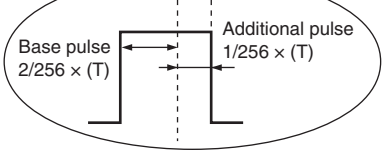


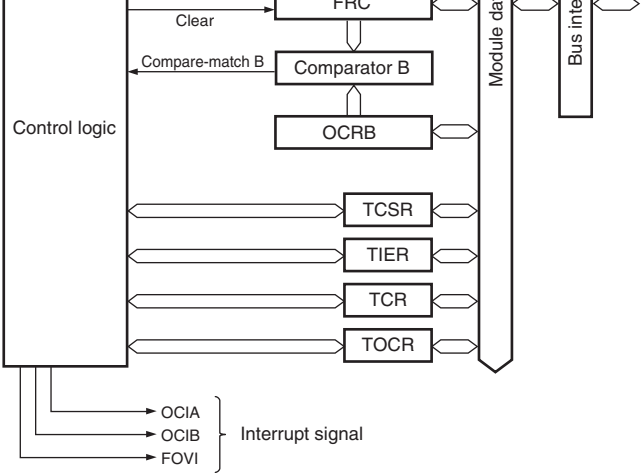
Figure 9.6 Output Waveform when DADR = H'0207 (OS = 1)

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

— 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.

input clock together with a set value of H'0001 or less for OCRAR (or OCRAF).

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

				request (OCIA) when output compare flag A (OC TCSR is set to 1. 0: OCIA requested by OCFA is disabled 1: OCIA requested by OCFA is enabled
2	OCIBE	0	R/W	Output Compare Interrupt B Enable Selects whether to enable output compare intern 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 c 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

				OCRA value. [Setting condition] When FRC = OCRA [Clearing condition] Read OCFA when OCFA = 1, then write 0 to OCI
2	OCFB	0	R/(W)*	Output Compare Flag B This status flag indicates that the FRC value mat OCRB value. [Setting condition] When FRC = OCRB [Clearing condition] Read OCFB when OCFB = 1, then write 0 to OCI
1	OVF	0	R/(W)*	Overflow Flag This status flag indicates that the FRC has overfl [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 This bit selects whether the FRC is to be cleared compare-match A (when the FRC and OCRA val match). 0: FRC clearing is disabled 1: FRC is cleared at compare-match A

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

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

				Specifies whether OCRA is used in the normal op mode or in the operating mode using OCRAR and OCRAF. 0: The normal operating mode is specified for OC 1: The operating mode using OCRAR and OCRA specified for OCRA
5	ICRS	0	R/W	Input Capture Register Select Controls the access to OCRAR and OCRAF. 0: Access is disabled. 1: Access is enabled.
4	OCRS	0	R/W	Output Compare Register Select OCRA and OCRB share the same address. Wher 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

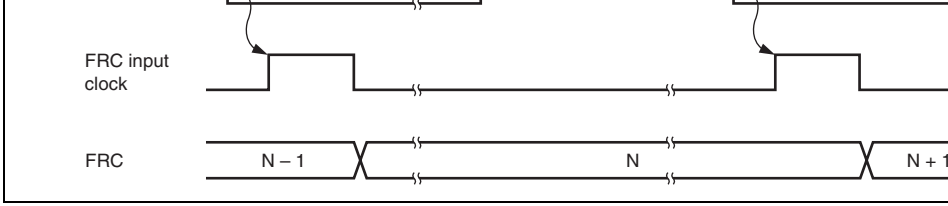


Figure 10.2 Increment Timing with Internal Clock Source

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.

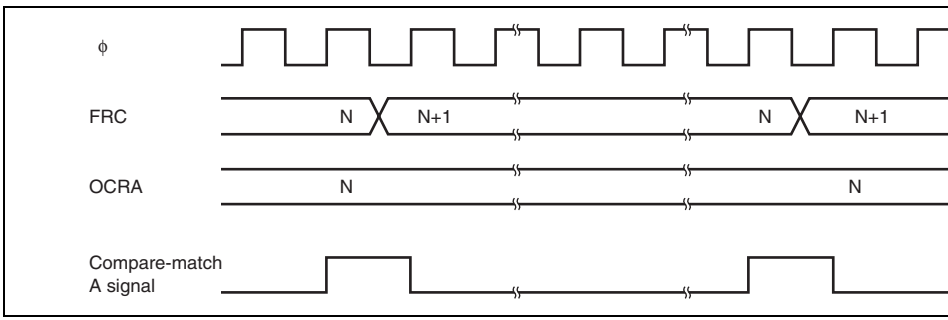


Figure 10.3 Timing of Output Compare A Output

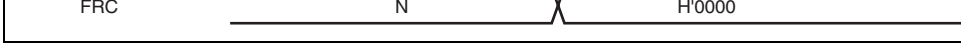


Figure 10.4 Clearing of FRC by Compare-Match A Signal

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

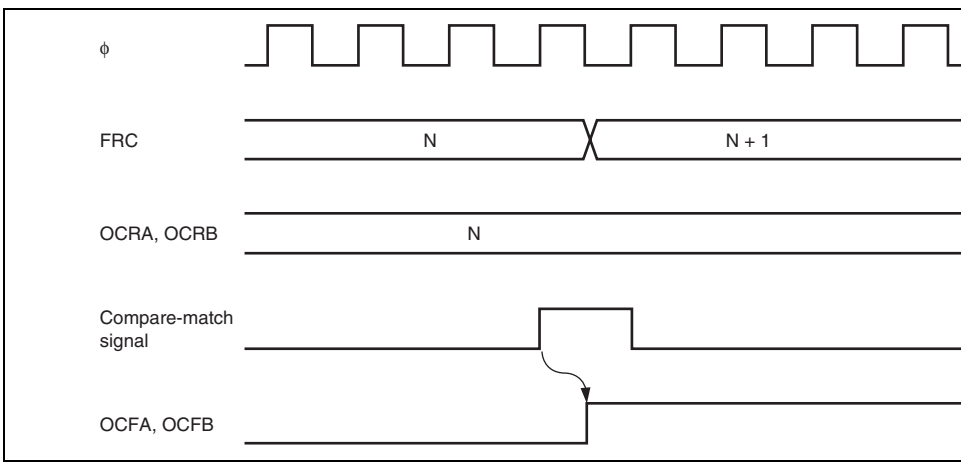


Figure 10.5 Timing of Output Compare Flag (OCFA or OCFB) Setting

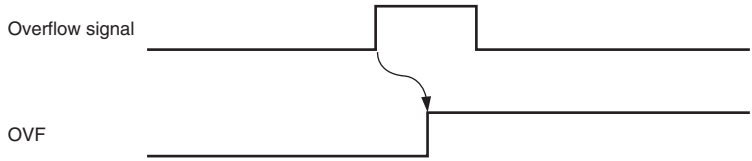


Figure 10.6 Timing of Overflow Flag (OVF) Setting

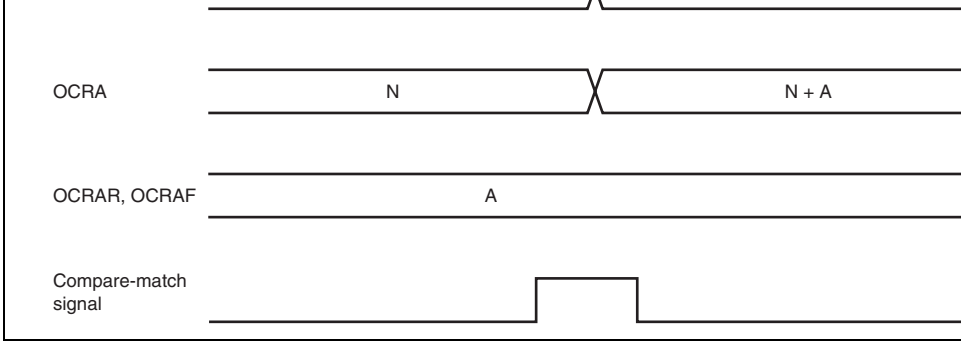


Figure 10.7 OCRA Automatic Addition Timing

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 intern
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	Pric
OCIA	Compare match of OCRA	OCFA	Possible	Hig
OCIB	Compare match of OCRB	OCFB	Possible	↑
FOVI	Overflow of FRC	OVF	Not possible	Low

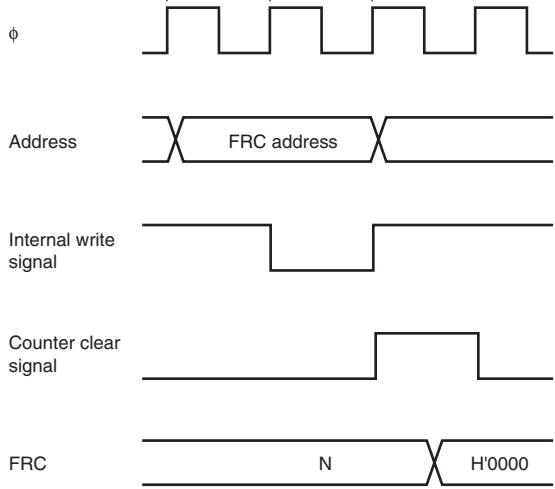


Figure 10.8 Conflict between FRC Write and Clear

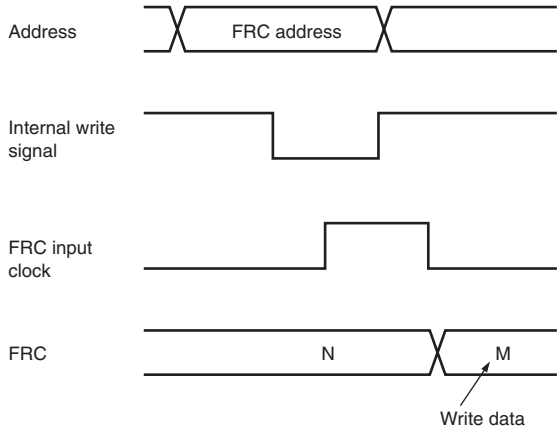


Figure 10.9 Conflict between FRC Write and Increment

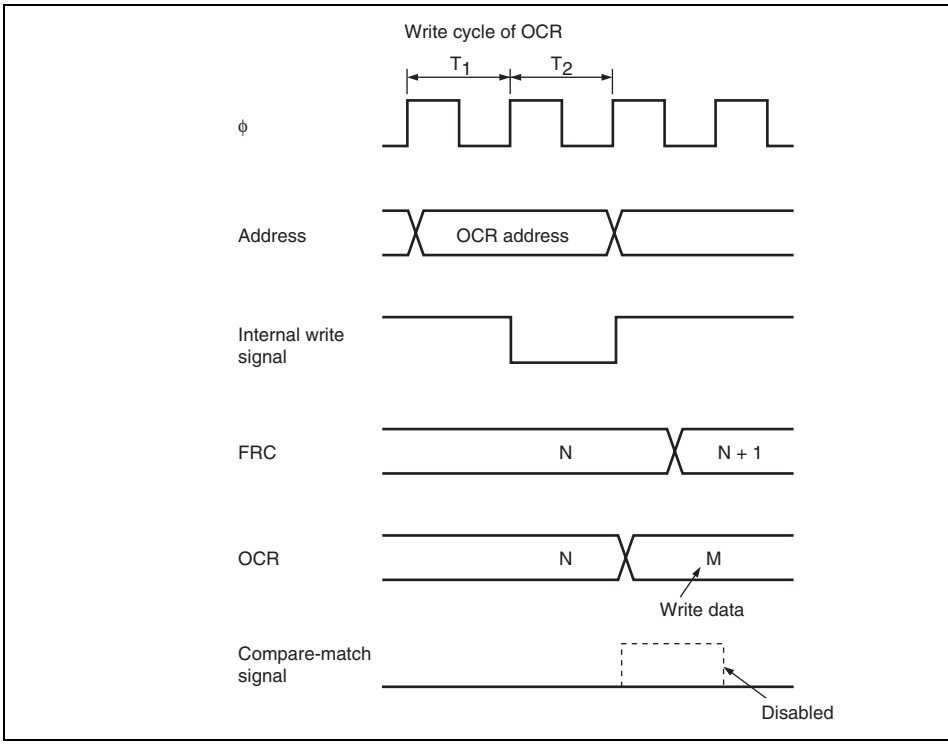


Figure 10.10 Conflict between OCR Write and Compare-Match (When Automatic Addition Function is Not Used)

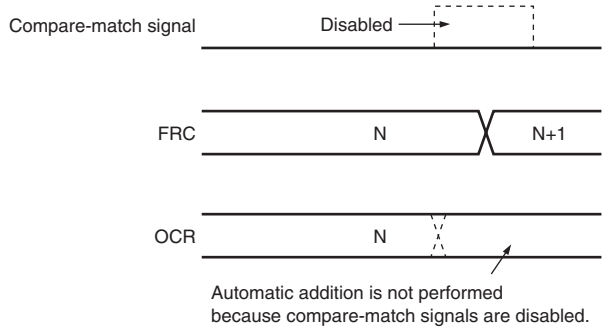
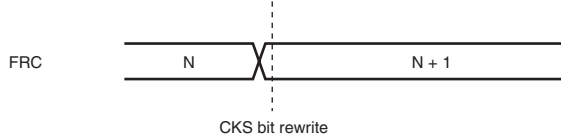
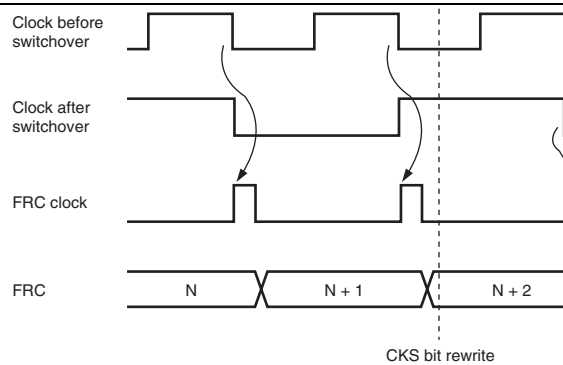
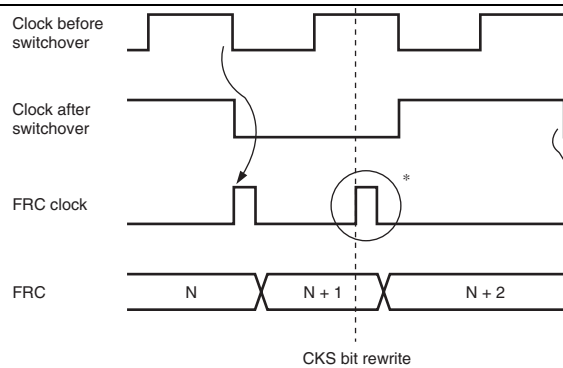


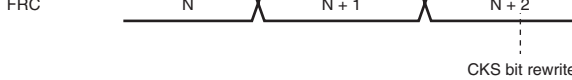
Figure 10.11 Conflict between OCR Write and Compare-Match (When Automatic Addition Function is Used)

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.

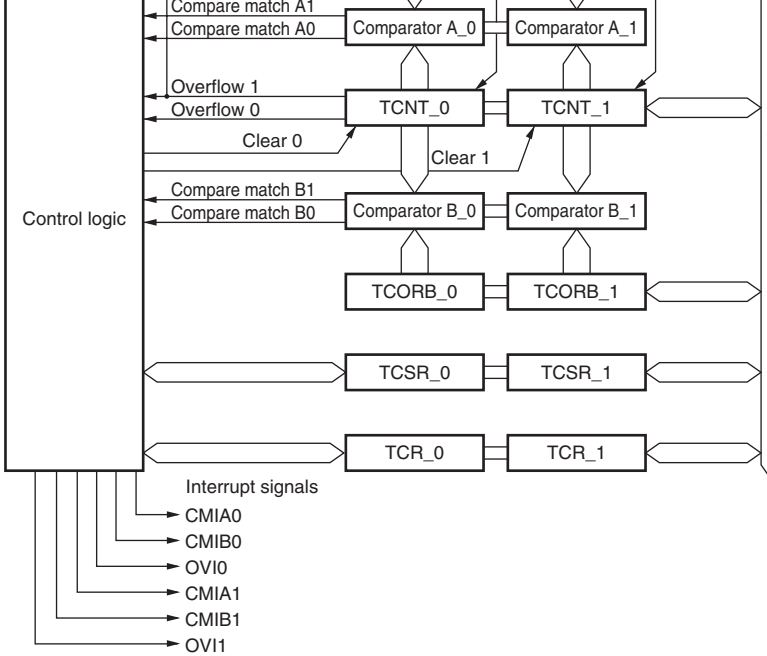
		
2	Switching from low to high	
3	Switching from high to low	



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

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

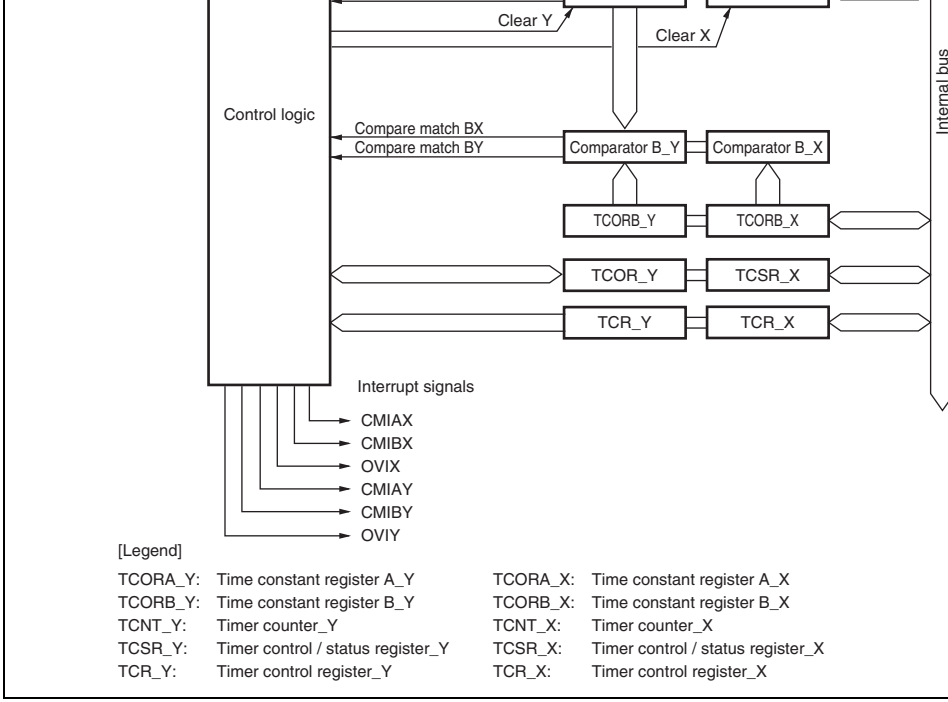
- Selection of two ways to clear the counters
  — The counters can be cleared on compare-match A and compare-match B.
- Cascading of TMR_0 and TMR_1
  (Cascading of TMR_Y and TMR_X is not allowed)
  — Operation as a 16-bit timer can be performed using TMR_0 as the upper half and as the lower half (16-bit 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, and TMR_Y: One interrupt: Overflow
  — TMR_X: Three interrupts: Compare-match A, compare-match B, and



[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

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.

* Only for the 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 in
When TCNT overflows (changes from H'FF to H'00), the OVF bit in TCSR is set to 1. TC
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. Note however that comparison is disabled during the T2 state of a T write cycle. 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 (CMIB) enabled or disabled when the CMFB flag in TCSR is 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 (CMIA) enabled or disabled when the CMFA flag in TCSR is 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) is e or disabled when the OVF flag in TCSR is set to 1. 0: OVF interrupt request (OVI) is disabled 1: OVF interrupt request (OVI) is enabled
4	CCLR1	0	R/W	Counter Clear 1 and 0
3	CCLR0	0	R/W	Specify the cleaning 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 These bits select the clock input to TCNT and count condition, together with the ICKS1 and ICKS0 bits in For details, see table 11.1.

0	1	1	1	Increments at falling edge of internal cloc
1	0	0	—	Increments at overflow signal from TCNT

Table 11.1 (2) Clock Input to TCNT and Count Condition (TMR_1)

TCR			STCR	
CKS2	**CKS1**	**CKS0**	**ICKS1**	**Description**
0	0	0	—	Disables clock input
0	0	1	0	Increments at falling edge of internal cloc
0	0	1	1	Increments at falling edge of internal cloc
0	1	0	0	Increments at falling edge of internal cloc
0	1	0	1	Increments at falling edge of internal cloc
0	1	1	0	Increments at falling edge of internal cloc
0	1	1	1	Increments at falling edge of internal cloc
1	0	0	—	Increments at compare-match A from TC

	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	0	0	Setting prohibited
Common	1	0	1	Setting prohibited
	1	1	0	Setting prohibited
	1	1	1	Setting prohibited

				When the values of TCNT_0 and TCORB_0 ma [Clearing condition] Read CMFB when CMFB = 1, then write 0 in CM
6	CMFA	0	R/(W)*	Compare-Match Flag A [Setting condition] When the values of TCNT_0 and TCORA_0 ma [Clearing condition] Read CMFA when CMFA = 1, then write 0 in CM
5	OVF	0	R/(W)*	Timer Overflow Flag [Setting condition] When TCNT_0 overflows from H′FF to H′00 [Clearing condition] Read OVF when OVF = 1, then write 0 in OVF
4	ADTE	0	R/W	A/D Trigger Enable Selects whether the A/D conversion start reques compare match A is enabled or disabled. 0: A/D conversion start request is disabled 1: A/D conversion start request is enabled
3 to 0	—	All 1	R	Reserved These bits are always read as 1 and cannot be

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

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

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

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

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

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

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

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

Table 11.2 Registers Accessible by TMR_X/TMR_Y

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

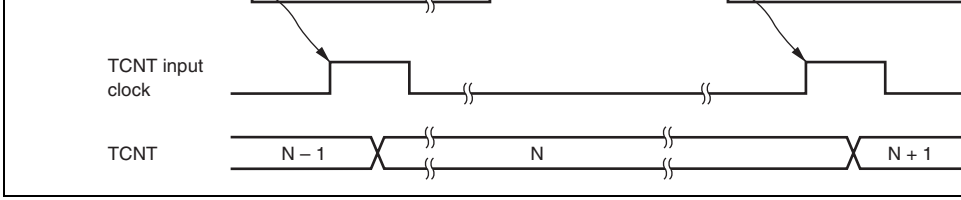


Figure 11.3 Count Timing for Internal Clock Input

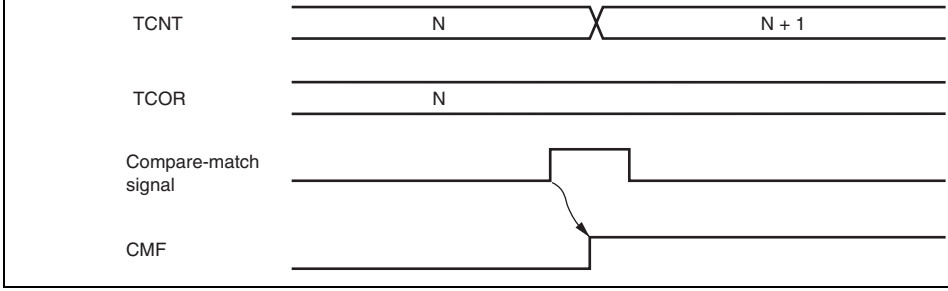


Figure 11.4 Timing of CMF Setting at Compare-Match

11.3.3 Timing of Counter Clear at Compare-Match

TCNT is cleared when compare-match A or compare-match B occurs, depending on the
the CCLR1 and CCLR0 bits in TCR. Figure 11.5 shows the timing of clearing the count
compare-match.

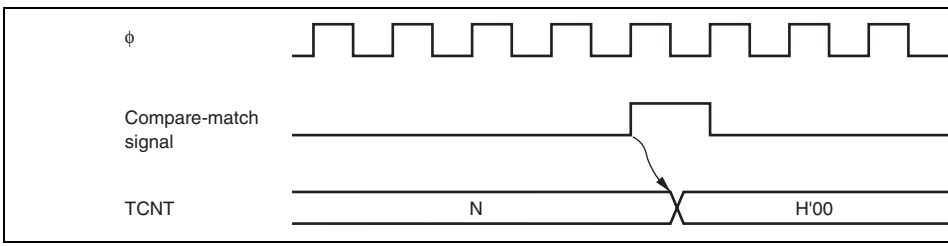


Figure 11.5 Timing of Counter Clear by Compare-Match

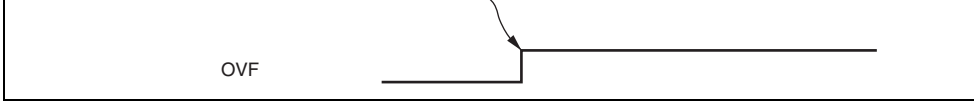


Figure 11.6 Timing of OVF Flag Setting

- 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 comp the 16-bit counter (TCNT_0 and TCNT_1 together) is cleared when a 16-bit com match occurs. The 16-bit counter (TCNT_0 and TCNT_1 together) is also cleare 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 b 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 comp A for TMR_0. TMR_0 and TMR_1 are controlled independently. Conditions such as se CMF flag, generation of interrupts, and counter clearing are in accordance with the setti each channel.

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

Channel	Name	Interrupt Source	Interrupt Flag	DTC Activation	Inter Prio
TMR_X	CMIAX	TCORA_X compare-match	CMFA	Possible	High
	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	Low

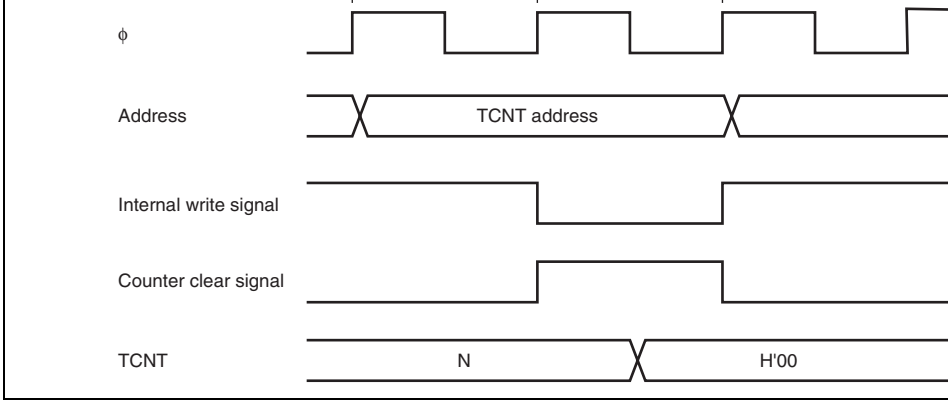


Figure 11.7 Conflict between TCNT Write and Counter Clear

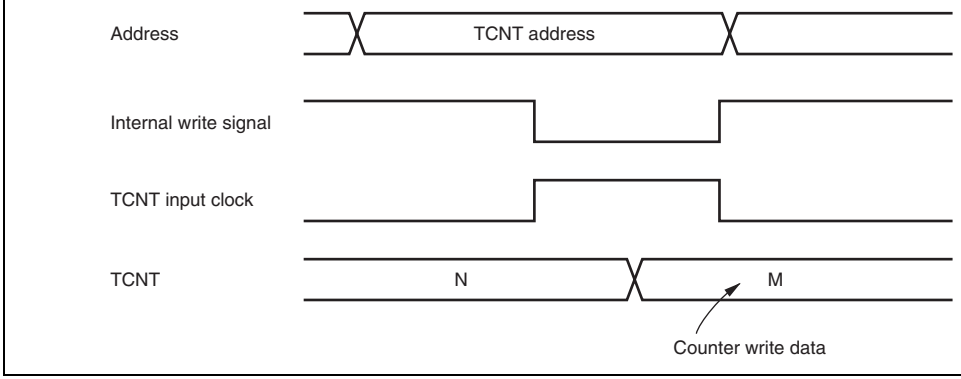


Figure 11.8 Conflict between TCNT Write and Increment

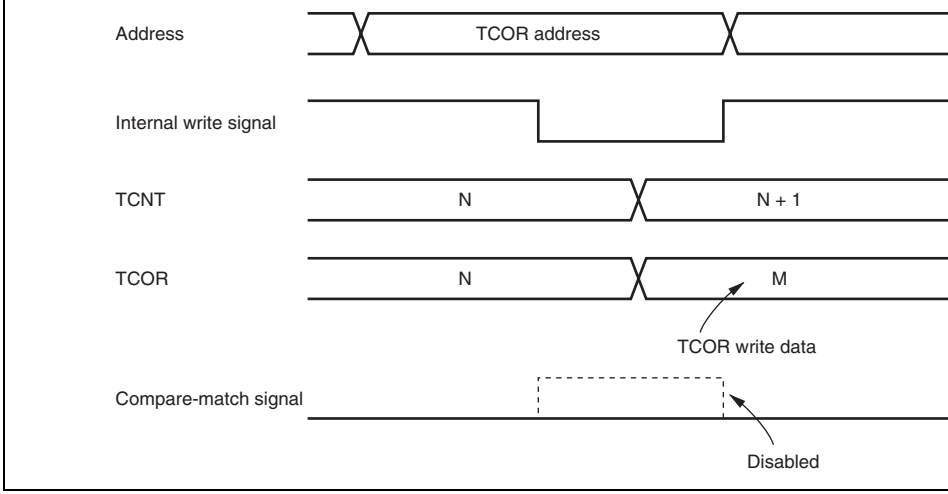
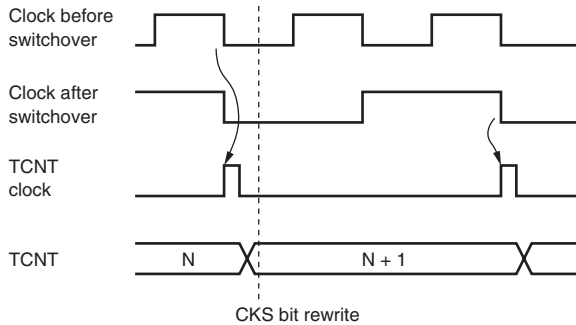
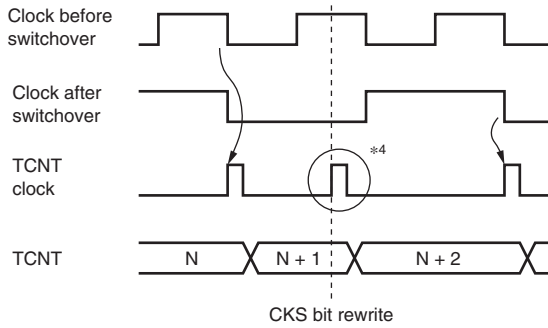
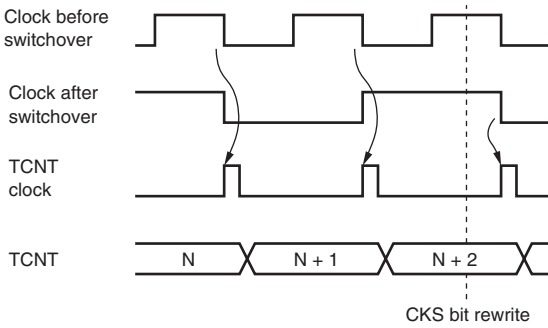


Figure 11.9 Conflict between TCOR Write and Compare-Match

Erroneous incrementation can also happen when switching between internal and external

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]	

		TCNT N N + 1 N + 2 CKS bit rewrite
3	Clock switching from high to low level*[3]	
4	Clock switching from high to high level	

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

12.1 Features

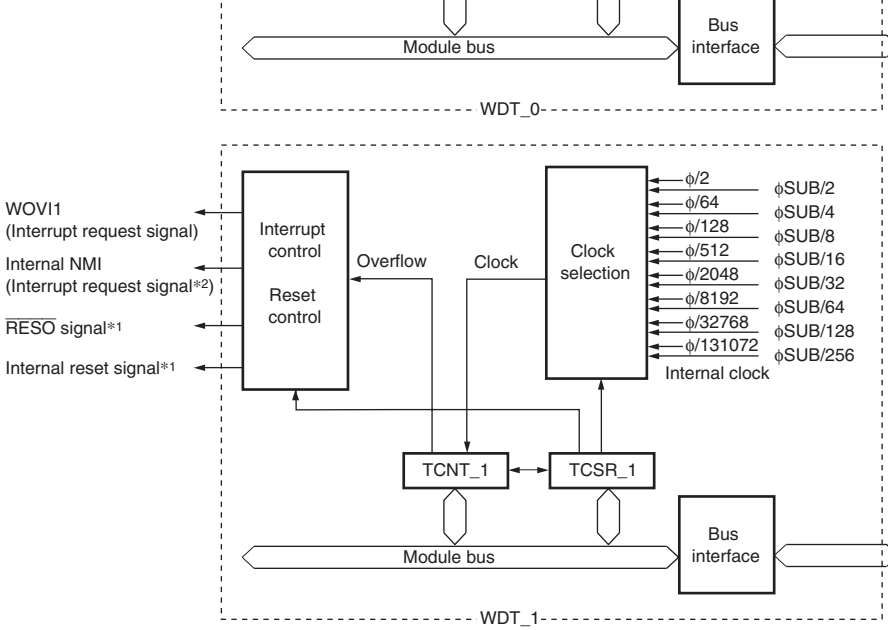
- Selectable from eight (WDT_0) or 16 (WDT_1) counter input clocks.
- Switchable between watchdog timer mode and interval timer mode

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/$\overline{\text{IT}}$	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/$\overline{\text{NMI}}$	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 (frequency: 83.89 ms)
	110: ϕ/32768 (frequency: 335.5 ms)
	111: ϕ/131072 (frequency: 1.34 s)

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/$\overline{\text{IT}}$	0	R/W	Timer Mode Select Selects whether the WDT is used as a watchdog interval timer. 0: Interval timer mode 1: Watchdog timer mode
5	TME	0	R/W	Timer Enable When this bit is set to 1, TCNT starts counting. When this bit is cleared, TCNT stops counting an initialized to H'00. When the PSS bit is 1, TCNT is not initialized. Wr to initialize TCNT.
4	PSS	0	R/W	Prescaler Select Selects the clock source to be input to TCNT. 0: Counts the divided cycle of ϕ–based prescaler 1: Counts the divided cycle of ϕSUB–based presc (PSS)
3	RST/$\overline{\text{NMI}}$	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 (frequency: 5.243 ms)

100: ϕ/2048 (frequency: 20.97 ms)

101: ϕ/8192 (frequency: 83.89 ms)

110: ϕ/32768 (frequency: 335.5 ms)

111: ϕ/131072 (frequency: 1.34 s)

When PSS = 1:

000: ϕSUB/2 (cycle: 15.6 ms)

001: ϕSUB/4 (cycle: 31.3 ms)

010: ϕSUB/8 (cycle: 62.5 ms)

011: ϕSUB/16 (cycle: 125 ms)

100: ϕSUB/32 (cycle: 250 ms)

101: ϕSUB/64 (cycle: 500 ms)

110: ϕSUB/128 (cycle: 1 s)

111: ϕSUB/256 (cycle: 2 s)

Notes: 1. Only 0 can be written to clear the flag.

2. When OVF is polled with the interval timer interrupt disabled, OVF = 1 must b
least twice.

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 cause
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.

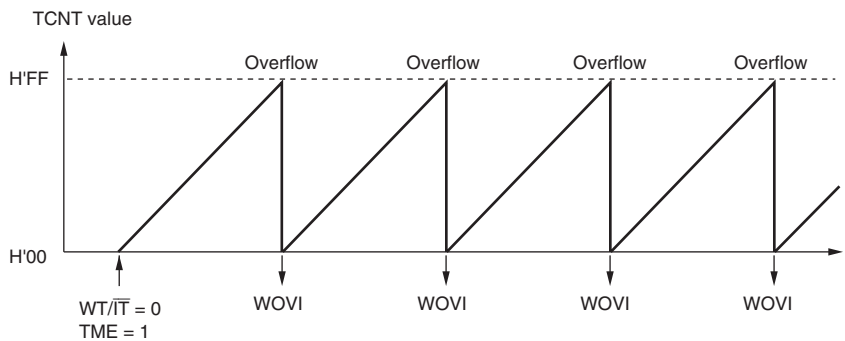
WT/$\overline{\text{IT}}$: Timer mode select bit
TME: Timer enable bit
OVF: Overflow flag

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/$\overline{\text{NMI}}$ = 1) Operation

12.4.2 Interval Timer Mode

When the WDT is used as an interval timer, an interval timer interrupt (WOVI) is gener
time the TCNT overflows, as shown in figure 12.3. Therefore, an interrupt can be genera
intervals. When the TCNT overflows in interval timer mode, an interval timer interrupt
requested at the same time the OVF bit of TCSR is set to 1. The timing is shown in figu



WOVI : Interval timer interrupt request occurrence

Figure 12.3 Interval Timer Mode Operation

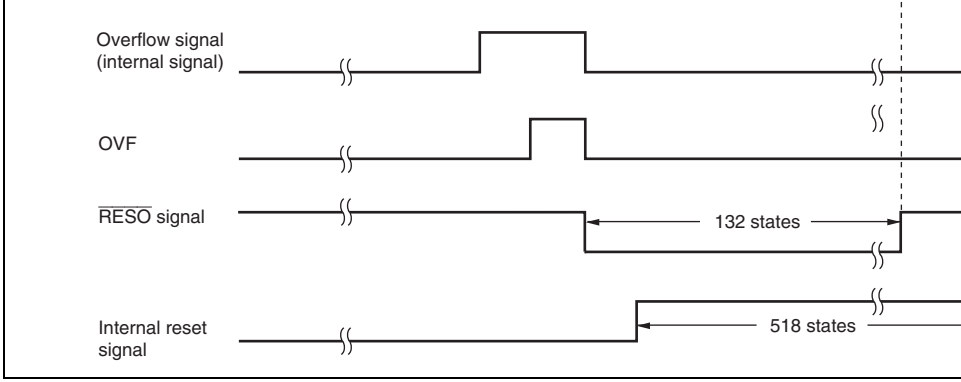


Figure 12.5 Output Timing of $\overline{\text{RESO}}$ Signal

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

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

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

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

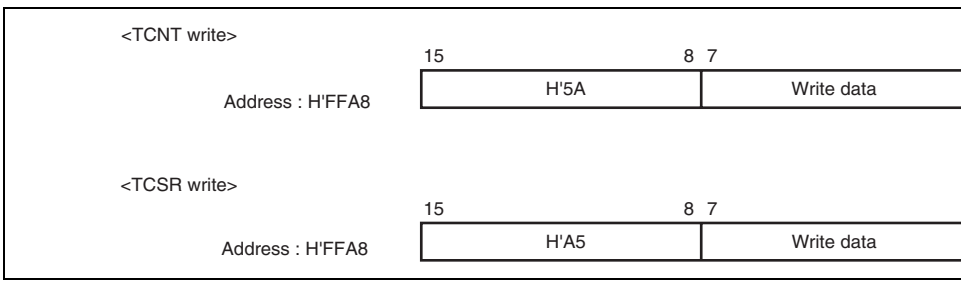


Figure 12.6 Writing to TCNT and TCSR (WDT_0)

Reading from TCNT and TCSR (Example of WDT_0):

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

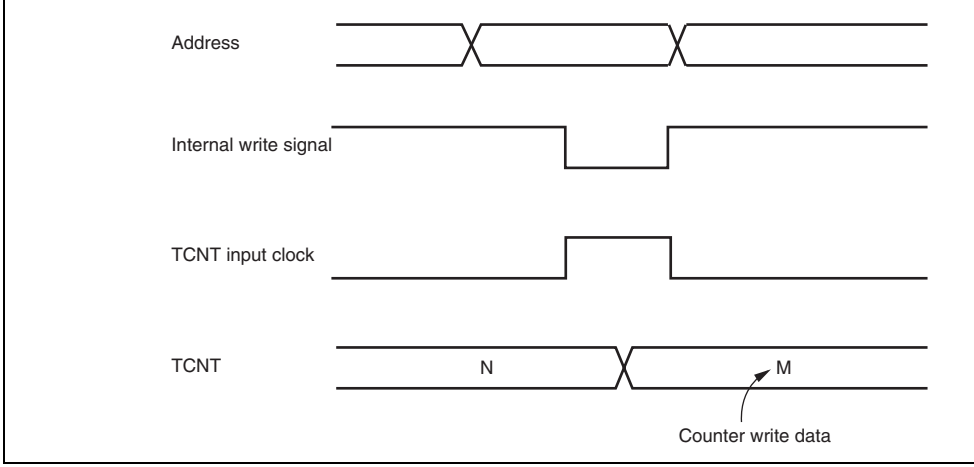


Figure 12.7 Conflict between TCNT Write and Increment

12.6.3 Changing Values of CKS2 to CKS0 Bits

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

12.6.4 Changing Value of PSS Bit

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

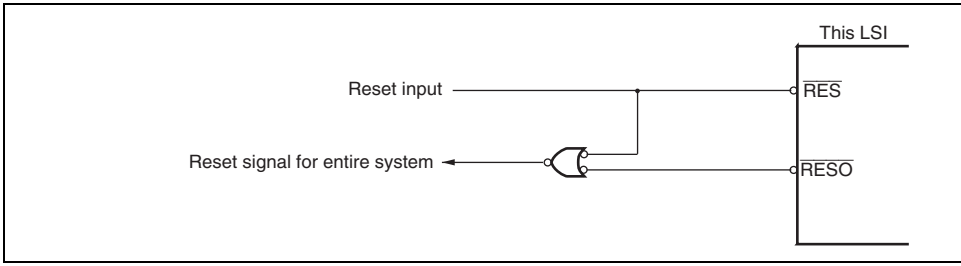


Figure 12.8 Sample Circuit for Resetting the System by the $\overline{\text{RESO}}$ Signa

13.1 Features

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

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

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

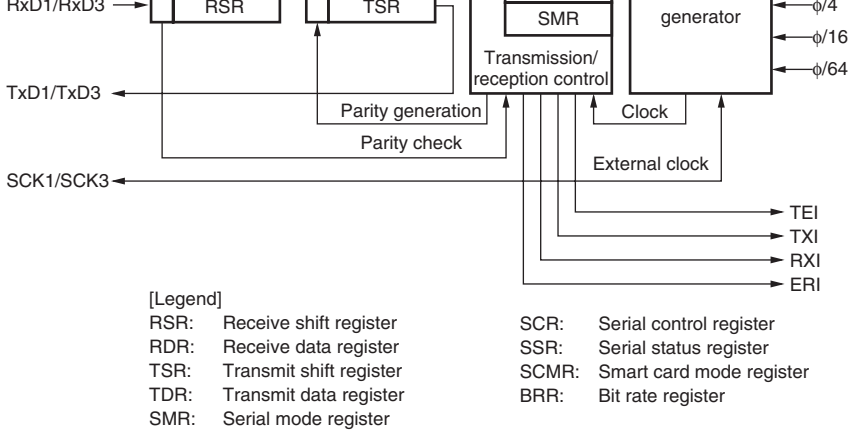
  Four interrupt sources — transmit-end, transmit-data-empty, receive-data-full, and re
  error — that can issue requests.

  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



[Legend]

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

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 canno
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, Cloc Control.
6	BLK	0	R/W	Setting this bit to 1 allows block transfer mode c 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 settin 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)

				Transmit Interrupt Enable 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 c On receiving data in which the multiprocessor bi bit is automatically cleared and normal reception resumed. For details, see section 13.5, Multipro Communication Function.
2	TEIE	0	R/W	Transmit End Interrupt Enable When this bit is set to 1, a TEI interrupt request enabled.

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

[Legend]

x: Don't care

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

[Legend]

x: Don't care.

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

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

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

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

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

[Legend]

B: Bit rate (bit/s)

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

ϕ: Operating frequency (MHz)

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

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

SMR Setting		
BCP1	**BCP0**	
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	–0.4
4800	0	129	0.16	0	162	0.15
9600	0	64	0.16	0	80	–0.4
19200	0	32	–1.36	0	40	–0.7
31250	0	19	0.00	0	24	0.00
38400	0	15	1.73	0	19	1.73

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

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

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

[Legend]

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

*: Continuous transfer or reception is not possible.

Table 13.7 Maximum Bit Rate with External Clock Input (Clock Synchronous M

ϕ (MHz)	External Input Clock (MHz)	Maximum Bit Rate (bit
20	3.3333	3333333.3
25	4.1667	4166666.7

20.00	26882	0	0
21.4272	28800	0	0
25.00	33602	0	0

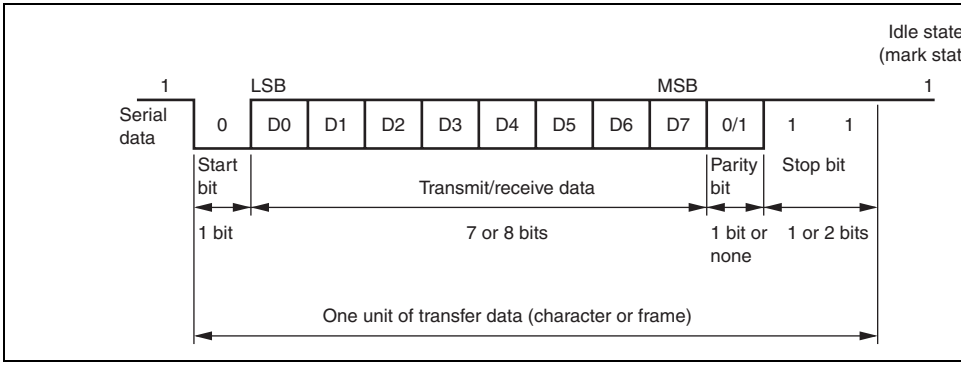


Figure 13.2 Data Format in Asynchronous Communication (Example with 8-Bit Data, Parity, Two Stop Bits)

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 determine
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.

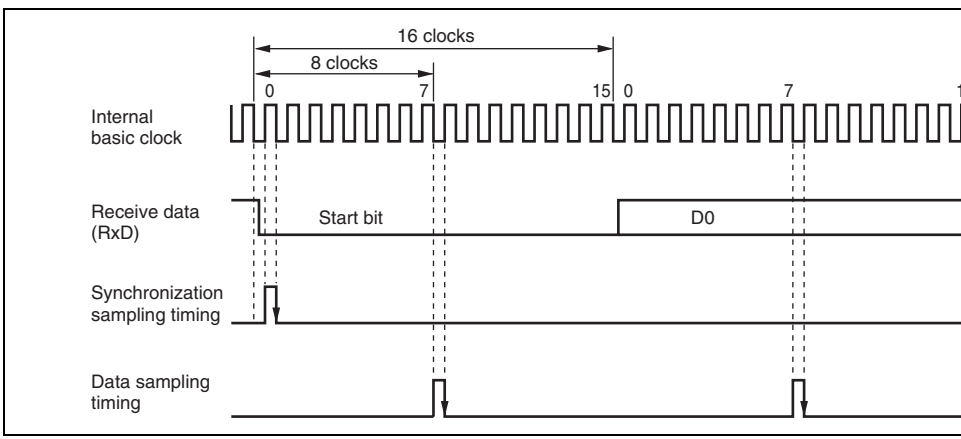


Figure 13.3 Receive Data Sampling Timing in Asynchronous Mode

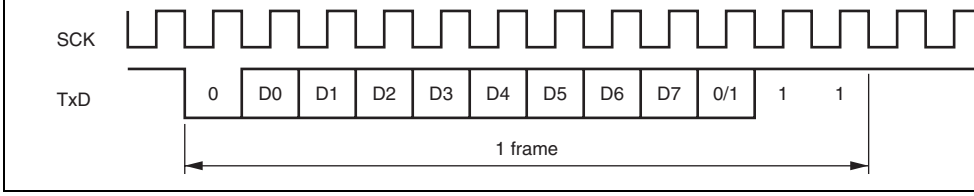
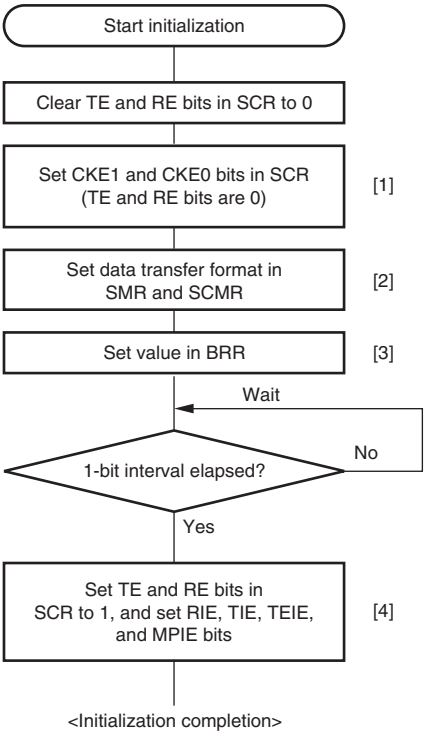


Figure 13.4 Relation between Output Clock and Transmit Data Phase (Asynchronous Mode)



[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

TDR before transmission of the current transmit data has finished, continuous transmi
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.

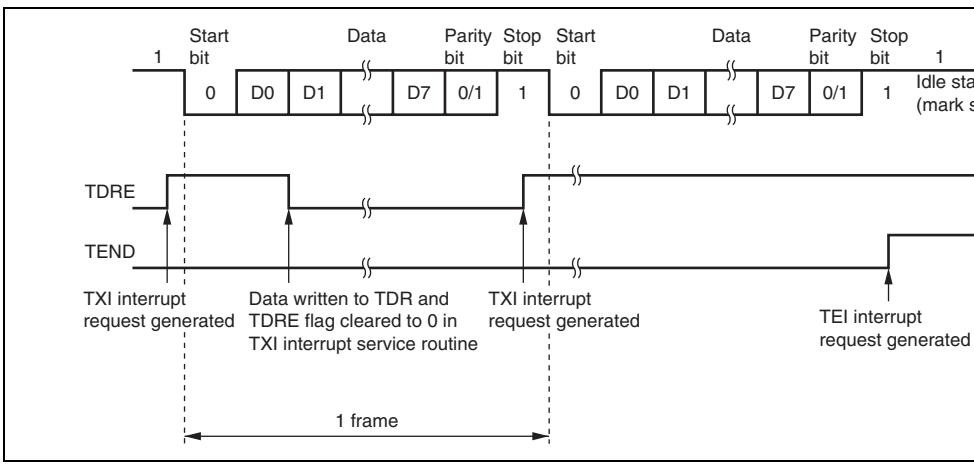
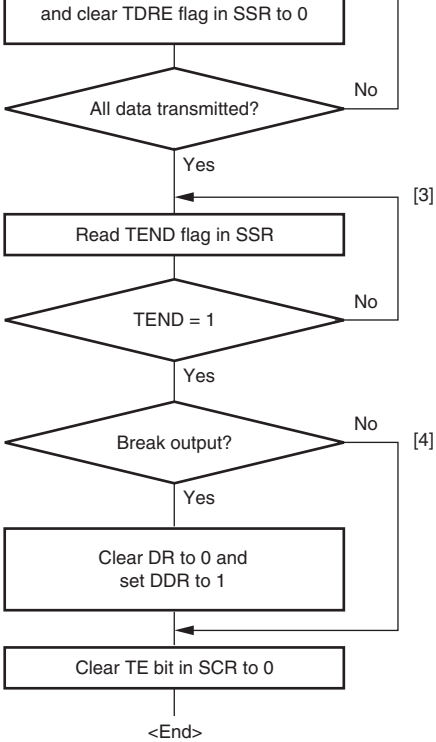


Figure 13.6 Example of Operation in Transmission in Asynchronous Mode (Example with 8-Bit Data, Parity, One Stop Bit)



[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

flag remains to be set to 1.

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.

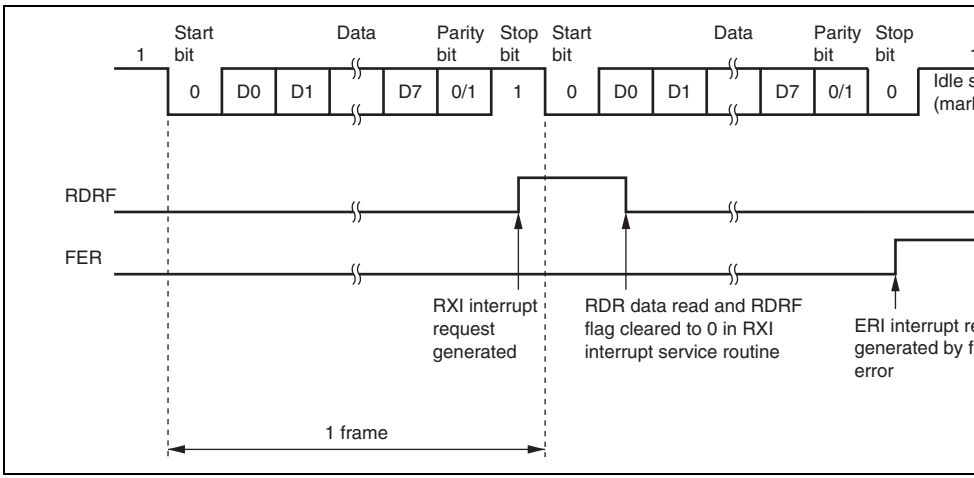
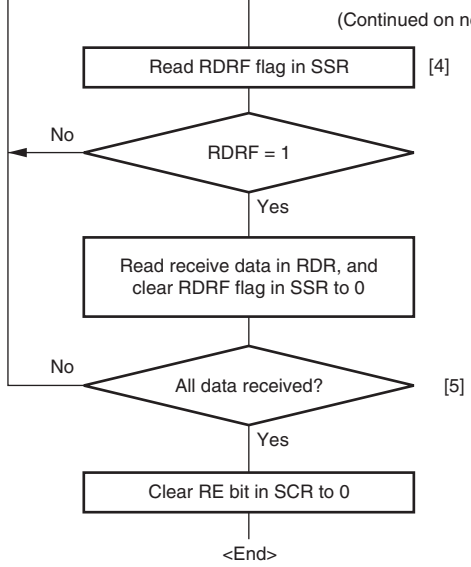


Figure 13.8 Example of SCI Operation in Reception (Example with 8-Bit Data, Parity, One Stop Bit)

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 correspondin the RxD pin.

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

[5] Serial reception continuation proce To continue serial reception, before stop bit for the current frame is received, read the RDRF flag, read RDR, and clear the RDRF flag to 0 However, the RDRF flag is cleared automatically when the DTC is initia by an RXI interrupt and reads data RDR.

[Legend]
∨ : Logical add (OR)

Figure 13.9 Sample Serial Reception Flowchart (1)

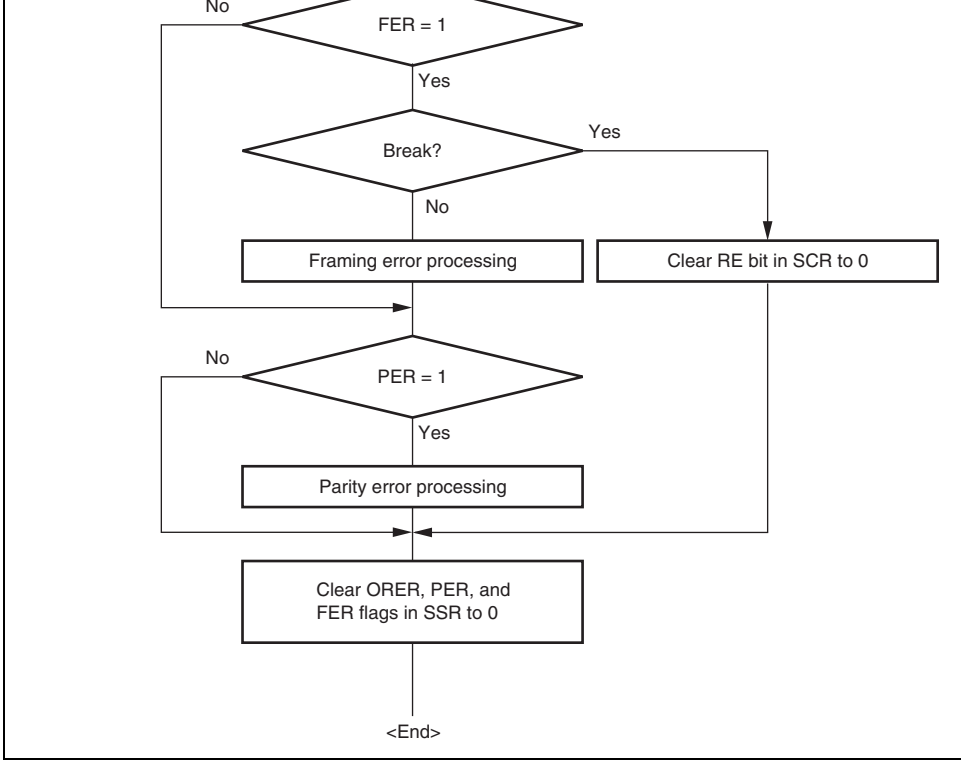


Figure 13.9 Sample Serial Reception Flowchart (2)

transmission cycle, and if the multiprocessor bit is 0, the cycle is a data transmission cycl
13.10 shows an example of inter-processor communication using the multiprocessor form
transmitting station first sends the ID code of the receiving station with which it wants to
serial communication as data with a 1 multiprocessor bit added. It then sends transmit dat
with a 0 multiprocessor bit added. When data with a 1 multiprocessor bit is received, the
station compares that data with its own ID. The station whose ID matches then receives th
sent next. Stations whose ID does not match continue to skip data until data with a 1
multiprocessor bit is again received.

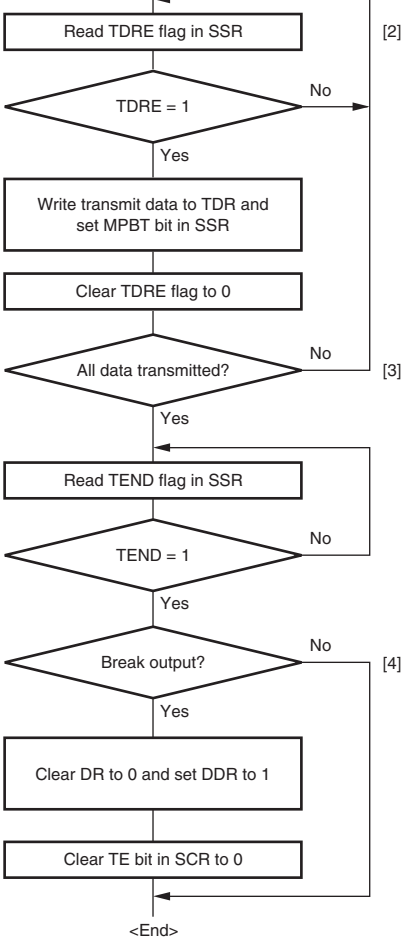
The SCI uses the MPIE bit in SCR to implement this function. When the MPIE bit is set t
transfer of receive data from RSR to RDR, error flag detection, and setting the RDRF, FE
ORER status flags in SSR to 1 are prohibited until data with a 1 multiprocessor bit is rece
reception of a receive character with a 1 multiprocessor bit, the MPB bit in SSR is set to
MPIE bit is automatically cleared, thus normal reception is resumed. If the RIE bit in SCI
1 at this time, an RXI interrupt is generated.

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

	ID transmission cycle = receiving station specification	Data transmission cycle = Data transmission to receiving station specified by ID

[Legend]
MPB: Multiprocessor bit

Figure 13.10 Example of Communication Using Multiprocessor Format (Transmission of Data H'AA to Receiving Station A)



After the TE bit is set to 1, a frame of 1s is output, and transmission is enabled.

[2] SCI status check and transmit data write:
Read SSR and check that the TDRE flag is set to 1, then write transmit data to TDR. Set the MPBT bit in SSR to 0 or 1. Finally, clear the TDRE flag to 0.

[3] Serial transmission continuation procedure:
To continue serial transmission, be sure to read 1 from the TDRE flag to confirm that writing is possible, then write data to TDR, and then clear the TDRE flag to 0. However, the TDRE flag is checked and cleared automatically when the DTC is initiated by a transmit data empty interrupt (TXI) request and writes data to TDR.

[4] Break output at the end of serial transmission:
To output a break in serial transmission, set port DDR to 1, clear DR to 0, and then clear the TE bit in SCR to 0.

Figure 13.11 Sample Multiprocessor Serial Transmission Flowchart

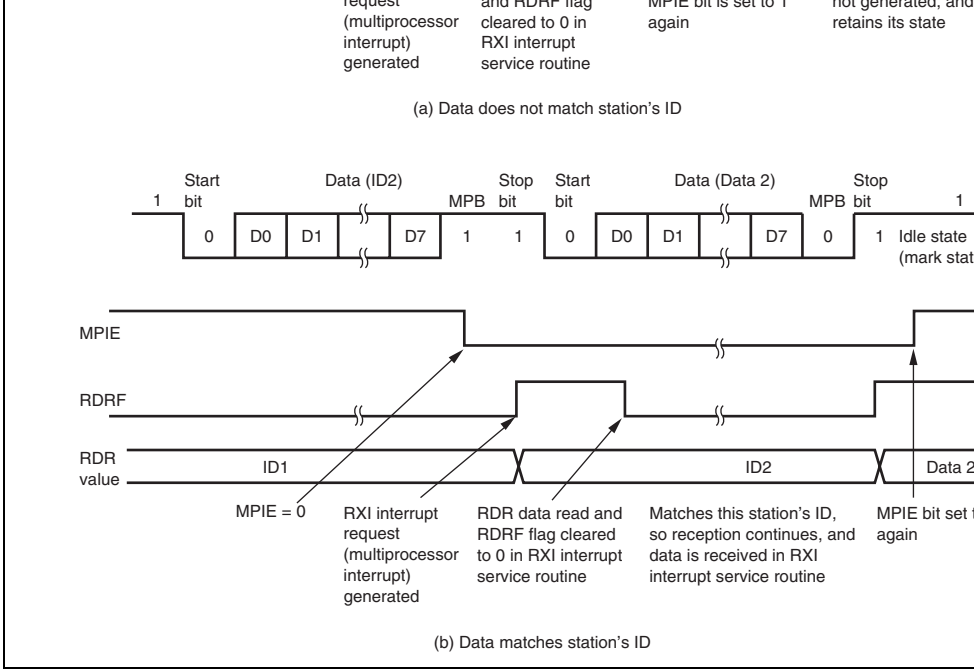
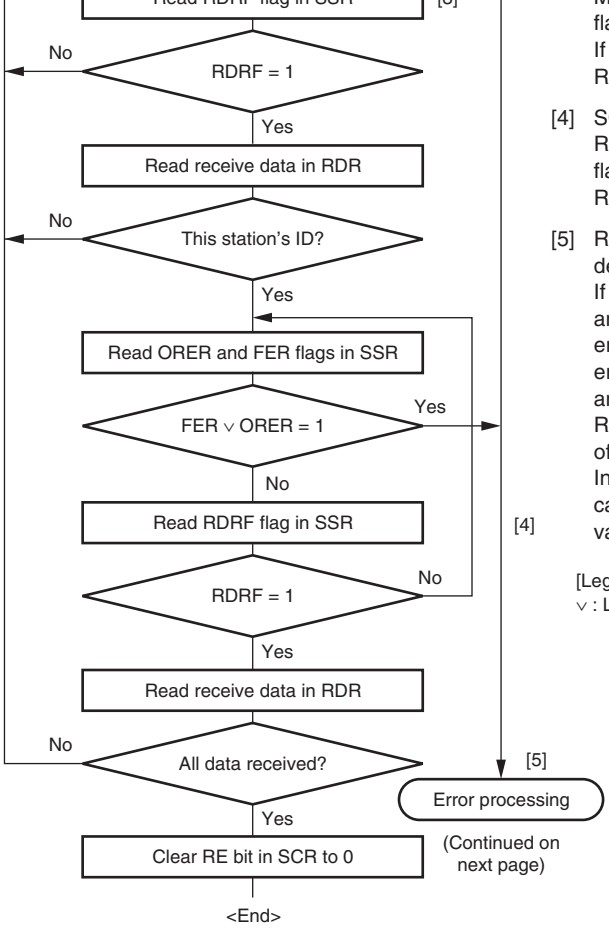


Figure 13.12 Example of SCI Operation in Reception (Example with 8-Bit Data, Multiprocessor Bit, One Stop Bit)



MPIE bit to 1 again, and clear the
flag to 0.
If the data is this station's ID, clea
RDRF flag to 0.

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

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

[Legend]
∨ : Logical add (OR)

Figure 13.13 Sample Multiprocessor Serial Reception Flowchart (1)

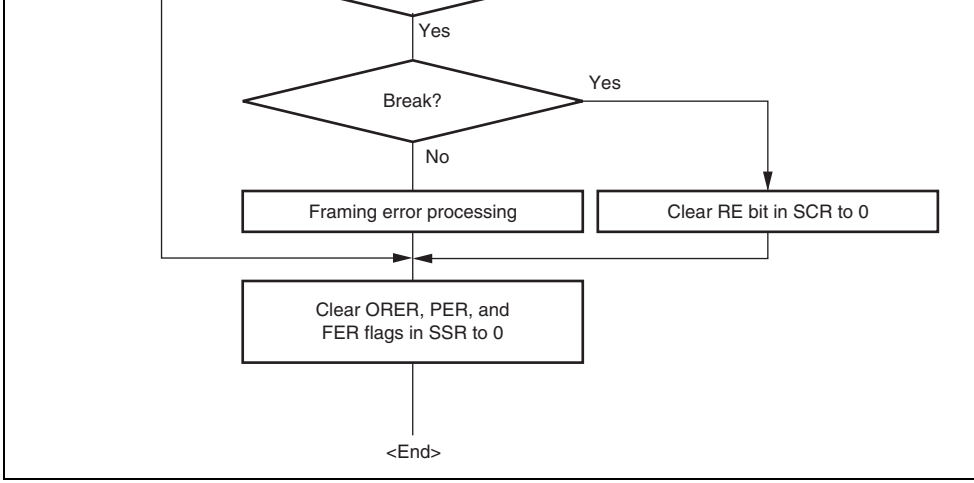


Figure 13.13 Sample Multiprocessor Serial Reception Flowchart (2)

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.

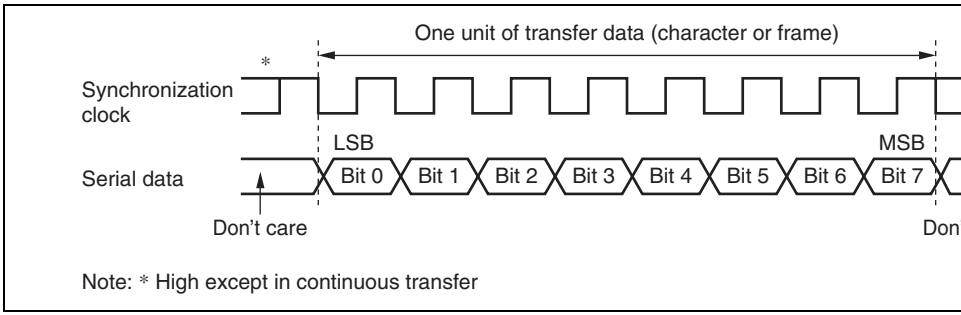
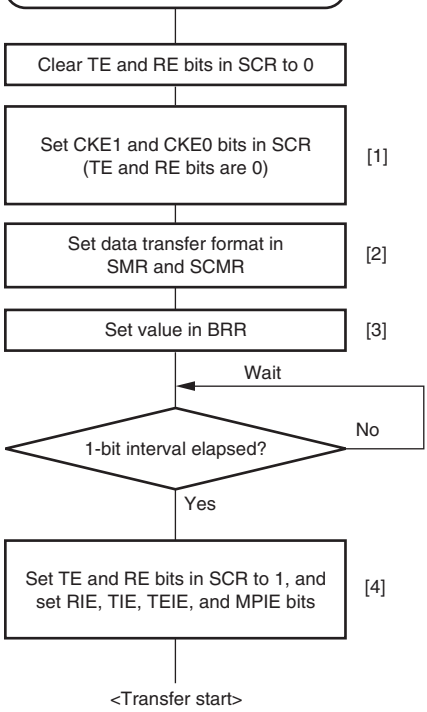


Figure 13.14 Data Format in Synchronous Communication (LSB-First)

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 c
mode has been specified and synchronized with the input clock when use of an exter
has been specified.
4. The SCI checks the TDRE flag at the timing for sending the last bit.
5. If the TDRE flag is cleared to 0, data is transferred from TDR to TSR, and serial tran
of the next frame is started.
6. If the TDRE flag is set to 1, the TEND flag in SSR is set to 1, and the TxD pin main
output state of the last bit. If the TEIE bit in SCR is set to 1 at this time, a TEI interru
is generated. The SCK pin is fixed high.

Figure 13.17 shows a sample flowchart for serial data transmission. Even if the TDRE fl
cleared to 0, transmission will not start while a receive error flag (ORER, FER, or PER)
Make sure to clear the receive error flags to 0 before starting transmission. Note that cle
RE bit to 0 does not clear the receive error flags.

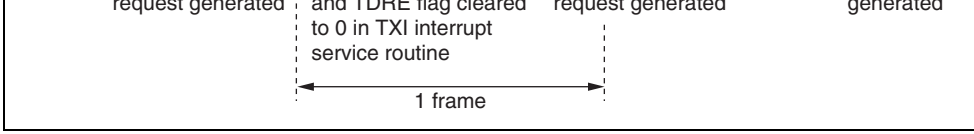
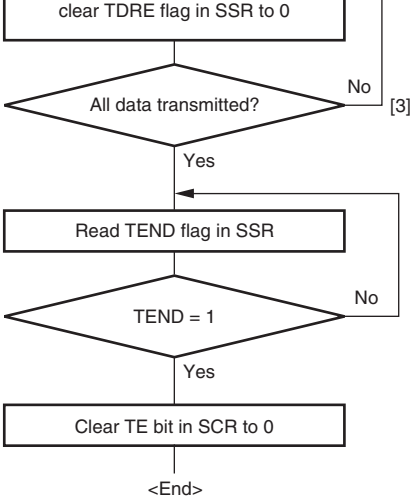


Figure 13.16 Sample SCI Transmission Operation in Clock Synchronous Mo



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.

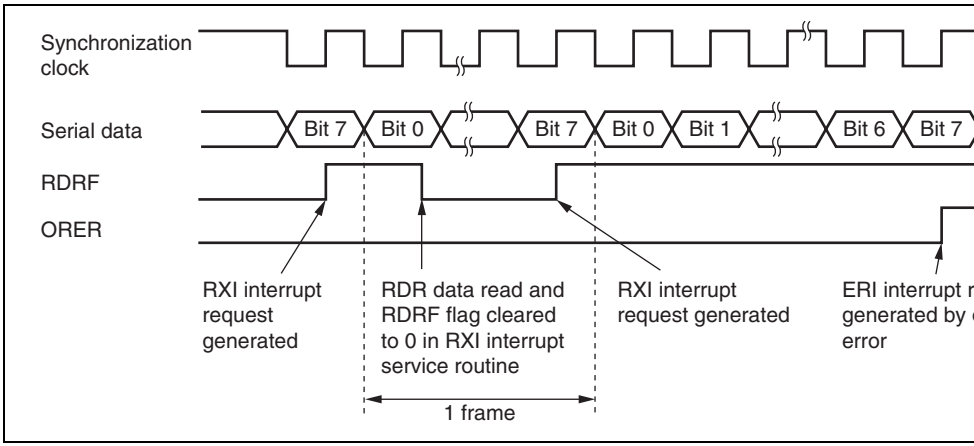
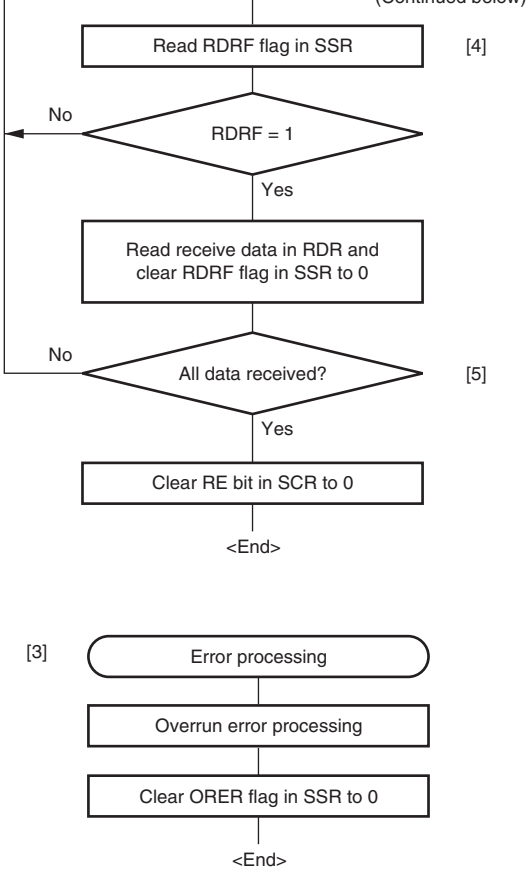


Figure 13.18 Example of SCI Receive Operation in Clock Synchronous Moc

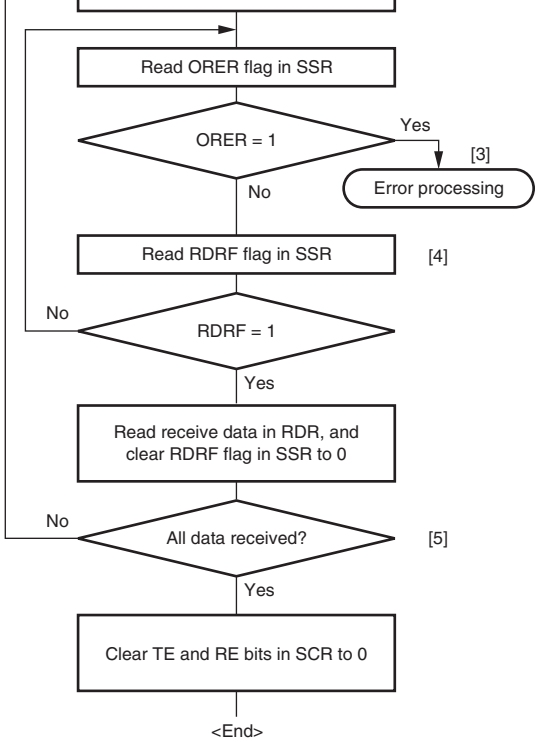
Reception cannot be resumed while a receive error flag is set to 1. Accordingly, clear the FER, PER, and RDRF bits to 0 before resuming reception. Figure 13.19 shows a sample for serial data reception.



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, bef
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
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.

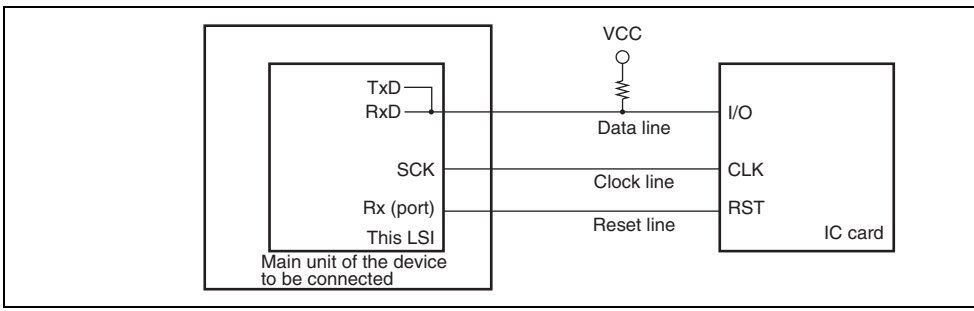


Figure 13.21 Pin Connection for Smart Card Interface

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 afte
  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.

Output from
the receiving st

[Legend]

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

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.

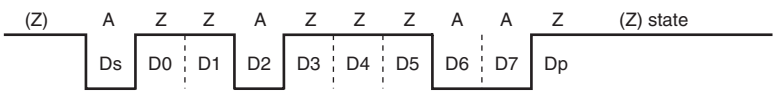


Figure 13.23 Direct Convention (SDIR = SINV = O/$\overline{E}$ = 0)

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.

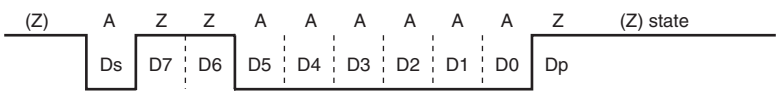


Figure 13.24 Inverse Convention (SDIR = SINV = O/$\overline{E}$ = 1)

- 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

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

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.

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

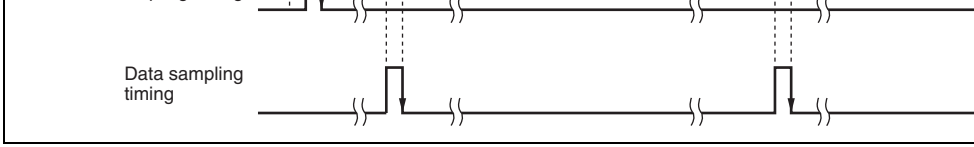


**Figure 13.25 Receive Data Sampling Timing in Smart Card Interface Mode (Whe
Frequency is 372 Times the Bit Rate)**

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.

To switch from reception to transmission, first verify that reception has completed, and in
the SCI. At the end of initialization, RE and TE should be set to 0 and 1, respectively. Re

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. For DTC settings, see section 7, Data Transfer Controller (DTC).

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.

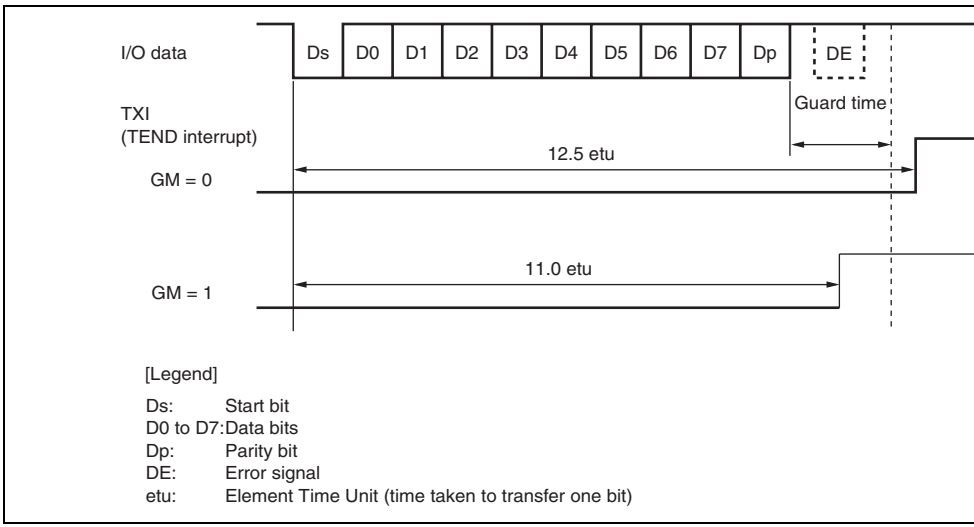


Figure 13.27 TEND Flag Set Timings during Transmission

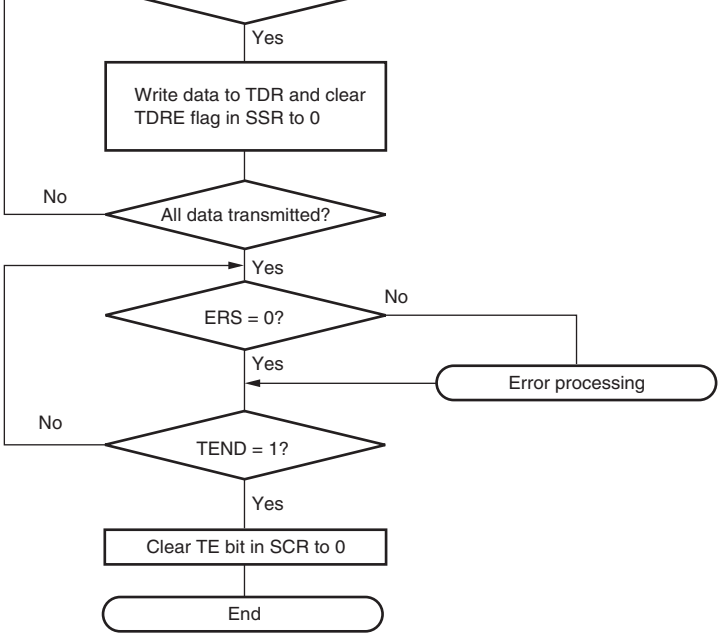


Figure 13.28 Sample Transmission Flowchart

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 acti 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

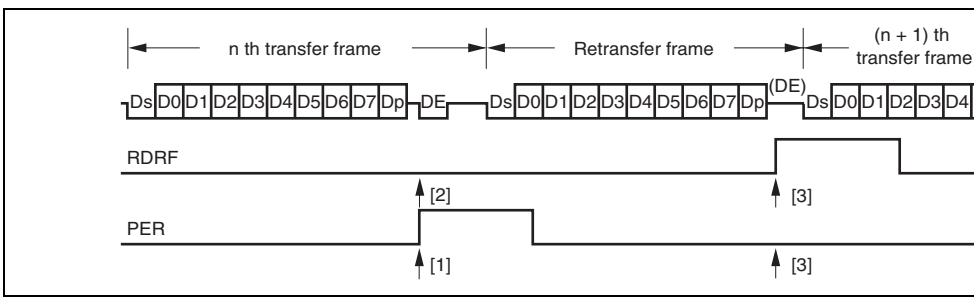


Figure 13.29 Data Re-transfer Operation in SCI Reception Mode

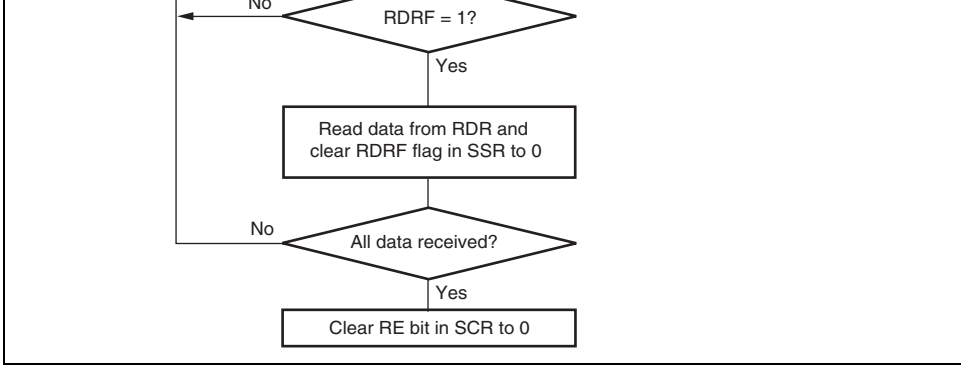


Figure 13.30 Sample Reception Flowchart

13.7.8 Clock Output Control

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

Figure 13.31 shows an example of clock output fixing timing when the CKE0 bit is cont
with GM = 1 and CKE1 = 0.

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 Smart Card Interface Mode to Software Standby Mode:

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

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

1. Cancel software standby mode.
2. Write 1 to the CKE0 bit in SCR to start clock output. A clock signal with the appropr
   ratio is then generated.

13.8 Interrupt Sources

13.8.1 Interrupts in Normal Serial Communication Interface Mode

Table 13.12 shows the interrupt sources in normal serial communication interface mode different interrupt vector is assigned to each interrupt source, and individual interrupt so be enabled or disabled using the enable bits in SCR.

When the TDRE flag in SSR is set to 1, a TXI interrupt request is generated. When the T in SSR is set to 1, a TEI interrupt request is generated. A TXI interrupt can activate the I allow data transfer. The TDRE flag is automatically cleared to 0 at data transfer by the I

When the RDRF flag in SSR is set to 1, an RXI interrupt request is generated. When the PER, or FER flag in SSR is set to 1, an ERI interrupt request is generated. An RXI inter activate the DTC to allow data transfer. The RDRF flag is automatically cleared to 0 at c transfer by the DTC.

A TEI interrupt is requested when the TEND flag is set to 1 while the TEIE bit is set to interrupt and a TXI interrupt are requested simultaneously, the TXI interrupt has priority acceptance. However, note that if the TDRE and TEND flags are cleared simultaneously TXI interrupt routine, the SCI cannot branch to the TEI interrupt routine later.

TEI3	Transmit end	TEND	Not possible

13.8.2 Interrupts in Smart Card Interface Mode

Table 13.13 shows the interrupt sources in smart card interface mode. A TEI interrupt req
cannot be used in this mode.

Table 13.13 SCI Interrupt Sources

Channel	Name	Interrupt Source	Interrupt Flag	DTC Activation
1	ERI1	Receive error, error signal detection	ORER, PER, ERS	Not possible
	RXI1	Receive data full	RDRF	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 simul
set to 1, thus generating a TXI interrupt request. This activates the DTC by a TXI interrup
thus allowing transfer of transmit data if the TXI interrupt request is specified as a source
activation beforehand. The TDRE and TEND flags are automatically cleared to 0 at data
by the DTC. If an error occurs, the SCI automatically re-transmits the same data. During
transmission, the TEND flag remains as 0, thus not activating the DTC. Therefore, the SC
DTC automatically transmit the specified number of bytes, including re-transmission in t

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.

13.9 Usage Notes

13.9.1 Module Stop Mode Setting

SCI operation can be disabled or enabled using the module stop control register. The ini
is for SCI operation to be halted. Register access is enabled by clearing module stop mo
details, see section 22, Power-Down Modes.

13.9.2 Break Detection and Processing

When framing error detection is performed, a break can be detected by reading the RxD
directly. In a break, the input from the RxD pin becomes all 0s, and so the FER flag in S
and the PER flag may also be set. Note that, since the SCI continues the receive operatio
after receiving a break, even if the FER flag is cleared to 0, it will be set to 1 again.

13.9.3 Mark State and Break Sending

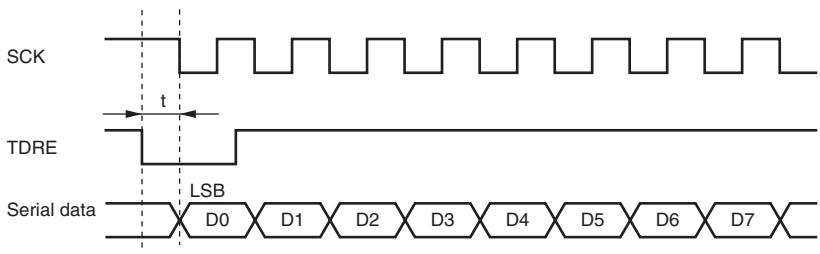
When the TE bit in SCR is 0, the TxD pin is used as an I/O port whose direction (input o
and level are determined by DR and DDR of the port. This can be used to set the TxD pi
state (high level) or send a break during serial data transmission. To maintain the comm
line at mark state until TE is set to 1, set both DDR and DR to 1. Since the TE bit is clea
this point, the TxD pin becomes an I/O port, and 1 is output from the TxD pin. To send a
during serial transmission, first set DDR to 1 and DR to 0, and then clear the TE bit to 0
TE bit is cleared to 0, the transmitter is initialized regardless of the current transmission
TxD pin becomes an I/O port, and 0 is output from the TxD pin.

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.

13.9.6 Restrictions on Using DTC

When the external clock source is used as a synchronization clock, update TDR by the D wait for at least five ϕ clock cycles before allowing the transmit clock to be input. If the t clock is input within four clock cycles after TDR modification, the SCI may malfunction 13.33).

When using the DTC to read RDR, be sure to set the receive end interrupt source (RXI) a activation source.

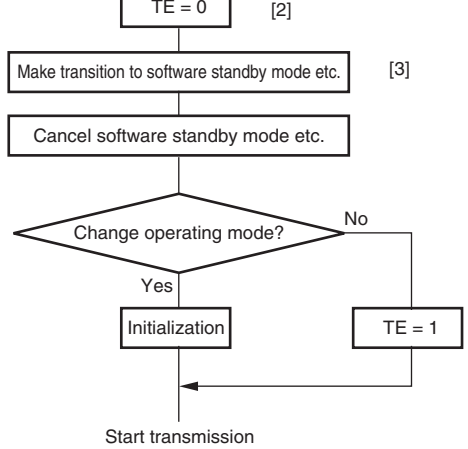


Note: When external clock is supplied, t must be more than four clock cycles.

Figure 13.33 Sample Transmission using DTC in Clock Synchronous Mod

Figure 13.34 shows a sample flowchart for mode transition during transmission. Figures
13.36 show the pin states during transmission.

Before making the transition from the transmission mode using DTC transfer to module
software standby, stop all transmit operations (TE = TIE = TEIE = 0). Setting TE and T
after mode cancellation generates a TXI interrupt request to start transmission using the



[2] Also clear TIE and TEIE to 0 when they are 1.

[3] Module stop mode is included.

Figure 13.34 Sample Flowchart for Mode Transition during Transmission

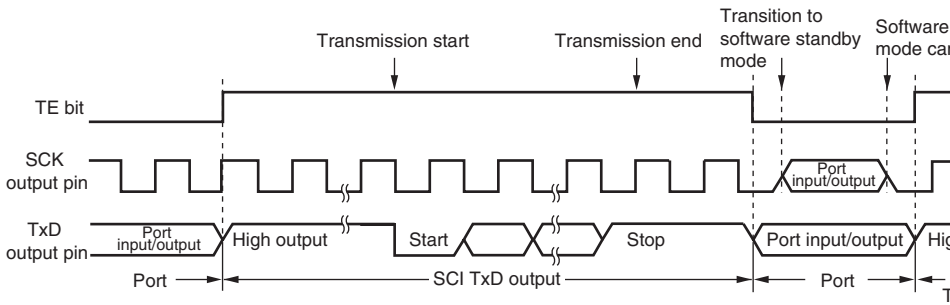


Figure 13.35 Pin States during Transmission in Asynchronous Mode (Internal C

Figure 13.36 Pin States during Transmission in Clock Synchronous Mod (Internal Clock)

Reception: Before making the transition to module stop or software standby, stop recep 0). RSR, RDR, and SSR are reset. If transition is made during data reception, the data be received will be invalid.

To receive data in the same reception mode after mode cancellation, set RE to 1, and the reception. To receive data in a different reception mode, initialize the SCI first.

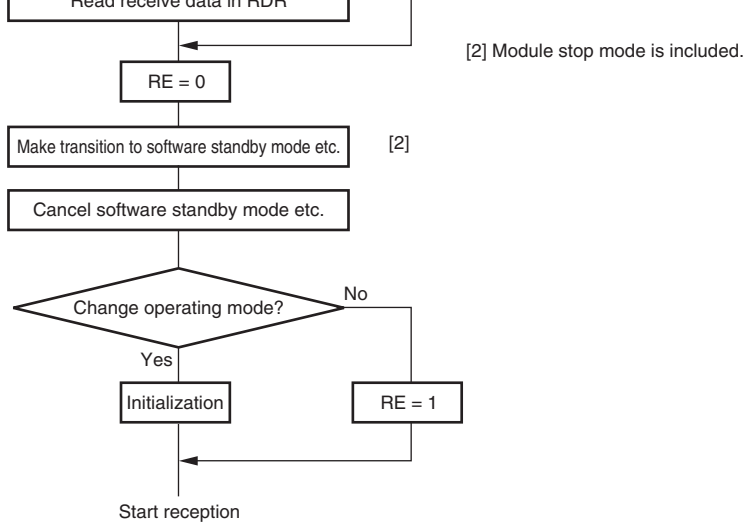


Figure 13.37 Sample Flowchart for Mode Transition during Reception

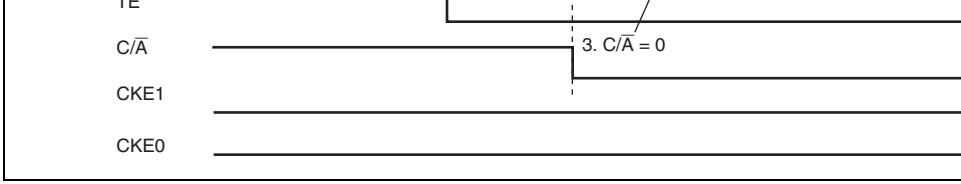


Figure 13.38 Switching from SCK Pins to Port Pins

To prevent the low pulse output that is generated when switching the SCK pins to the po
specify the SCK pins for input (pull up the SCK/port pins externally), and follow the pro
below with DDR = 1, DR = 1, C/$\overline{A}$ = 1, CKE1 = 0, CKE1 = 0, and TE = 1.

1. End serial data transmission
2. TE bit = 0
3. CKE1 bit = 1
4. C/$\overline{A}$ bit = 0 (switch to port output)
5. CKE1 bit = 0

Figure 13.39 Prevention of Low Pulse Output at Switching from SCK Pins to Po

- CRC code generated for any desired data length in an 8-bit unit
- CRC operation executed on eight bits in parallel
- One of three generating polynomials selectable
- CRC code generation for LSB-first or MSB-first communication selectable

Figure 14.1 shows a block diagram of the CRC operation circuit.

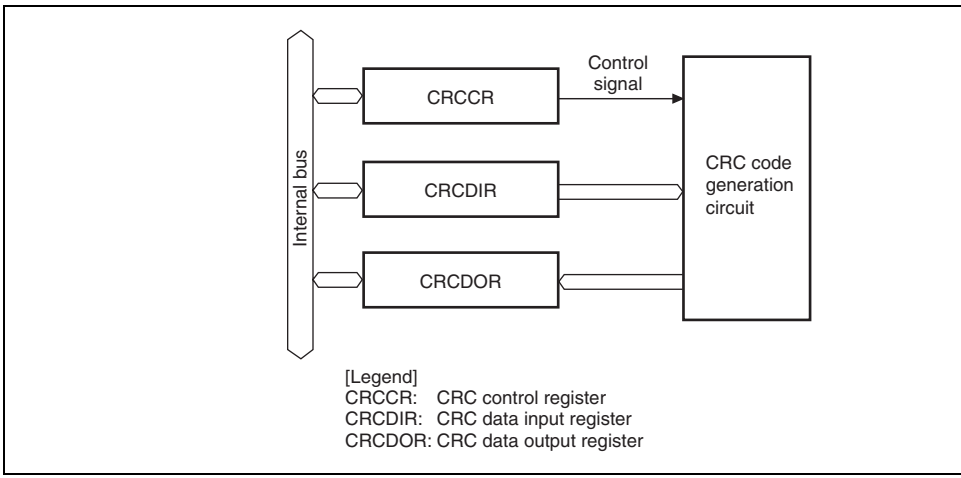


Figure 14.1 Block Diagram of CRC Operation Circuit

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 commun 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 (G1 and G0 bits) are set to 0 and 1, respectively, the lower byte of this register
the 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.

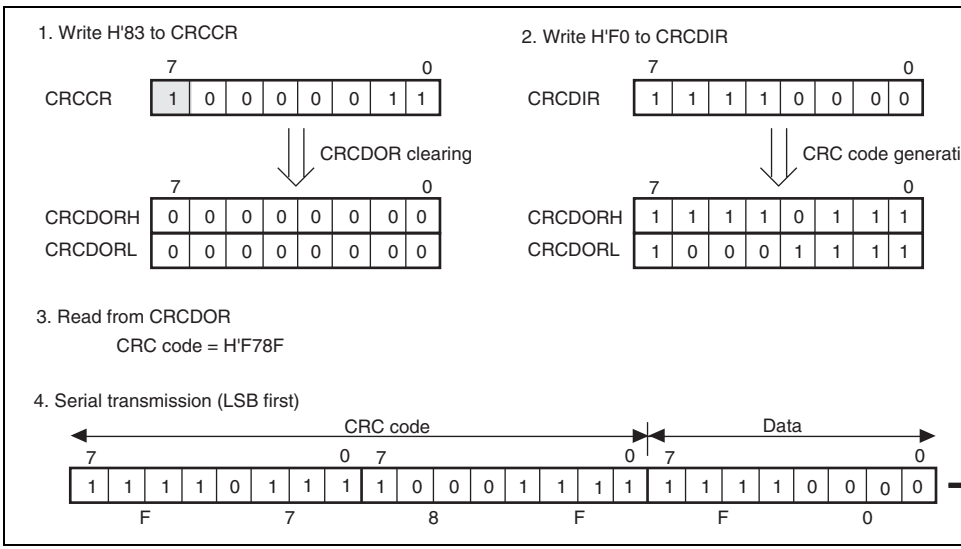


Figure 14.2 LSB-First Data Transmission

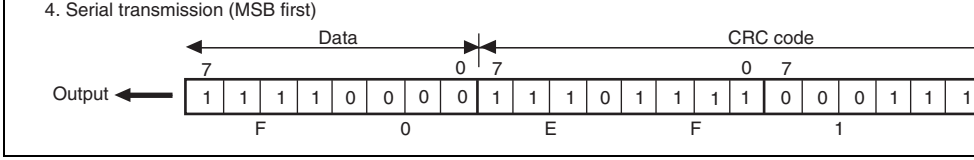


Figure 14.3 MSB-First Data Transmission

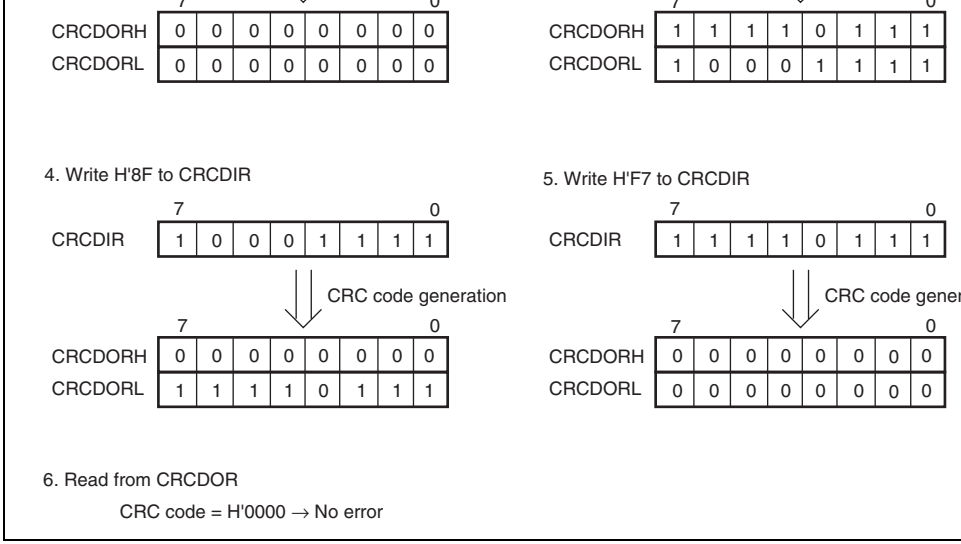


Figure 14.4 LSB-First Data Reception

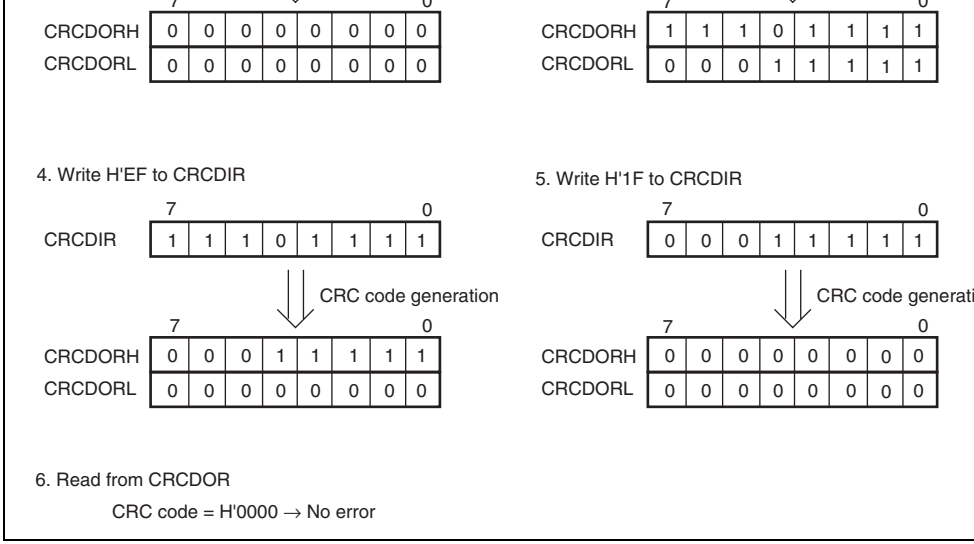


Figure 14.5 MSB-First Data Reception

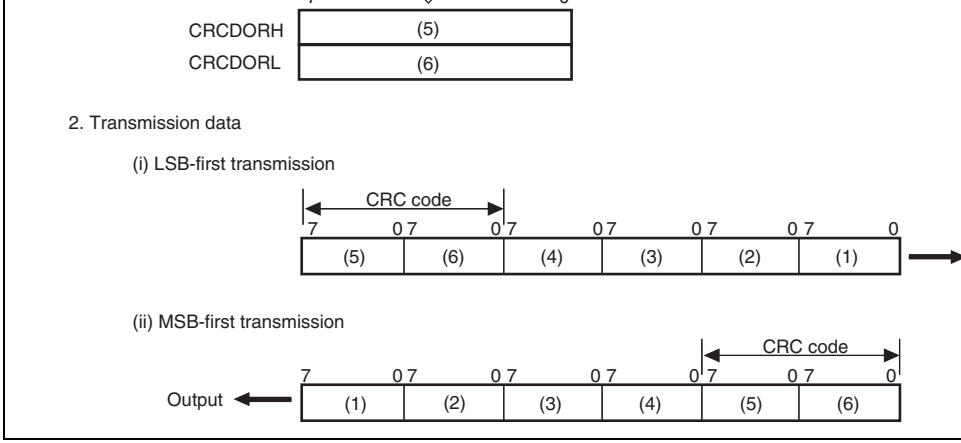
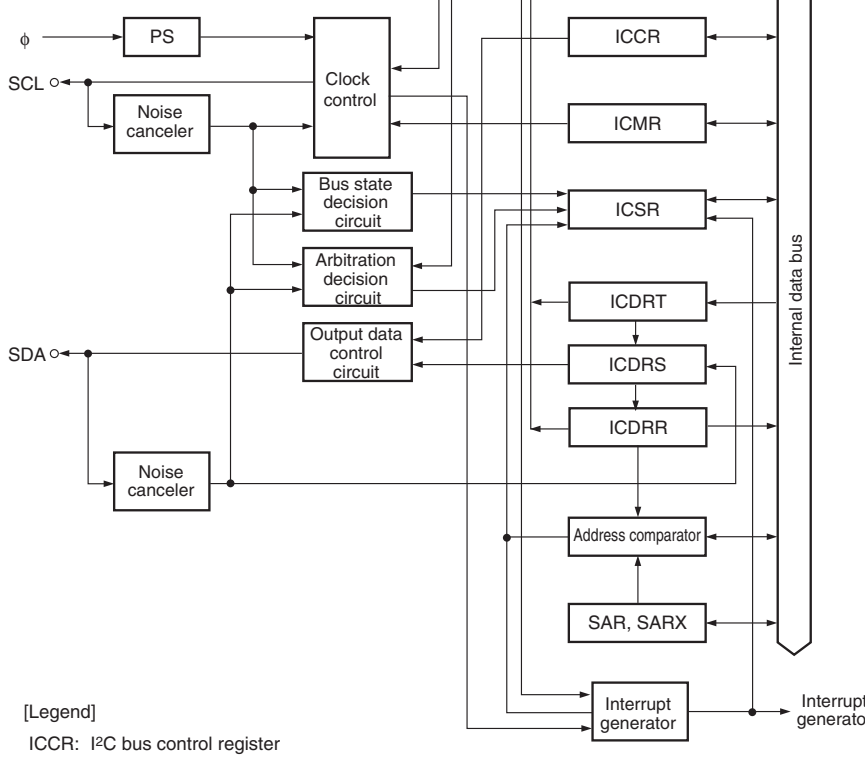


Figure 14.6 LSB-First and MSB-First Transmit Data

— 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



[Legend]

ICCR: I²C bus control register
ICMR: I²C bus mode register
ICSR: I²C bus status register
ICDR: I²C bus data register
ICXR: I²C bus extended control register
SAR: Slave address register
SARX: Slave address register X
PS: Prescaler

Figure 15.1 Block Diagram of I²C Bus Interface

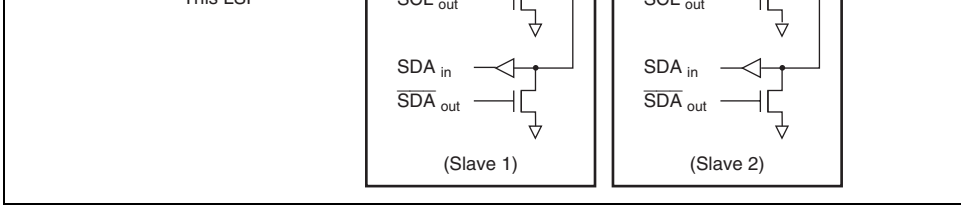


Figure 15.2 I²C Bus Interface Connections (Example: This LSI as Master

	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

Note: * In the text, the channel subscript is omitted, and only SCL and SDA are used.

- I²C bus mode register (ICMR)
- I²C bus transfer rate select register (IICX3)
- I²C bus control register (ICCR)
- I²C bus status register (ICSR)
- I²C bus extended control register (ICXR)
- I²C SMbus control register (ICSMBCR)

15.3.1 I²C Bus Data Register (ICDR)

ICDR is an 8-bit readable/writable register that is used as a transmit data register when transmitting and a receive data register when receiving. ICDR is divided internally into a register (ICDRS), receive buffer (ICDRR), and transmit buffer (ICDRT). Data transfers three registers are performed automatically in accordance with changes in the bus state, affect the status of internal flags such as ICDRE and ICDRF.

In master transmit mode with the I²C bus format, writing transmit data to ICDR should b performed after start condition detection. When the start condition is detected, previous is ignored. In slave transmit mode, writing should be performed after the slave addresses and the TRS bit is automatically changed to 1.

If IIC is in transmit mode (TRS=1) and the next data is in ICDRT (the ICDRE flag is 0), transferred automatically from ICDRT to ICDRS, following transmission of one frame 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 I

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.

15.3.2 Slave Address Register (SAR)

SAR sets the slave address and selects the communication format. When the LSI is in sla
with the I^2C bus format selected, if the FS bit is set to 0 and the upper 7 bits of SAR matc
upper 7 bits of the first frame received after a start condition, the LSI operates as the slav
specified by the master device. SAR can be accessed only when the ICE bit in ICCR is cl
0.

Bit	Bit Name	Initial Value	R/W	Description
7	SVA6	All 0	R/W	Slave 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 15.2. 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 15.2.

		• General call address recognized
1	0	I^2C bus format • SAR slave address ignored • SARX slave address recognized • General call address ignored
	1	Clocked synchronous serial format • SAR and SARX slave addresses ignored • General call address ignored

- I^2C bus format: addressing format with acknowledge bit
- Clocked synchronous serial format: non-addressing format without acknowledge bit, master mode only

				Set this bit to 0 when the I²C bus format is used.
6	WAIT	0	R/W	Wait Insertion Bit This bit is valid only in master mode with the I²C format. 0: Data and the acknowledge bit are transferred consecutively with no wait inserted. 1: After the fall of the clock for the final data bit (clock), the IRIC flag is set to 1 in ICCR, and a begins (with SCL at the low level). When the I is cleared to 0 in ICCR, the wait ends and the acknowledge bit is transferred. For details, refer to section 15.4.7, IRIC Setting and SCL Control.
5	CKS2	All 0	R/W	Transfer Clock Select
4	CKS1			These bits are used only in master mode.
3	CKS0			These bits select the required transfer rate, toge the IICX3 (channel 3) bit in IICX3, the IICX2 (cha IICX1 (channel 1), and IICX0 (channel 0) bits in Refer to table 15.3.

I²C Bus Format	Clocked Synchronous Serial
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: $\phi/2$ 1: $\phi/4$
2, 1	—	—	—	Reserved These bits cannot be modified. The read values undefined.
0	IICX3			IIC Transfer Rate Select 3 These bits are used to control IIC_3 operation. These bits select the transfer rate in master moc together with the CKS2 to CKS0 bits in ICMR. F transfer rate, see table 15.3.

	1	0	0	ϕ/80	250.0	312.5
			1	ϕ/100	200.0	250.0
		1	0	ϕ/112	178.6	223.2
			1	ϕ/128	156.3	195.3
1	0	0	0	ϕ/56	357.1	446.4*
			1	ϕ/80	250.0	312.5
		1	0	ϕ/96	208.3	260.4
			1	ϕ/128	156.3	195.3
	1	0	0	ϕ/160	125.0	156.3
			1	ϕ/200	100.0	125.0
		1	0	ϕ/224	89.3	111.6
			1	ϕ/256	78.1	97.7

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 3)

	1	0	0	ϕ/160	125.0	156.3
			1	ϕ/200	100.0	125.0
		1	0	ϕ/224	89.3	111.6
			1	ϕ/256	78.1	97.7
1	0	0	0	ϕ/112	178.6	223.2
			1	ϕ/160	125.0	156.3
		1	0	ϕ/190	104.2	130.2
			1	ϕ/256	78.1	97.7
	1	0	0	ϕ/320	62.5	78.1
			1	ϕ/400	50.0	62.5
		1	0	ϕ/448	44.6	55.8
			1	ϕ/512	39.1	48.8

Note: * The correct operation cannot be guaranteed since the value is outside the I^2C interface specifications (high-speed mode: max. 400 kHz).
(n = 0 to 3)

				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 o 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: * If the BBSY bit is written to, the value of the flag is not changed.

I²C bus format master mode:

- When a start condition is detected in the bus l
after a start condition is issued (when the ICD
set to 1 because of first frame transmission)
- When a wait is inserted between the data and
acknowledge bit when the WAIT bit is 1 (fall o
transmit/receive clock)
- At the end of data transfer (rise of the 9th
transmit/receive clock)
- When a slave address is received after bus m
is lost
- If 1 is received as the acknowledge bit (when
bit in ICSR is set to 1) when the ACKE bit is 1
- When the AL flag is set to 1 after bus masters
while the ALIE bit is 1

I²C bus format slave mode:

- When the slave address (SVA or SVAX) matc
(when the AAS or AASX flag in ICSR is set to
the end of data transfer up to the subsequent
retransmission start condition or stop conditio
detection (rise of the 9th clock)
- When the general call address is detected (wl
is received for R/$\overline{W}$ bit, and ADZ flag in ICSR
1) and at the end of data reception up to the
subsequent retransmission start condition or s
condition detection (rise of the 9th receive clo
- When 1 is received as an acknowledge bit wh
ACKE bit is 1 (when the ACKB bit is set to 1)
- When a stop condition is detected while the S
is 0 (when the STOP or ESTP flag in ICSR is

- When transmitting the data in the ICDR registe
  (when data is transferred from ICDRT to ICDR
  transmit mode and the ICDRE flag is set to 1,
  is transferred from ICDRS to ICDRR in receive
  and the ICDRF flag is set to 1.)

[Clearing conditions]

- When 0 is written in IRIC after reading IRIC =
- When ICDR is accessed by DTC * (This may
  clearing condition. For details, see the descrip
  the DTC operation on the next page.

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

Even when the IRIC flag and IRTR flag are set, the ICDRE or ICDRF flag may not be s
IRIC and IRTR flags are not cleared at the end of the specified number of transfers in co
transfer using the DTC. The ICDRE or ICDRF flag is cleared, however, since the specif
number of ICDR reads or writes have been completed.

Tables 15.4 and 15.5 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 aft 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

													n fr (
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 th s
0	0	1	0	0	1↑/0 *2	—	—	—	—	—	1↑	—	F 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 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 ARSA after reading ARSA = 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 SC 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 ge 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 0 transmission is ended and a stop condition is issu master mode, or before transmission is ended and released to issue a stop condition by a master dev

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

				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 fr
ICDRS to ICDRR. Be sure to read data from ICDR
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 ICDI
  1. When data is transmitted completely whil
     = 0 (at the rise of the 9th clock pulse).
  2. When data is written to ICDR completely
     mode after data was transmitted while ICI

[Clearing conditions]

- When data is written to ICDR (ICDRT).
- When the stop condition is detected in I^2C bu
  or serial format.
- When 0 is written to the ICE bit.

Note that if the ACKE bit is set to 1 in I^2C bus for
enabling acknowledge bit decision, ICDRE is not
data is transmitted completely while the acknowl
1.

When ICDRE is set due to the condition (2) abov
is temporarily cleared to 0 when data is written to
(ICDRT); however, since data is transferred from
ICDRS immediately, ICDRF is set to 1 again. Do
data to ICDR when TRS = 0 because the ICDRE
value is invalid during the time.

				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 15.6, 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	—	—	—	Reserved
6	—	—	—	These bits cannot be modified. The read values a undefined.
5	SMB3E	All 0	R/W	SMBus Enable
4	SMB2E			These bits enable/disable to support the SMBus, combining with bits FSEL1 and FSEL0. The SME controls IIC_3, the SMB2E bit controls IIC_2, the bit controls IIC_1, the SMB0E bit controls IIC_0.
3	SMB1E			
2	SMB0E			
				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 15.7.

1	0	Min.	300	240*
		Max.	550	440
	1	Min.	500	400
		Max.	950	760

[Legend] n = 0 to 3

Note: * Since the value is outside the SMBus specification, it should not be set.

Table 15.7 ISCMBCR Setting

System Clock	SMBnE	FSEL1	FSEL0
20 MHz	1	1	0
20 to 25 MHz	1	1	1

[Legend] n = 0 to 3

Figure 15.5 shows the I²C bus timing.

The symbols used in figures 15.3 to 15.5 are explained in table 15.8.

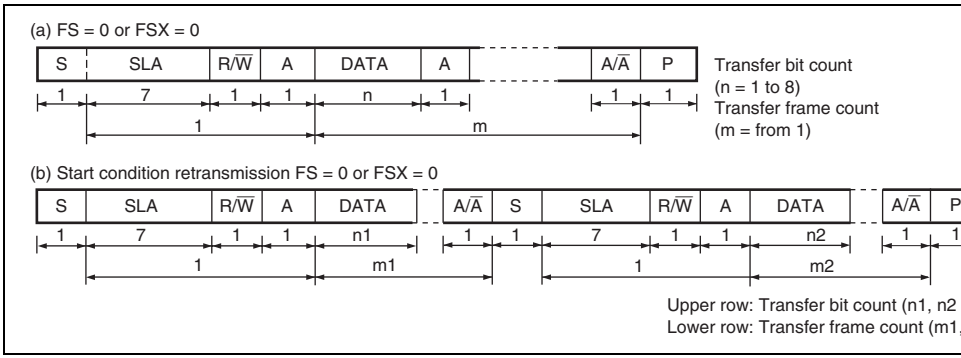


Figure 15.3 I²C Bus Data Formats (I²C Bus Formats)

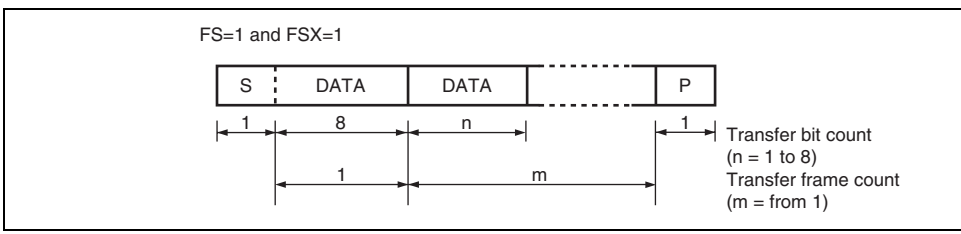


Figure 15.4 I²C Bus Data Formats (Serial Formats)

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

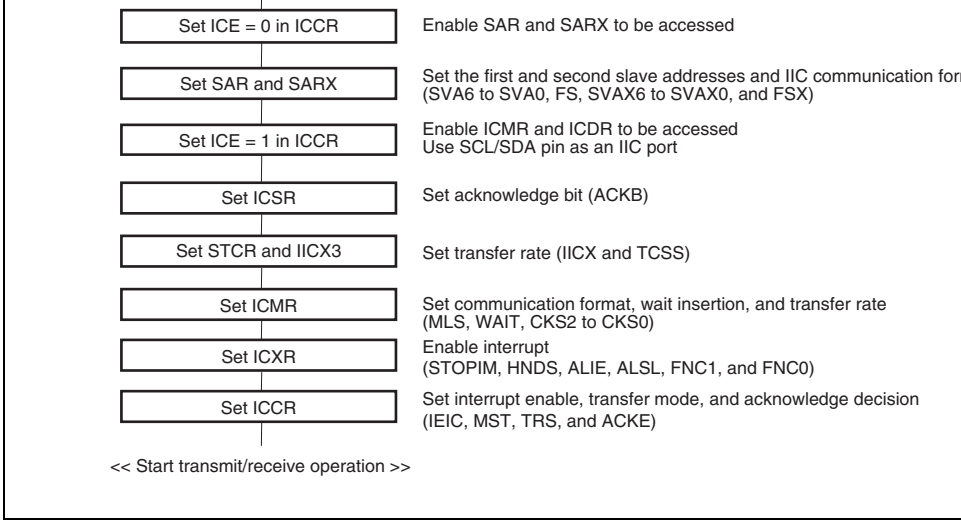


Figure 15.6 Sample Flowchart for IIC Initialization

Note: Be sure to modify the ICMR register after transmit/receive operation has been c
If the ICMR register is modified during transmit/receive operation, bit counter B
BC0 will be modified erroneously, thus causing incorrect operation.

15.4.3 Master Transmit Operation

In I^2C bus format master transmit mode, the master device outputs the transmit clock an
data, and the slave device returns an acknowledge signal.

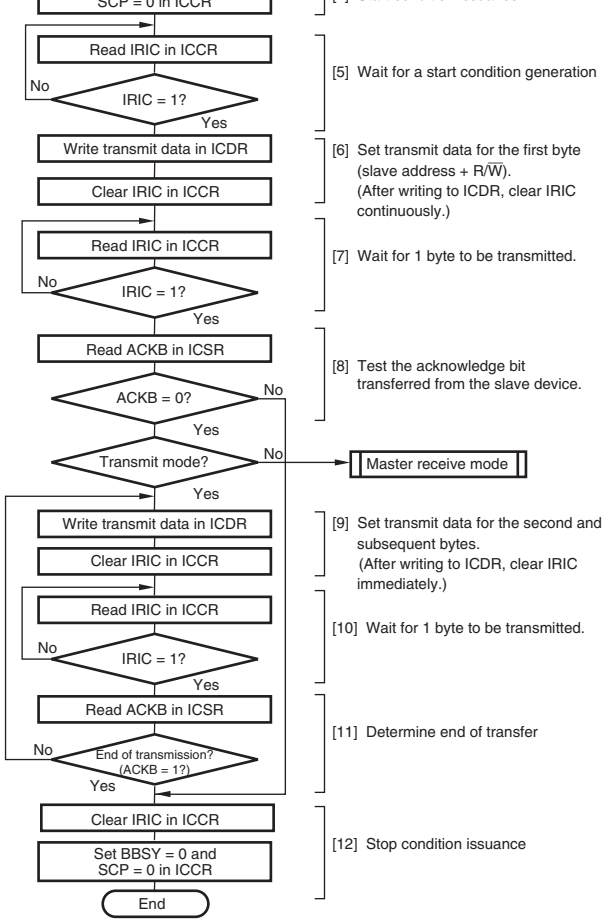


Figure 15.7 Sample Flowchart for Operations in Master Transmit Mode

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 frame data following the start condition indicates the 7-bit slave address and transm direction (R/$\overline{W}$).

   To determine the end of the transfer, the IRIC flag is cleared to 0. After writing to clear IRIC continuously so no other interrupt handling routine is executed. If the ti transmission of one frame of data has passed before the IRIC clearing, the end of transmission cannot be determined. The master device sequentially sends the trans clock and the data written to ICDR. The selected slave device (i.e. the slave device matching slave address) drives SDA low at the 9th transmit clock pulse and returns acknowledge signal.

7. When one frame of data has been transmitted, the IRIC flag is set to 1 at the rise of transmit clock pulse. After one frame has been transmitted, SCL is automatically fi synchronization with the internal clock until the next transmit data is written.

8. Read the ACKB bit in ICSR to confirm that ACKB is cleared to 0. When the slave has not acknowledged (ACKB bit is 1), operate step [12] to end transmission, and transmit operation.

9. Write the transmit data to ICDR.

   As indicating the end of the transfer, the IRIC flag is cleared to 0. Perform the ICD and the IRIC flag clearing sequentially, just as in step [6]. Transmission of the next performed in synchronization with the internal clock.

10. When one frame of data has been transmitted, the IRIC flag is set to 1 at the rise of transmit clock pulse. After one frame has been transmitted, SCL is automatically fi synchronization with the internal clock until the next transmit data is written.

11. Read the ACKB bit in ICSR.

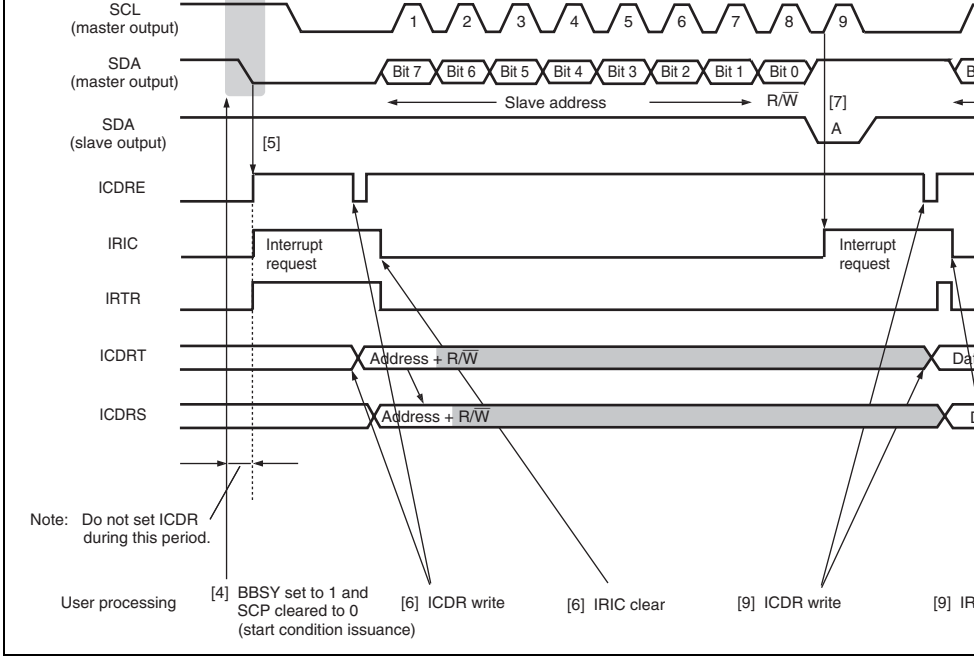


Figure 15.8 Operation Timing Example in Master Transmit Mode (MLS = WA

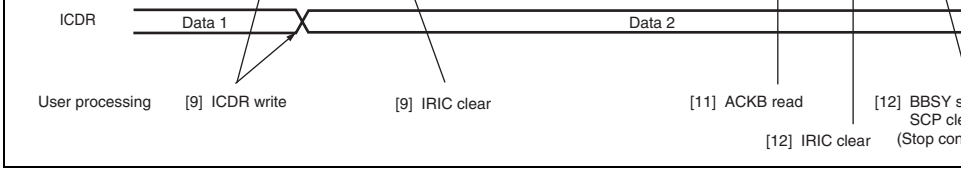


Figure 15.9 Stop Condition Issuance Operation Timing Example in Master Trans (MLS = WAIT = 0)

15.4.4 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.

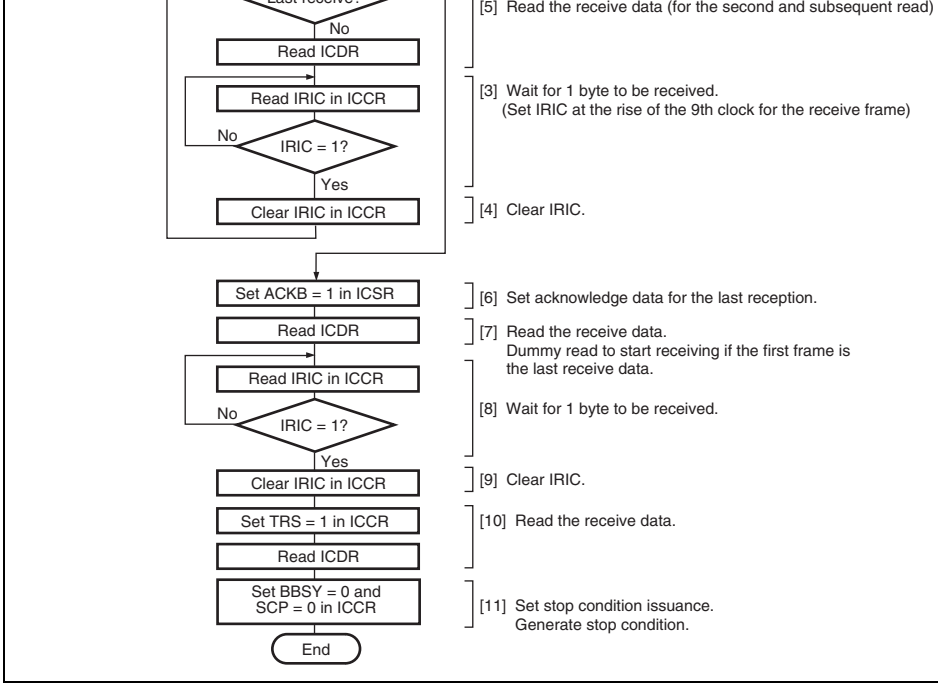


Figure 15.10 Sample Flowchart for Operations in Master Receive Mode (HNDS

pin is sequentially transferred to ICDRS in synchronization with the rise of the re
clock pulses.)

3. The master device drives SDA low to return the acknowledge data at the 9th rece
pulse. The receive data is transferred to ICDRR from ICDRS at the rise of the 9th
pulse, setting the ICDRF, IRIC, and IRTR flags to 1. If the IEIC bit has been set
interrupt request is sent to the CPU.

   The master device drives SCL low from the fall of the 9th receive clock pulse to
data reading.

4. Clear the IRIC flag to determine the next interrupt.

   Go to step [6] to halt reception operation if the next frame is the last receive data.

5. Read ICDR receive data. This clears the ICDRF flag to 0. The master device outp
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 outp
receive clock to receive data.
8. When one frame of data has been received, the ICDRF, IRIC, and IRTR flags are
the rise of the 9th receive clock pulse.
9. Clear the IRIC flag to 0.
10. Read ICDR receive data after setting the TRS bit. This clears the ICDRF flag to 0
11. Clear the BBSY bit and SCP bit to 0 in ICCR. This changes SDA from low to hig
SCL is high, and generates the stop condition.

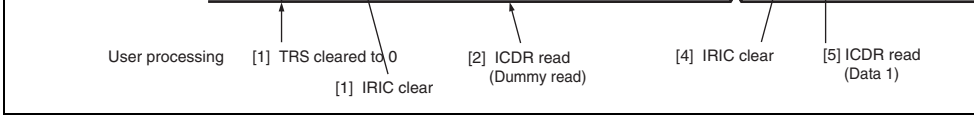


Figure 15.11 Master Receive Mode Operation Timing Example (MLS = WAIT = 0, HNDS = 1)

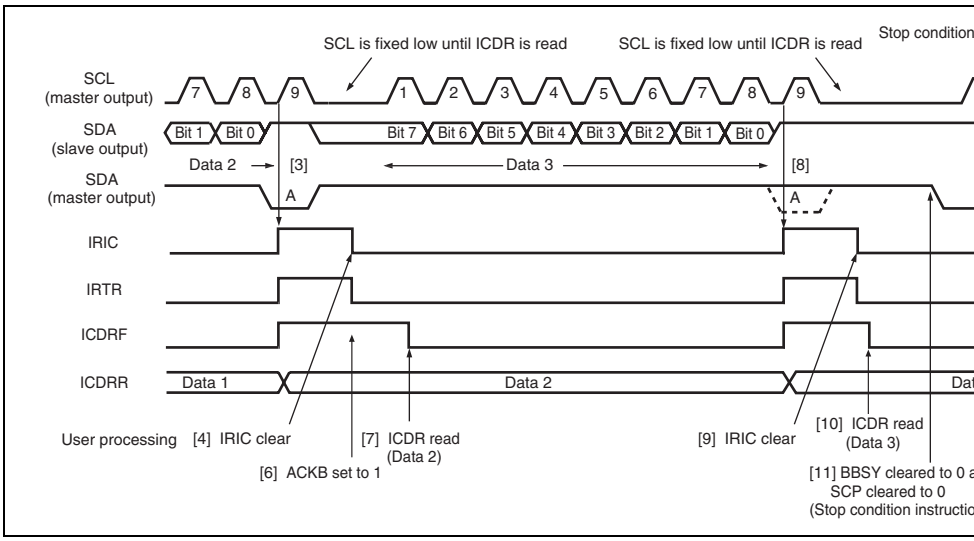


Figure 15.12 Stop Condition Issuance Timing Example in Master Receive Mo (MLS = WAIT = 0, HNDS = 1)

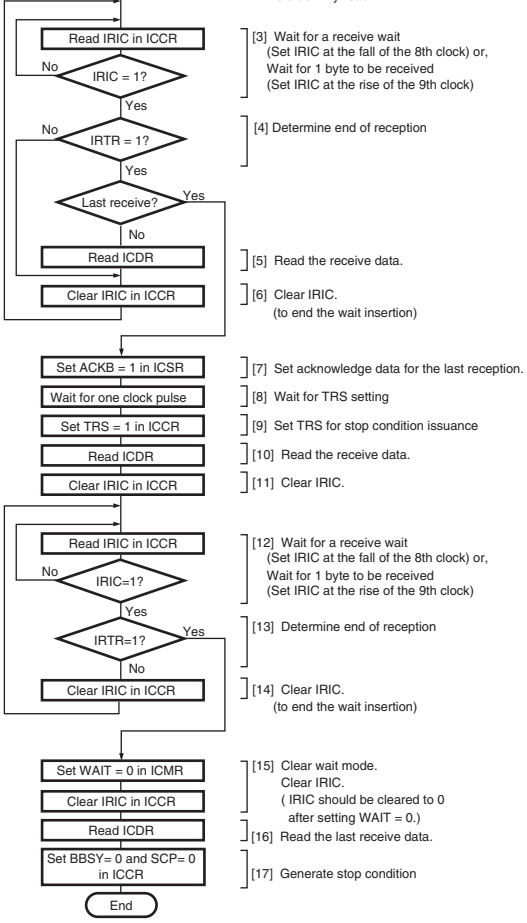


Figure 15.13 Sample Flowchart for Operations in Master Receive Mode (receiving multiple bytes) (WAIT = 1)

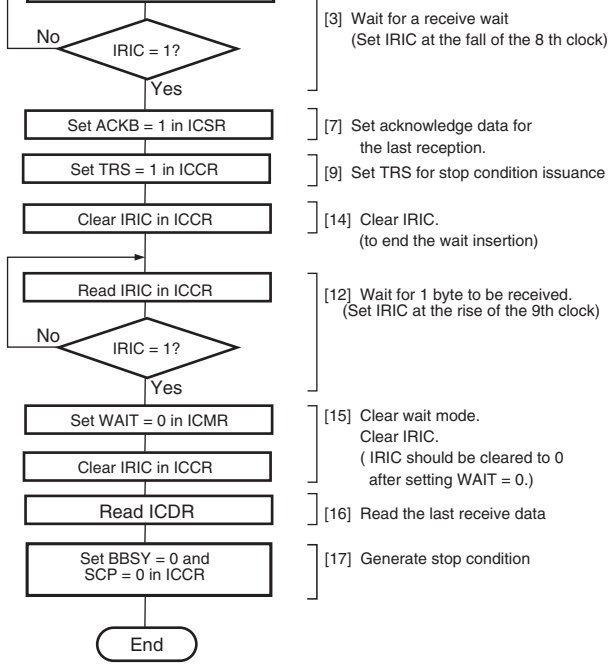


Figure 15.14 Sample Flowchart for Operations in Master Receive Mode (receiving a single byte) (WAIT = 1)

The reception procedure and operations using the wait function (WAIT bit), by which 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 15.14.

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.
12. The IRIC flag is set to 1 in either of the following cases.

    (1) At the fall of the 8th receive clock pulse for one frame

    SCL is automatically fixed low in synchronization with the internal clock unt flag is cleared.

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.

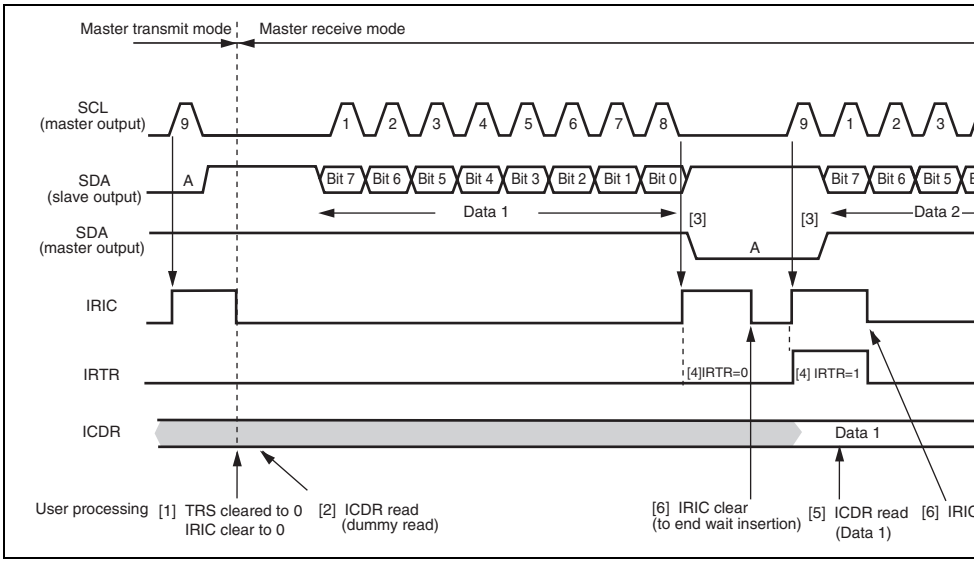


Figure 15.15 Master Receive Mode Operation Timing Example (MLS = ACKB = 0, WAIT = 1)

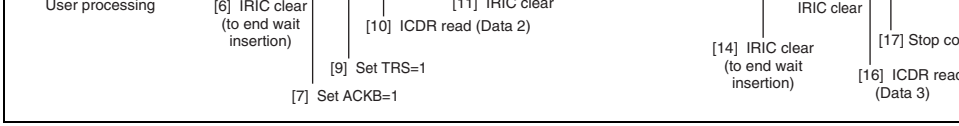


Figure 15.16 Stop Condition Issuance Timing Example in Master Receive M
(MLS = ACKB = 0, WAIT = 1)

15.4.5 Slave Receive Operation

In I^{2}C bus format slave receive mode, the master device outputs the transmit clock and t
data, and the slave device returns an acknowledge signal.

The slave device operates as the device specified by the master device when the slave ac
the first frame following the start condition that is issued by the master device matches i
address.

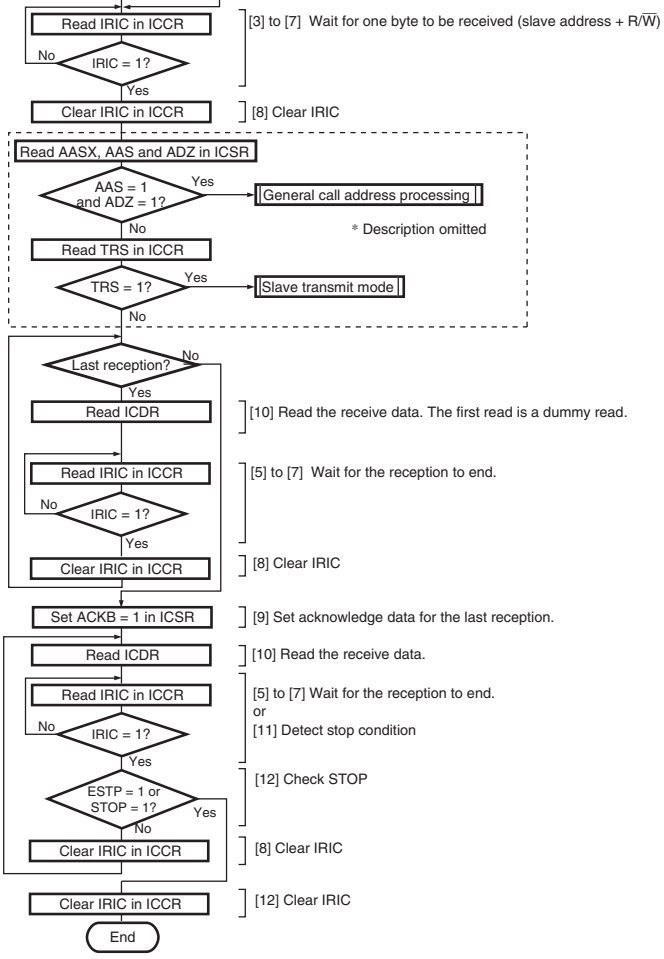


Figure 15.17 Sample Flowchart for Operations in Slave Receive Mode (HNDS

direction (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 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 d (R/$\overline{W}$) is 1, the TRS bit is set to 1, and slave transmit operation is performed. Wher address does not match, receive operation is halted until the next start condition is
5. At the 9th clock pulse of the receive frame, the slave device returns the data in the as the acknowledge data.
6. At the rise of the 9th clock pulse, the IRIC flag is set to 1. If the IEIC bit has been interrupt request is sent to the CPU.

   If the AASX bit has been set to 1, IRTR flag is also set to 1.
7. At the rise of the 9th clock pulse, the receive data is transferred from ICDRS to IC setting the ICDRF flag to 1. The slave device drives SCL low from the fall of the 9 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 er 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 the BBSY flag is cleared to 0 and the STOP bit is set to 1. If the STOPIM bit has b cleared to 0, the IRIC flag is set to 1.
12. Confirm that the STOP bit is set to 1, and clear the IRIC flag to 0.

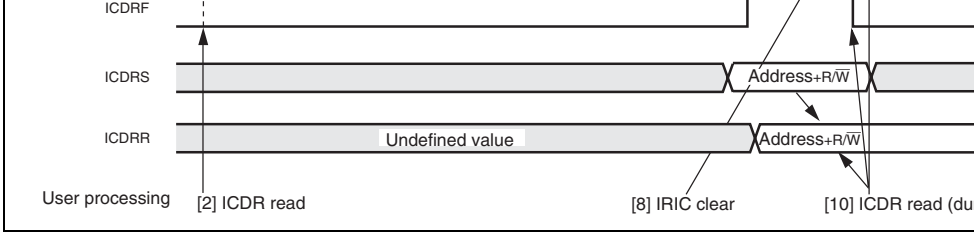


Figure 15.18 Slave Receive Mode Operation Timing Example (1) (MLS = 0, HNI

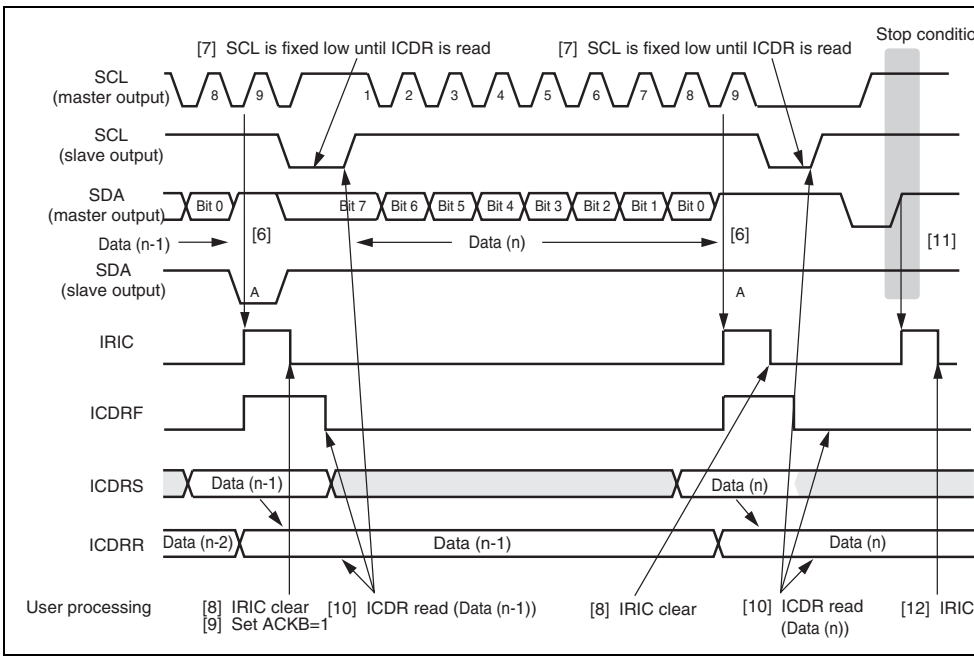


Figure 15.19 Slave Receive Mode Operation Timing Example (2) (MLS = 0, HNI

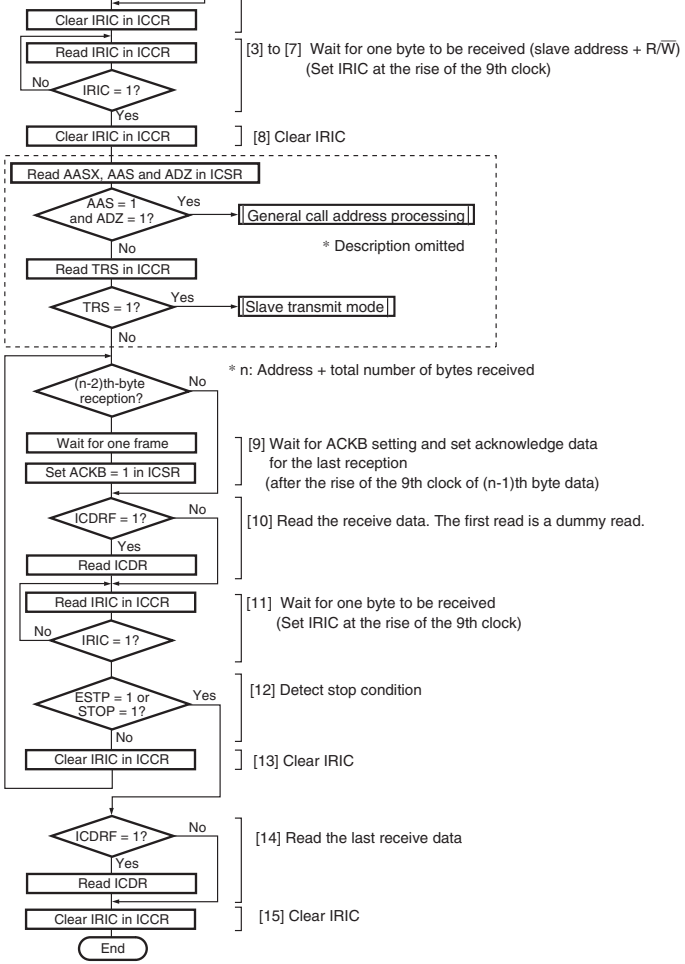


Figure 15.20 Sample Flowchart for Operations in Slave Receive Mode (HND

4. When the slave address matches in the first frame following the start condition, the
operates as the slave device specified by the master device. If the 8th data bit (R/$\overline{W}$)
TRS bit remains cleared to 0, and slave receive operation is performed. If the 8th da
(R/$\overline{W}$) is 1, the TRS bit is set to 1, and slave transmit operation is performed. When
address does not match, receive operation is halted until the next start condition is d
5. At the 9th clock pulse of the receive frame, the slave device returns the data in the A
as the acknowledge data.
6. At the rise of the 9th clock pulse, the IRIC flag is set to 1. If the IEIC bit has been se
interrupt request is sent to the CPU.

   If the AASX bit has been set to 1, 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 t
ACKB bit. Set the ACKB bit after the rise of the 9th clock pulse of the second last r
frame.
10. Confirm that the ICDRF flag is set to 1 and read ICDR. This clears the ICDRF flag
11. At the rise of the 9th clock pulse or when the receive data is transferred from 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
the BBSY flag is cleared to 0 and the STOP or ESTP flag is set to 1. If the STOPIM
been cleared to 0, the IRIC flag is set to 1. In this case, execute step 14 to read the la
receive data.
13. Clear the IRIC flag to 0.

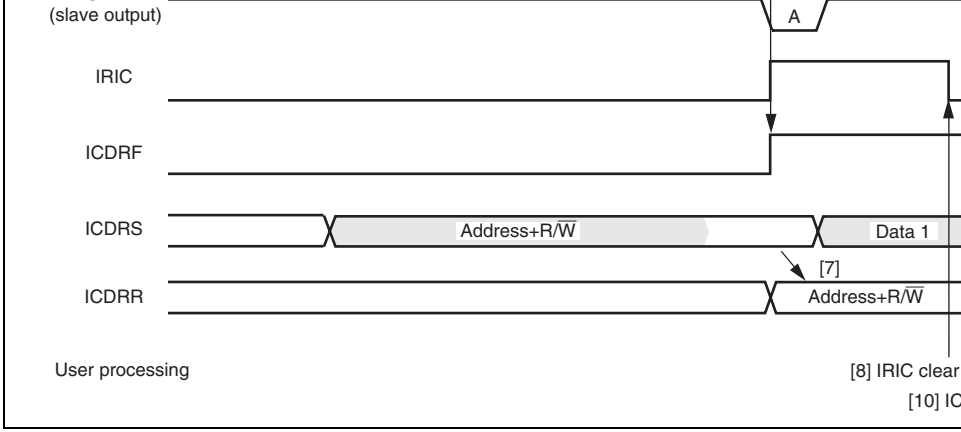


Figure 15.21 Slave Receive Mode Operation Timing Example (1)
(MLS = ACKB = 0, HNDS = 0)

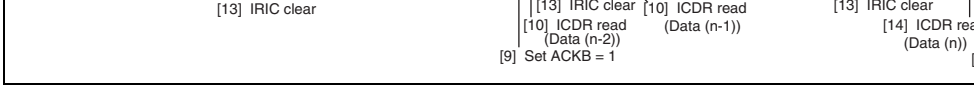


Figure 15.22 Slave Receive Mode Operation Timing Example (2)
(MLS = ACKB = 0, HNDS = 0)

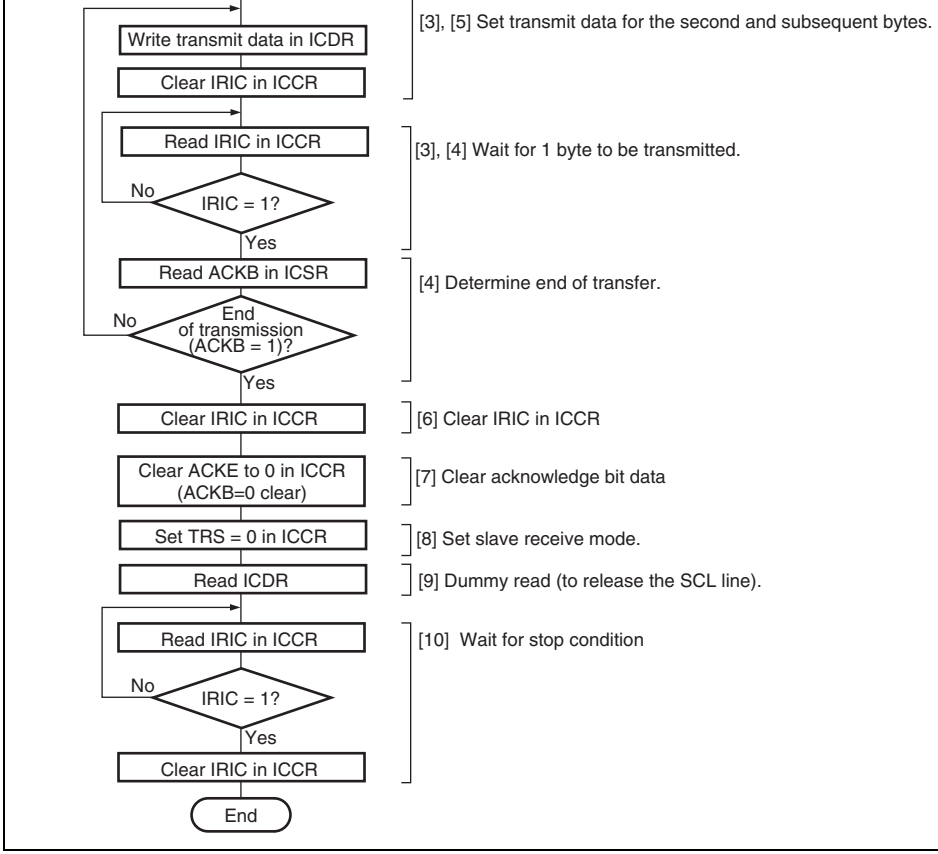


Figure 15.23 Sample Flowchart for Slave Transmit Mode

clock until ICDR data is written, to disable the master device to output the next tran
clock.

3. After clearing the IRIC flag to 0, write data to ICDR. At this time, the ICDRE flag i
to 0. The written data is transferred to ICDRS, and the ICDRE and IRIC flags are se
again. The slave device sequentially sends the data written into ICDRS in accordanc
the clock 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
other interrupt processing from being inserted.

4. The master device drives SDA low at the 9th clock pulse, and returns an acknowled
As this acknowledge signal is stored in the ACKB bit in ICSR, this bit can be used t
determine whether the transfer operation was performed successfully. When one fra
data has been transmitted, the IRIC flag in ICCR is set to 1 at the rise of the 9th tran
clock pulse. When the ICDRE flag is 0, the data written into ICDR is transferred to
and the ICDRE and IRIC flags are set to 1 again. If the ICDRE flag has been set to
slave device drives SCL low from the fall of the 9th transmit clock until data is writ
ICDR.

5. To continue transmission, write the next data to be transmitted into ICDR. The ICD
cleared to 0. The IRIC flag is cleared to 0 to detect the end of transmission. Processi
the ICDR register writing to the IRIC flag clearing should be performed continuousl
Prevent any other interrupt processing from being inserted.

   Transmit operations can be performed continuously by repeating steps 4 and 5.

6. Clear the IRIC flag to 0.
7. To end transmission, clear the ACKE bit in the ICCR register to 0, to clear the ackn
bit stored in the ACKB bit to 0.
8. Clear the TRS bit to 0 for the next address reception, to set slave receive mode.

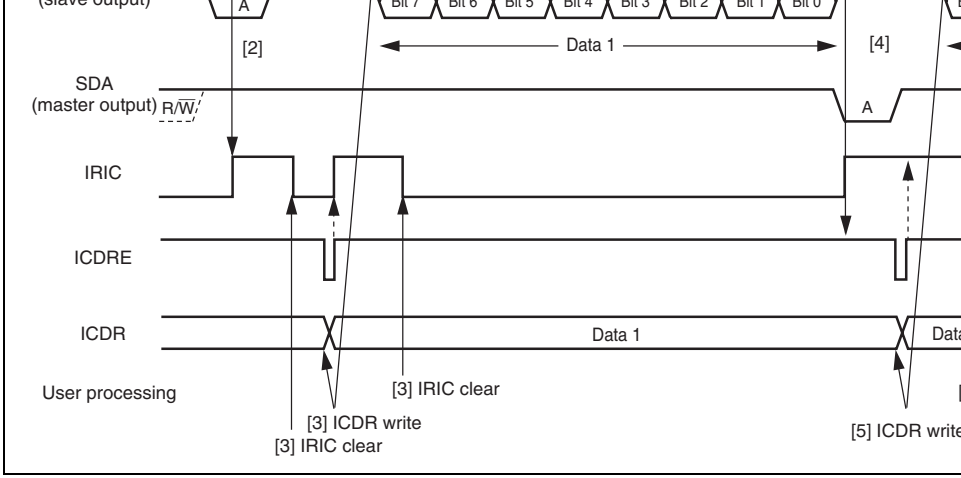
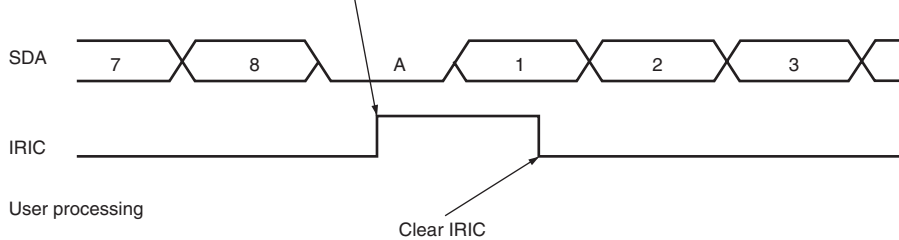
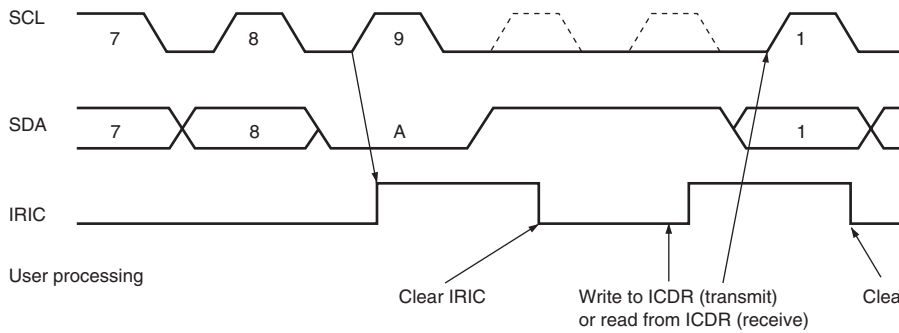


Figure 15.24 Slave Transmit Mode Operation Timing Example (MLS = 0)



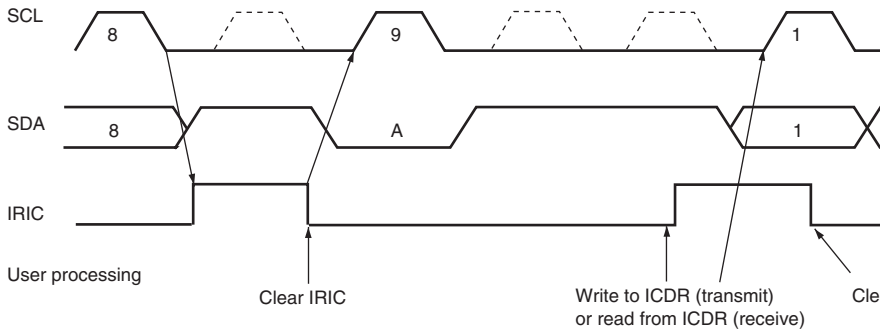
(a) Data transfer ends with ICDRE=0 at transmission, or ICDRF=0 at reception.



(b) Data transfer ends with ICDRE=1 at transmission, or ICDRF=1 at reception.

Figure 15.25 IRIC Setting Timing and SCL Control (1)

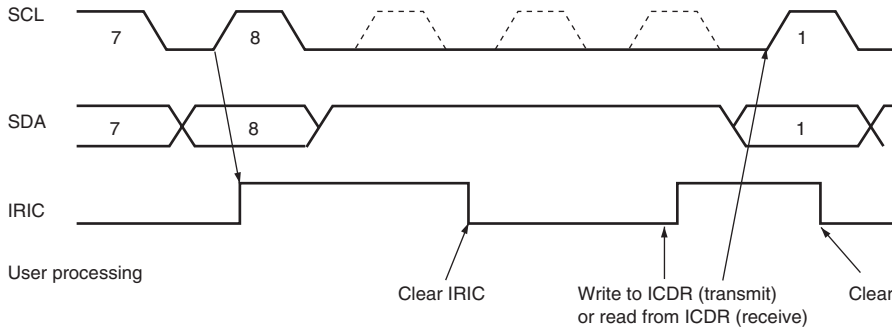
(a) Data transfer ends with ICDRE=0 at transmission, or ICDRF=0 at reception.



(b) Data transfer ends with ICDRE=1 at transmission, or ICDRF=1 at reception.

Figure 15.26 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 15.27 IRIC Setting Timing and SCL Control (3)

transmission is completed with the acknowledge bit value of 1 when the ACKE bit is 1,
not initiated, thus allowing an interrupt to be generated if enabled.

The acknowledge bit may indicate specific events such as completion of receive data pr
for some receiving devices, and for other receiving devices, the acknowledge bit may be
indicating no specific events.

The I^2C bus format provides for selection of the slave device and transfer direction by m
the slave address and the R/$\overline{W}$ bit, confirmation of reception with the acknowledge bit, i
of the last frame, and so on. Therefore, continuous data transfer using the DTC must be
in conjunction with CPU processing by means of interrupts.

Table 15.9 shows some examples of processing using the DTC. These examples assume
number of transfer data bytes is known in slave mode.

transmission/ reception	DTC (ICDR write)	DTC (ICDR read)	DTC (ICDR write)	(ICDR read
Dummy data (H'FF) write	—	—	Processing by DTC (ICDR write)	—
Last frame processing	Not necessary	Reception by CPU (ICDR read)	Not necessary	Reception (ICDR read
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 necess
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 count

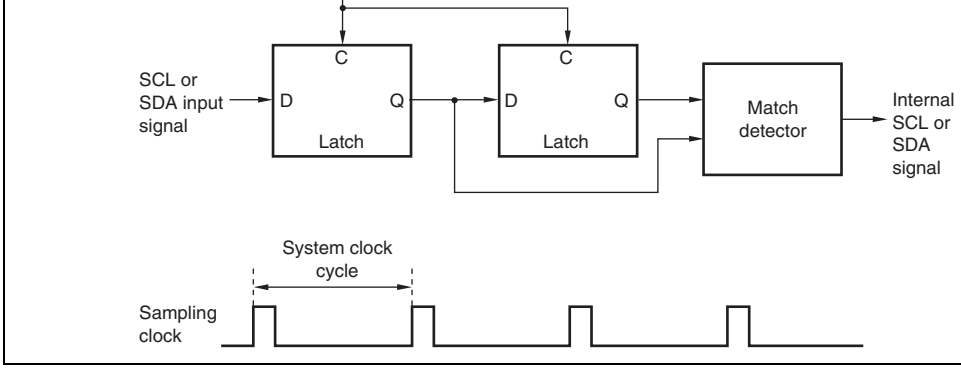


Figure 15.28 Block Diagram of Noise Canceler

15.4.10 Initialization of Internal State

The IIC has a function for forcible initialization of its internal state if a deadlock occurs
communication.

Initialization is executed in accordance with clearing ICE bit.

Scope of Initialization: The initialization executed by this function covers the followin

- 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,
  output, etc.)

- Interrupt flags and interrupt sources are not cleared, and so flag clearing measures mu taken as necessary.
- Basically, other register flags are not cleared either, and so flag clearing measures mu taken as necessary.
- If a flag clearing setting is made during transmission/reception, the IIC module will st transmitting/receiving at that point and the SCL and SDA pins will be released. When transmission/reception is started again, register initialization, etc., must be carried out necessary to enable correct communication as a system.

The value of the BBSY bit cannot be modified directly by this module clear function, but 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 bit 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

2. Either of the following two conditions will start the next transfer. Pay attention to thes conditions when accessing to ICDR.
   — Write to ICDR when ICE = 1 and TRS = 1 (including automatic transfer from ICD ICDRS)
   — Read from ICDR when ICE = 1 and TRS = 0 (including automatic transfer from I ICDRR)
3. Table 15.11 shows the timing of SCL and SDA outputs in synchronization with the in clock. Timings on the bus are determined by the rise and fall times of signals affected bus load capacitance, series resistance, and parallel resistance.

Table 15.11 I²C Bus Timing (SCL and SDA Outputs)

Item	Symbol	Output Timing	Unit	Not
SCL output cycle time	t_{SCLO}	$28t_{cyc}$ to $512t_{cyc}$	ns	See
SCL output high pulse width	t_{SCLHO}	$0.5t_{SCLO}$	ns	24.1
SCL output low pulse width	t_{SCLLO}	$0.5t_{SCLO}$	ns	(ref
SDA output bus free time	t_{BUFO}	$0.5t_{SCLO} – 1t_{cyc}$	ns	
Start condition output hold time	t_{STAHO}	$0.5t_{SCLO} – 1t_{cyc}$	ns	
Retransmission start condition output setup time	t_{STASO}	$1t_{SCLO}$	ns	
Stop condition output setup time	t_{STOSO}	$0.5t_{SCLO} + 2t_{cyc}$	ns	
Data output setup time (master)	t_{SDASO}	$1t_{SCLLO} – 3t_{cyc}$	ns	
Data output setup time (slave)		$1t_{SCLLO} – (6t_{cyc}$ or $12t_{cyc}$*)		
Data output hold time	t_{SDAHO}	$3t_{cyc}$	ns	

Note: * $6t_{cyc}$ when IICXn is 0, $12t_{cyc}$ when IICXn is 1 (n = 0 to 3).

Table 15.12 Permissible SCL Rise Time (t_{sr}) Values

				Time Indication [ns]		
TCSS	**IICXn**	**t_{cyc} Indication**		**I²C Bus Specification (Max.)**	**ϕ = 20 MHz**	**ϕ = 2 MHz**
0	0	7.5 t_{cyc}	Standard mode	1000	375	300
			High-speed mode	300	300	300
	1	17.5 t_{cyc}	Standard mode	1000	875	700
1	0		High-speed mode	300	300	300
1	1	37.5 t_{cyc}	Standard mode	1000	1000	1000
			High-speed mode	300	300	300

[Legend] n = 0 to 3

6. The I²C bus interface specifications for the SCL and SDA rise and fall times are und and 300 ns. The I²C bus interface SCL and SDA output timing is prescribed by t_{cyc}, a table 15.11. However, because of the rise and fall times, the I²C bus interface specifi may not be satisfied at the maximum transfer rate. Table 15.13 shows output timing calculations for different operating frequencies, including the worst-case influence o fall times.

   t_{BUFO} fails to meet the I²C bus interface specifications at any frequency. The solution i to provide coding to secure the necessary interval (approximately 1 μs) between issu stop condition and issuance of a start condition, or (b) to select devices whose input permits this output timing for use as slave devices connected to the I²C bus.

t_{SCLLO}	$0.5\ t_{SCLO}\ (-t_{Sr})$	Standard mode	–250	4700	4750	423
		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]	344
		High-speed mode	–300	1300	850*[1]	780
t_{STAHO}	$0.5\ t_{SCLO} - 1\ t_{cyc}$ $(-t_{Sr})$	Standard mode	–250	4000	4700	419
		High-speed mode	–250	600	900	830
t_{STASO}	$1\ t_{SCLO}\ (-t_{Sr})$	Standard mode	–1000	4700	9000	796
		High-speed mode	–300	600	2100	194
t_{STOSO}	$0.5\ t_{SCLO} + 2\ t_{cyc}$ $(-t_{Sr})$	Standard mode	–1000	4000	4100	356
		High-speed mode	300	600	1000	900
t_{SDASO} (master)	$1\ t_{SCLLO}$*[3] $-3\ t_{cyc}$ $(-t_{Sr})$	Standard mode	–1000	250	3600	310
		High-speed mode	–300	100	500	450
t_{SDASO} (slave)	$1\ t_{SCLL}$*[3] $-12\ t_{cyc}$*[2] $(-t_{Sr})$	Standard mode	–1000	250	3100	322
		High-speed mode	–300	100	400	520
t_{SDAHO}	$3\ t_{cyc}$	Standard mode	0	0	150	120
		High-speed mode	0	0	150	120

Notes: 1. Does not meet the I²C bus interface specification. Remedial action such as th 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 T 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 I²C bus in specifications are met must be determined in accordance with the actual setti conditions.

2. Value when the IICXn bit is set to 1. When the IICXn bit is cleared to 0, the va $(-6t_{cyc})$ (n = 0 to 3).

mode (i.e. with the TRS bit cleared to 0). When reading the receive data, first confirm
BBSY bit in the ICCR register is cleared to 0, the stop condition has been generated, a
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 t
instruction for issuance of the stop condition (writing of 0 to BBSY and SCP in ICCR
actual generation of the stop condition, the clock may not be output correctly in subse
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 set
must be carried out during interval (a) in figure 15.29 (after confirming that the BBSY
been cleared to 0 in the ICCR register).

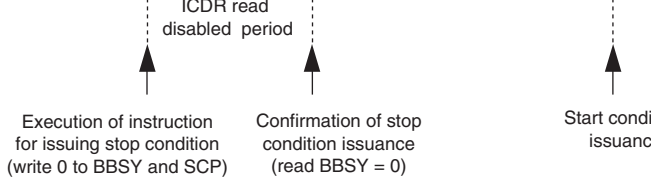


Figure 15.29 Notes on Reading Master Receive Data

Note: This restriction on usage can be canceled by setting the FNC1 and FNC0 bits to
ICXR.

8. Notes on start condition issuance for retransmission

   Figure 15.30 shows the timing of start condition issuance for retransmission, and the
   subsequently writing data to ICDR, together with the corresponding flowchart. Write
   transmit data to ICDR after the start condition for retransmission is issued and then t
   condition is actually generated.

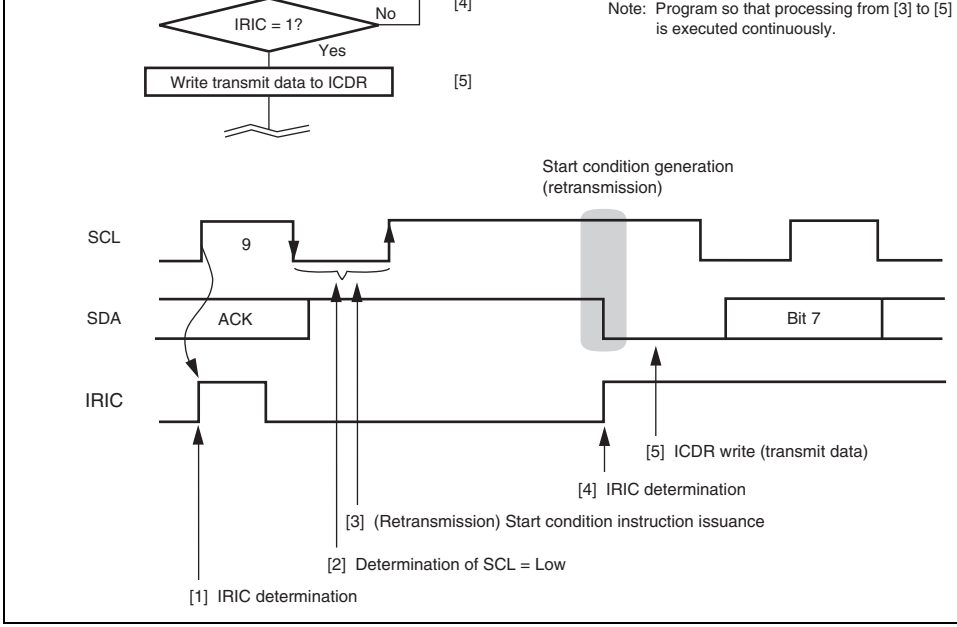


Figure 15.30 Flowchart for Start Condition Issuance Instruction for Retransmission and Timing

Note: This restriction on usage can be canceled by setting the FNC1 and FNC0 bits to E
ICXR.

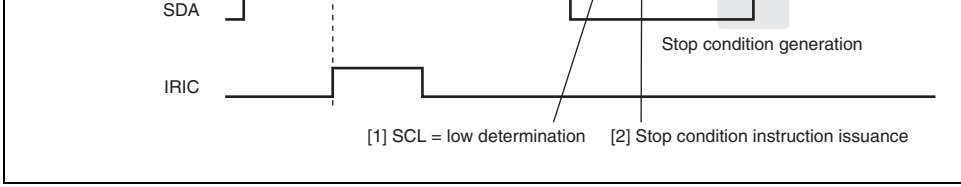


Figure 15.31 Stop Condition Issuance Timing

Note: This restriction on usage can be canceled by setting the FNC1 and FNC0 bits to
ICXR.

10. Note on IRIC flag clear when the wait function is used

When the wait function is used in I^2C bus interface master mode and in a situation w
rise time of SCL exceeds the stipulated value or where a slave device in which a wai
inserted by driving the SCL pin low is used, the IRIC flag should be cleared after de
that the SCL is low.

If the IRIC flag is cleared to 0 when WAIT = 1 while the SCL is extending the high
the SDA level may change before the SCL goes low, which may generate a start or s
condition erroneously.

Note: This restriction on usage can be canceled by setting the FNC1 and FNC0 bits to E
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 15.33. However, such read and write operations sour
problem in interrupt handling processing that is generated in synchronization with the
edge of the 9th clock pulse because the shaded time has passed before making the tra
interrupt handling.

To handle interrupts securely, be sure to keep either of the following conditions.

— Read ICDR data that has been received so far or read/write from/to ICCR before s
the receive operation of the next slave address.

— Monitor the BC2 to BC0 counter in ICMR; when the count is B'000 (8th or 9th cl
pulse), wait for at least two transfer clock times in order to read ICDR or read/wri
ICCR during the time other than the shaded time.

The rise of the 9th clock is detected

Figure 15.33 ICDR Register Read and ICCR Register Access Timing in Slave T Mode

Note: This restriction on usage can be canceled by setting the FNC1 and FNC0 bits to ICXR.

12. Note on TRS bit setting in slave mode

In I^2C bus interface slave mode, if the TRS bit value in ICCR is set after detecting th edge of the 9th clock pulse or the stop condition before detecting the next rising edg SCL pin (the time indicated as (a) in figure 15.34), the bit value becomes valid imme when it is set. However, if the TRS bit is set during the other time (the time indicate figure 15.34), the bit value is suspended and remains invalid until the rising edge of clock pulse or the stop condition is detected. Therefore, when the address is received restart condition is input without the stop condition, the effective TRS bit value rema (transmit mode) internally and thus the acknowledge bit is not transmitted after the a been received at the 9th clock pulse.

To receive the address in slave mode, clear the TRS bit to 0 during the time indicate figure 15.34. To release the SCL low level that is held by means of the wait function mode, clear the TRS bit to and then dummy-read ICDR.

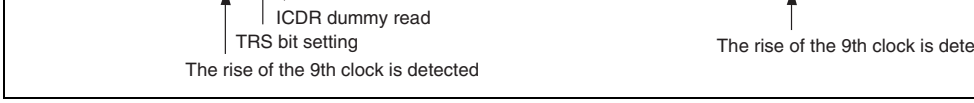


Figure 15.34 TRS Bit Set Timing in Slave Mode

Note: This restriction on usage can be canceled by setting the FNC1 and FNC0 bits to E
ICXR.

13. Note on ICDR read in transmit mode and ICDR write in receive mode

When ICDR is read in transmit mode (TRS = 1) or ICDR is written to in receive mod
0), the SCL pin may not be held low in some cases after transmit/receive operation ha
completed, thus inconveniently allowing clock pulses to be output on the SCL bus lin
ICDR is accessed correctly. To access ICDR correctly, read the ICDR after setting rec
mode or write to the ICDR after setting transmit mode.

of a series of transmit operation, clear the ACKE bit in ICCR once to initialize th
bit to 0.

— Set receive mode (TRS = 0) before the next start condition is input in slave mode
Complete transmit operation by the procedure shown in figure 15.23, in order to
from slave transmit mode to slave receive mode.

15. Notes on Arbitration Lost in Master Mode Operation

The I^2C bus interface recognizes the data in transmit/receive frame as an address whe
arbitration is lost in master mode and a transition to slave receive mode is automatica
carried out.

When arbitration is lost not in the first frame but in the second frame or subsequent f
transmit/receive data that is not an address is compared with the value set in the SAR
register as an address. If the receive data matches with the address in the SAR or SA
register, the I^2C bus interface erroneously recognizes that the address call has occurre
figure 15.35.)

In multi-master mode, a bus conflict could happen. When the I^2C bus interface is ope
master mode, check the state of the AL bit in the ICSR register every time after one
data has been transmitted or received.

When arbitration is lost during transmitting the second frame or subsequent frame, ta
avoidance measures.

• When the receive data matches to the address set in the SAR or SARX register, the I2C bus interface operates as a slave device.

Figure 15.35 Diagram of Erroneous Operation when Arbitration Lost

Though it is prohibited in the normal I^2C protocol, the same problem may occur when bit is erroneously set to 1 and a transition to master mode is occurred during data tran or reception in slave mode.

When the MST bit is set to 1 during data transmission or reception in slave mode, the arbitration decision circuit is enabled and arbitration is lost if conditions are satisfied. case, the transmit/receive data which is not an address may be erroneously recognized address.

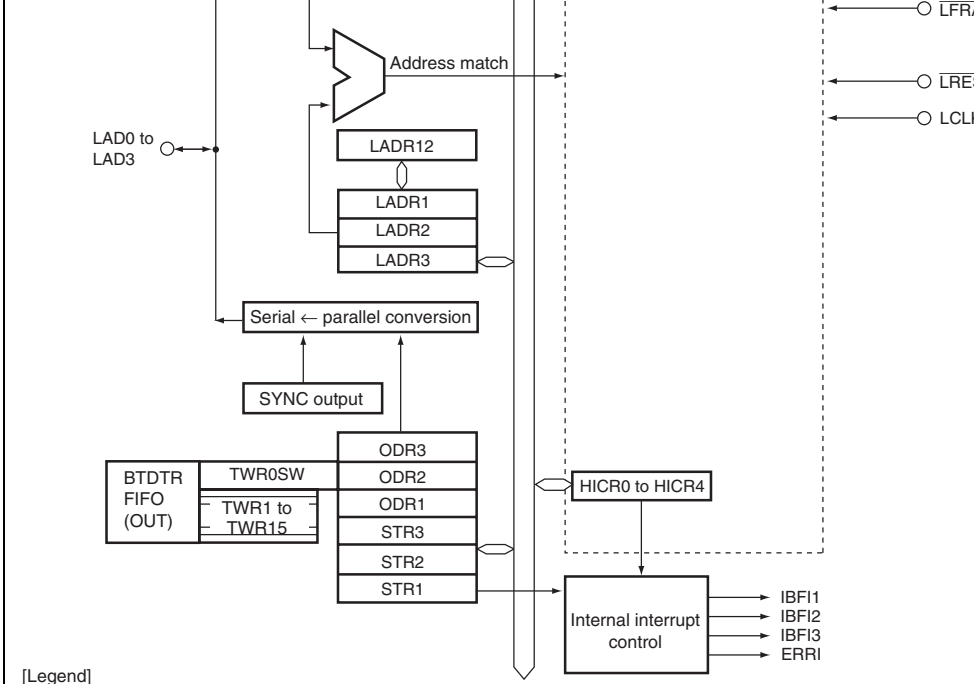
In multi-master mode, pay attention to the setting of the MST bit when a bus conflict occur. In this case, the MST bit in the ICCR register should be set to 1 according to th below.

A. Make sure that the BBSY flag in the ICCR register is 0 and the bus is free before the MST bit.
B. Set the MST bit to 1.
C. To confirm that the bus was not entered to the busy state while the MST bit is bein check that the BBSY flag in the ICCR register is 0 immediately after the MST bit set.

Note: Above restrictions can be released by setting the bits FNC1 and FNC2 in ICXR t

16.1 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.
  - For channel 3, sixteen bidirectional data register bytes can be manipulated in add the basic register set.
- Supports SERIRQ
  - Host interrupt requests are transferred serially on a single signal line (SERIRQ).
  - On channel 1, HIRQ1 and HIRQ12 can be generated.
  - On channels 2 and 3, SMI, HIRQ6, and HIRQ9 to HIRQ11 can be generated.
  - Operation can be switched between quiet mode and continuous mode.
- Supports version 1.5 of the Intelligent Platform Management Interface (IPMI) specif
  - Channel 3 supports the SMIC interface, KCS interface, and BT interface.



[Legend]

HICR0 to HICR4:	Host interface control registers 0 to 4	TWR0MW:	Bidirectional data regi
LADR12H, LADR12L:	LPC channel 1, 2 address registers 12H and 12L	TWR0SW:	Bidirectional data regi
LADR3H, LADR3L:	LPC channel 3 address registers 3H and 3L	TWR1 to TWR15:	Bidirectional data regi
IDR1 to IDR3:	Input data registers 1 to 3	SIRQCR0,1,2,4,5:	SERIRQ control regis
ODR1 to ODR3:	Output data registers 1 to 3	HISEL:	Host interface select r
STR1 to STR3:	Status registers 1 to 3		

Figure 16.1 Block Diagram of LPC

				termination signal
LPC reset	$\overline{\text{LRESET}}$	PE5	Input*	LPC interface reset signal
LPC clock	LCLK	PE6	Input	33-MHz PCI clock signal
Serialized interrupt request	SERIRQ	PE7	I/O*	Serialized host interrupt re signal (SMI, HIRQ1 to HIR synchronization with LCLK

Note: * Pin state monitoring input is possible in addition to the LPC interface control input/output function.

- 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)
- 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 4 (SIRQCR4)
- SERIRQ control register 5 (SIRQCR5)
- Host interface select register (HISEL)

- 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)

7	LPC3E	0	R/W	—	LPC Enable 3 to 1
6	LPC2E	0	R/W	—	Enable or disable the LPC interface function. W LPC interface is enabled (one of the three bits 1), processing for data transfer between the sl LSI) and the host is performed using pins LAD LAD0, $\overline{\text{LFRAME}}$, $\overline{\text{LRESET}}$, LCLK, and SERIR • LPC3E 0: LPC channel 3 operation is disabled No address (LADR3) matches for IDR3, O STR3, TWR0 to TWR15, SMIC, KCS, or B 1: LPC channel 3 operation is enabled • LPC2E 0: LPC channel 2 operation is disabled No address (LADR2) matches for IDR2, O STR2 1: LPC channel 2 operation is enabled
5	LPC1E	0	R/W	—	• LPC1E 0: LPC channel 1 operation is disabled No address (LADR1) matches for IDR1, O STR1 1: LPC channel 1 operation is enabled

					enabled [Clearing conditions] • Writing 0 • LPC hardware reset or LPC software rese [Setting condition] Writing 1 after reading SDWNE = 0
2 to 0	—	All 0	R/W	—	Reserved The initial value should not be changed.

- Cycle type or address indeterminate during cycle

[Clearing conditions]

- LPC hardware reset or LPC software reset
- LPC software shutdown
- Forced termination (abort) of transfer cycle to processing
- Normal termination of transfer cycle subjec processing

1: LPC interface is performing transfer cycle processing

[Setting condition]

Match of cycle type and address

					• There are no further interrupts for transfe host in quiet mode 1: LCLK restart request issued [Setting condition] In quiet mode, SERIRQ interrupt output beco necessary while LCLK is stopped
5	IRQBSY	0	R	—	SERIRQ Busy Indicates that the LPC interface's SERIRQ is in transfer processing. 0: SERIRQ transfer frame wait state [Clearing conditions] • LPC hardware reset or LPC software rese • 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 o LPC shutdown function, and the scope of initia by an LPC reset and an LPC shutdown, see se 16.4.5, LPC Interface Shutdown Function. 0: Normal state [Clearing conditions] • Writing 0 • LPC hardware reset or LPC software reset 1: LPC software shutdown state [Setting condition] Writing 1 after reading SDWNB = 0
2 to 0	—	All 0	R/W	—	Reserved The initial value should not be changed.

Bit	Bit Name	Value	Slave	Host	Description
7	—	Undefined	R	—	Reserved
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	—	0	R/(W)*	—	Reserved The initial value should not be changed.
4	ABRT	0	R/(W)*	—	LPC Abort Interrupt Flag This bit is a flag that generates an ERRI inte when a forced termination (abort) of an LPC cycle occurs. 0: [Clearing conditions] • Writing 0 after reading ABRT = 1 • LPC hardware reset ($\overline{\text{LRESET}}$ pin falling edge detection) • LPC software reset (LRSTB = 1) • LPC software shutdown (SDWNB = 1) 1: [Setting condition] $\overline{\text{LFRAME}}$ pin falling edge detection during L transfer cycle

					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
2	IBFIE2	0	R/W	—	IDR2 Receive Complete Interrupt Enable Enables or disables IBFI2 interrupt to the sla LSI). 0: Input data register (IDR2) receive complete interrupt requests disabled 1: Input data register (IDR2) receive complete interrupt requests enabled
1	IBFIE1	0	R/W	—	IDR1 Receive Complete Interrupt Enable Enables or disables IBFI1 interrupt to the sla LSI). 0: Input data register (IDR1) receive complete interrupt requests disabled 1: Input data register (IDR1) receive complete interrupt requests enabled

Bit	Bit Name	Initial Value	R/W Slave	R/W Host	Description
7	LFRAME	Undefined	R	—	0: $\overline{\text{LFRAME}}$ Pin state is lo 1: $\overline{\text{LFRAME}}$ Pin state is h
6	—	Undefined	R	—	Reserved
5	SERIRQ	Undefined	R	—	0: SERIRQ Pin state is lo 1: SERIRQ Pin state is hi
4	LRESET	Undefined	R	—	0: $\overline{\text{LRESET}}$ Pin state is lo 1: $\overline{\text{LRESET}}$ Pin state is hi
3	—	Undefined	R	—	Reserved
2	—	Undefined	R	—	Reserved
1	—	Undefined	R	—	Reserved
0	—	Undefined	R	—	Reserved

6	—	0	R/W	—	Reserved The initial value should not be changed.
5	CH2OFFSEL	0	R/W	—	Channel 2 Offset Selects the address offset for LPC chann 0: Offset 4 1: Offset 1
4	CH1OFFSEL	0	R/W	—	Channel 1 Offset Selects the address offset for LPC chann 0: Offset 4 1: Offset 1
3	SWENBL	0	R/W	—	In BT mode, H'5 (short wait) or H'6 (long returned to the host in the synchronized r cycle from slave, thus can make the host 0: Short wait is issued 1: Long wait is issued
2	KCSENBL	0	R/W	—	Enables or disables the use of the KCS i included in channel 3. When the LPC3E I HICR0 is 0, this bit is valid. 0: KCS interface operation is disabled No address (LADR3) matches for IDR or STR3 in KCS mode 1: KCS interface operation is enabled

	included in channel 3. When the LPCSE HICR0 is 0, this bit is valid. 0: BT interface operation is disabled No address (LADR3) matches for BT BTCR, or BTDTR 1: BT interface operation is enabled

16.3.4 LPC Channel 1, 2 Address Register H, L (LADR12H, LADR12L)

LADR12H and LADR12L are temporary registers for accessing internal registers LADR LADR1L, LADR2H, and LADR2L.

When the LADR12SEL bit in HICR4 is 0, LPC channel 1 host addresses (LADR1H, LA are set through LADR12. The contents of the address field in LADR1 must not be chang channel 1 is operating (while LPC1E is set to 1).

When the LADR12SEL bit is 1, LPC channel 2 host addresses (LADR2H, LADR2L) ar through LADR12. The contents of the address field in LADR2 must not be changed whi 2 is operating (while LPC2E is set to 1).

Table 16.2 shows the initial value of each register. Table 16.3 shows the host register se address match determination. Table 16.4 shows the slave selection internal registers in s LSI) access.

LADR1 (bits 15 to 3)	0	LADR1 (bit 1)	LADR1 (bit 0)	I/O write	IDR1 write (data) C/$\overline{D}$1 ← 0
LADR1 (bits 15 to 3)	1	LADR1 (bit 1)	LADR1 (bit 0)	I/O write	IDR1 write (comm C/$\overline{D}$1 ← 1
LADR1 (bits 15 to 3)	0	LADR1 (bit 1)	LADR1 (bit 0)	I/O read	ORD1 read
LADR1 (bits 15 to 3)	1	LADR1 (bit 1)	LADR1 (bit 0)	I/O read	STR1 read
LADR2 (bits 15 to 3)	0	LADR2 (bit 1)	LADR2 (bit 0)	I/O write	IDR2 write (data) C/$\overline{D}$2 ← 0
LADR2 (bits 15 to 3)	1	LADR2 (bit 1)	LADR2 (bit 0)	I/O write	IDR2 write (comm C/$\overline{D}$2 ← 1
LADR2 (bits 15 to 3)	0	LADR2 (bit 1)	LADR2 (bit 0)	I/O read	ODR2 read
LADR2 (bits 15 to 3)	1	LADR2 (bit 1)	LADR2 (bit 0)	I/O read	STR2 read

Table 16.4 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				The host address of LPC channel 3 is se
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 reg 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 Cycle	Host Registe Selection
Bits 15 to5	**Bit 4**	**Bit 3**	**Bit 2**	**Bit 1**	**Bit 0**		
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.

16.3.7 Output Data Registers 0 to 3 (ODR1 to ODR3)

The ODR registers are 8-bit readable/writable registers to the slave processor (this LSI), 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 16.3.4, LPC Channel 1, 2 Address Register H, L (LADR12H, LADR12L), and fo
information on ODR3 selection, see section 16.3.5, 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 16.5.5, LPC Channel 5 Address Register H, L (LADR5H, 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 16.3.5, 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 6 5 4	DBU17 DBU16 DBU15 DBU14	All 0	R/W	R	Defined by User The user can use these bits as necessary.
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.

					[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 writter 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 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 rese • Specification by SERIRQ transfer cycle s 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 initiatio request when a host interrupt request is clea quiet mode. 0: Start frame initiation is requested when al interrupt requests are cleared. 1: Start frame initiation is requested when o more interrupt requests are cleared.
5	IEDIR2	0	R/W	—	Interrupt Enable Direct Mode Specifies whether LPC channel 2 and chann SERIRQ interrupt source (SMI, IRQ6, IRQ9 IRQ11) generation is conditional upon OBF, controlled only by the host interrupt enable b 0: Host interrupt is requested when host inte enable and corresponding OBF bits are b to 1. 1: Host interrupt is requested when host inte 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 IR is disabled. [Clearing conditions] • Writing 0 to IRQ12E1 • LPC hardware reset, LPC software rese • Clearing OBF1 to 0 1: HIRQ12 interrupt request by setting OBF enabled. [Setting condition] Writing 1 after reading IRQ12E1 = 0

- Clearing OBF1 to 0

1: HIRQ1 interrupt request by setting OBF1
enabled

[Setting condition]

Writing 1 after reading IRQ1E1 = 0

					IRQE11E3 is disabled [Clearing conditions] • Writing 0 to IRQ11E3 • LPC hardware reset, LPC software rese • 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 rese • 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 or by an OBF flag in addition to t enable bit. 0: A host interrupt is generated when both enable bit and the corresponding OBF f 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.

					0: Disables HIRQ14 interrupt request by IF
					1: Enables HIRQ14 interrupt request
5	IRQ13E	0	R/W	—	Host IRQ13 Interrupt Enable
					0: Disables HIRQ13 interrupt request by IF
					1: Enables HIRQ13 interrupt request
4	IRQ8	0	R/W	—	Host IRQ8 Interrupt Enable
					0: Disables HIRQ8 interrupt request by IRQ
					1: Enables HIRQ8 interrupt request
3	IRQ7	0	R/W	—	Host IRQ7 Interrupt Enable
					0: Disables HIRQ7 interrupt request by IRQ
					1: Enables HIRQ7 interrupt request
2	IRQ5	0	R/W	—	Host IRQ5 Interrupt Enable
					0: Disables HIRQ5 interrupt request by IRQ
					1: Enables HIRQ5 interrupt request
1	IRQ4	1	R/W	—	Host IRQ4 Interrupt Enable
					0: Disables HIRQ4 interrupt request by IRQ
					1: Enables HIRQ4 interrupt request
0	IRQ3	1	R/W	—	Host IRQ3 Interrupt Enable
					0: Disables HIRQ3 interrupt request by IRQ
					1: Enables HIRQ3 interrupt request

3	SELIRQ7	0	R/W	—	0: [When host interrupt request is cleared] SERIRQ pin output is in the Hi-Z state. [When host interrupt request is set] SERIRQ pin output is low. 1: [When host interrupt request is cleared] SERIRQ pin output is low. [When host interrupt request is set] SERIRQ pin output is in the Hi-Z state.
2	SELIRQ5	0	R/W	—	
1	SELIRQ4	0	R/W	—	
0	SELIRQ3	0	R/W	—	

					details of STR3, see section 16.5.9, Status 1 to 3 (STR1 to STR3). 0: Bits 7 to 4 in STR3 indicate processing s the LPC interface. 1: [When TWRE = 1] Bits 7 to 4 in STR3 indicate processing s the LPC interface. [When TWRE = 0] Bits 7 to 4 in STR3 are readable/writabl which user can use as necessary.
6	SELIRQ11	0	R/W	—	Host IRQ Interrupt Select These bits select the state of the output on SERIRQ pin. 0: [When host interrupt request is cleared] SERIRQ pin output is in the Hi-Z state. [When host interrupt request is set] SERIRQ pin output is low. 1: [When host interrupt request is cleared] SERIRQ pin output is low. [When host interrupt request is set] SERIRQ pin output is in the Hi-Z state.
5	SELIRQ10	0	R/W	—	
4	SELIRQ9	0	R/W	—	
3	SELIRQ6	0	R/W	—	
2	SELSMI	0	R/W	—	
1	SELIRQ12	1	R/W	—	
0	SELIRQ1	1	R/W	—	

					the host read transfer. 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.

16.3.17 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.

16.3.18 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 SM detected.

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

4	HDTWIE	0	R/W	—	Transfer Data Transmission End Interrupt E Enables or disables HDTWI interrupt that is interrupt source to the slave. 0: Disables transfer data transmission end i 1: Enables transfer data transmission end i
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 E Enables or disables CTLWI interrupt that is interrupt source to the slave. 0: Disables control code transmission end i 1: Enables control code transmission end i
0	BUSYIE	0	R/W	—	Transfer Start Interrupt Enable Enables or disables BUSYI interrupt that is interrupt source to the slave. 0: Disables transfer start interrupt. 1: Enables transfer start interrupt.

					This status flag indicates that host writes tr 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 reac 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	FSEL1	0	R/W	—	These bits select either FIFO during BT tran
5	FSEL0	0	R/W	—	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 which IBFI3 interrupt source to the slave. 0: FIFO read request interrupt is disabled. 1: FIFO read request interrupt is enabled.
3	HRDIE	0	R/W	—	BT Host Read Interrupt Enable Enables or disables the HRDI interrupt 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 to 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.

16.3.24 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 to 1. 1: Indicates that the BTDTR buffer is being [Setting condition] When the host writes a 1 while H_BUSY is to 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: An 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.

16.3.27 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.

Bit	Bit Name	Initial Value	R/W Slave	R/W 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 enables 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.

16.3.29 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 writter 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.

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. Set the enable bit (LPC3E to LPC1E) for the channel to be used. Also set SCIFE if th
   to be used.
5. Set the selection bits for other functions (SDWNE, IEDIR).
6. As a precaution, clear the interrupt flags (LRST, SDWN, ABRT, and OBF). Read IDI
   TWR15 to clear IBF.
7. Set receive complete interrupt enable bits (IBFIE3 to IBFIE1, and ERRIE) as necessa

16.4.2 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 (a
registers and flags are not changed.

The timing of the $\overline{\text{LFRAME}}$, LCLK, and LAD signals is shown in figures 16.2 and 19.3

Table 16.5 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 (3**
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 16.2 Typical LFRAME Timing

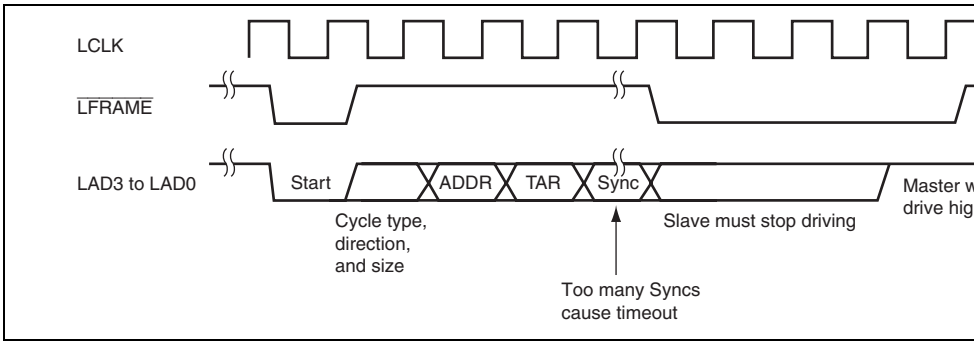


Figure 16.3 Abort Mechanism

16.4.3 SMIC Mode Transfer Flow

Figure 16.4 shows the write transfer flow and figure 16.5 shows the read transfer flow in
mode.

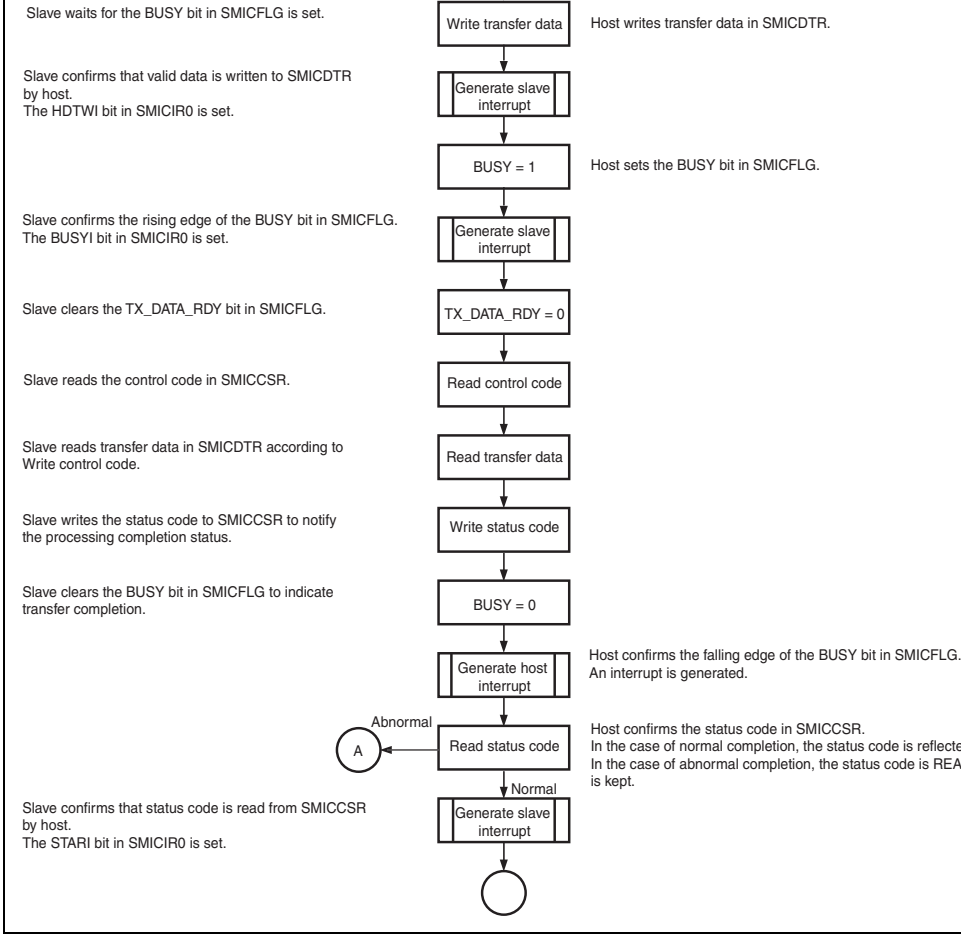


Figure 16.4 SMIC Write Transfer Flow

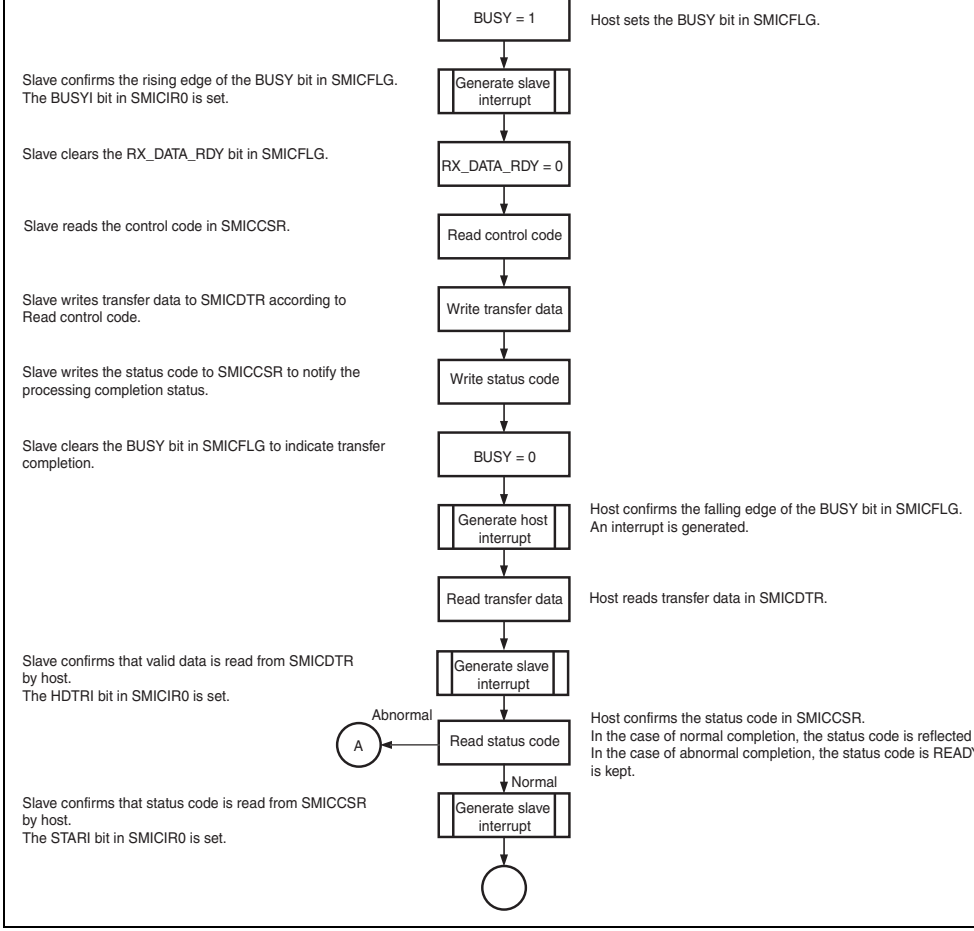


Figure 16.5 SMIC Read Transfer Flow

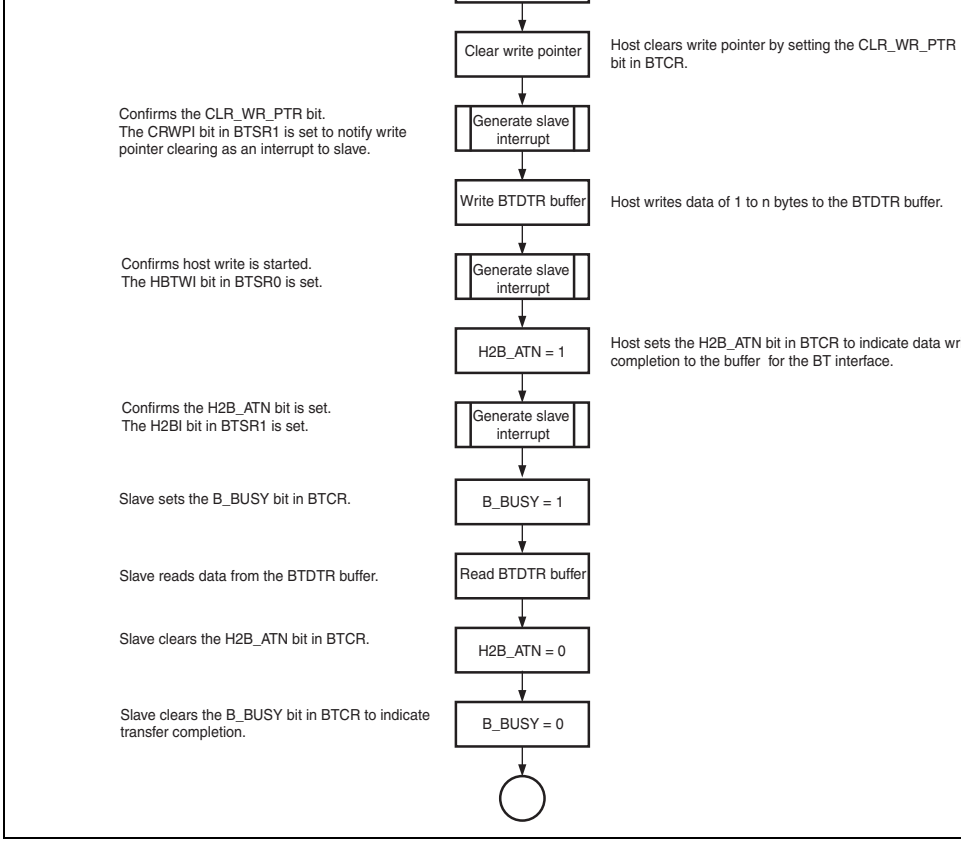


Figure 16.6 BT Write Transfer Flow

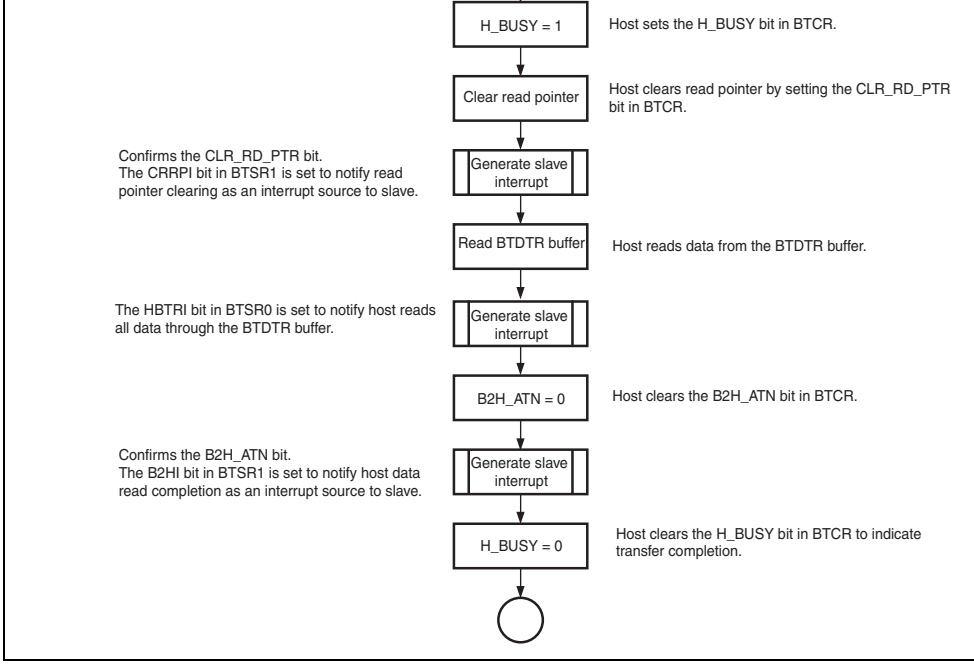


Figure 16.7 BT Read Transfer Flow

1. System reset (reset by $\overline{\text{RES}}$ pin input, or WDT0 overflow)
   All register bits, including bits LPC3E to LPC1E, are initialized.
2. LPC hardware reset (reset by $\overline{\text{LRESET}}$ pin input)
   LRSTB, SDWNE, and SDWNB bits are cleared to 0.
3. LPC software reset (reset by LRSTB)
   SDWNE and SDWNB bits are cleared to 0.
4. LPC software shutdown

The scope of the initialization in each mode is shown in table 16.9.

Host interrupt enable bits (IRQ1E1, IRQ12E1, SMIE2, IRQ6E2, IRQ9E2 to IRQ11E2, SMIE3B, SMIE3A, IRQ6E3, IRQ9E3 to IRQ11E3, SELREQ, 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, LADR1 to LADR3, IBFIE1 to IBFIE3, SELSTR3, SELIRQ1, SELSMI, SELIRQ3 to SELIRQ15, HICR4, HICR5, HISEL, BTCSR0, BTCSR1)	Initialized	Retained	Re
$\overline{\text{LRESET}}$ signal	Input (port function	Input	In
LAD3 to LAD0, $\overline{\text{LFRAME}}$, LCLK, SERIRQ, $\overline{\text{CLKRUN}}$ signals		Input	Hi

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 software shutdown (SS)

Figure 16.8 Power-Down State Termination Timing

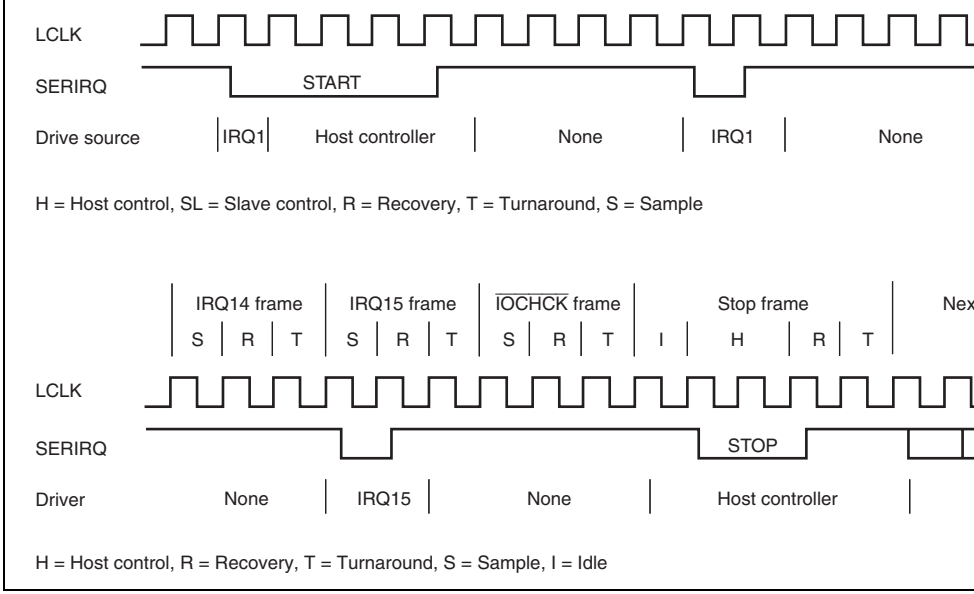


Figure 16.9 SERIRQ Timing

		Host		first state, then next 3 states 0-driven
1	IRQ0	Slave	3	Drive impossible
2	IRQ1	Slave	3	Drive possible in LPC channel 1
3	SMI	Slave	3	Drive possible in LPC channels 2 an
4	IRQ3	Slave	3	Drive possible by IRQ3E
5	IRQ4	Slave	3	Drive possible by IRQ4E
6	IRQ5	Slave	3	Drive possible by IRQ5E
7	IRQ6	Slave	3	Drive possible in LPC channels 2 an
8	IRQ7	Slave	3	Drive possible in SCIF or by IRQ7E
9	IRQ8	Slave	3	Drive possible by IRQ8E
10	IRQ9	Slave	3	Drive possible in LPC channels 2 an
11	IRQ10	Slave	3	Drive possible in LPC channels 2 an
12	IRQ11	Slave	3	Drive possible in LPC channels 2 an
13	IRQ12	Slave	3	Drive possible in LPC channel 1
14	IRQ13	Slave	3	Drive possible by IRQ13E
15	IRQ14	Slave	3	Drive possible by IRQ14E
16	IRQ15	Slave	3	Drive possible 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

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.

Table 16.9 summarizes the methods of setting and clearing these bits when the LPC chan used. Figure 16.10 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

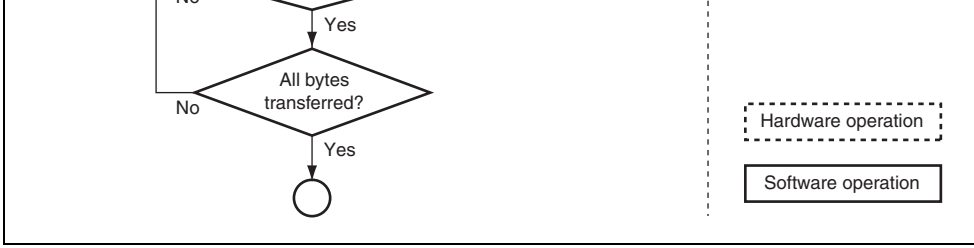


Figure 16.10 HIRQ Flowchart (Example of Channel 1)

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 16.10 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: 6.4 μs per channel (at 25-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

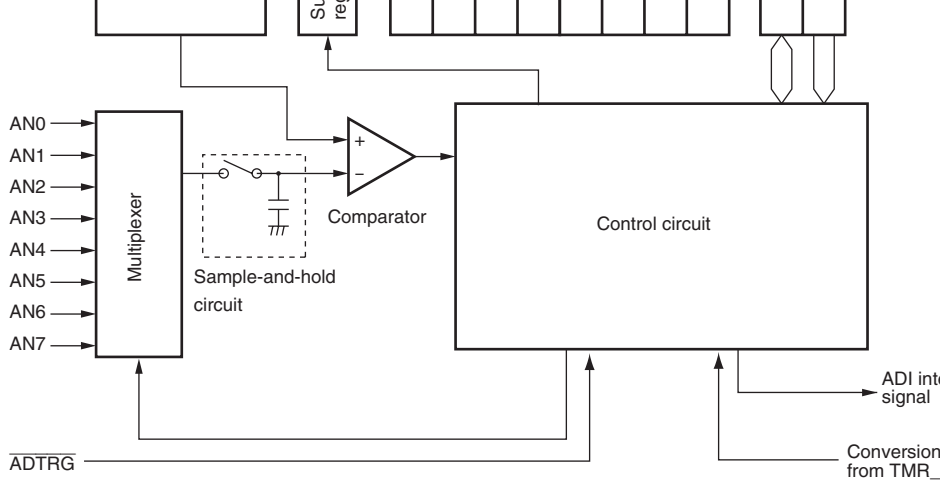
  Conversion results are held in a 16-bit data register for each channel
- Sample and hold function
- Three ways of conversion start

  Conversion start trigger from TMR_0

  Software

  External trigger signal
- Interrupt request

  A/D conversion end interrupt (ADI) request can be generated
- Module stop mode can be set



[Legend]

ADCR: A/D control register	ADDRD: A/D data register D
ADCSR: A/D control/status register	ADDRE: A/D data register E
ADDRA: A/D data register A	ADDRF: A/D data register F
ADDRB: A/D data register B	ADDRG: A/D data register G
ADDRC: A/D data register C	ADDRH: A/D data register H

Figure 17.1 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	
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)

The results of A/D conversion are stored in each registers, when the ADF flag is set to 1

Table 17.2 Analog Input Channels and Corresponding ADDR Registers

Analog Input Channel	A/D Data Register to Store A/D Conversion
AN0	ADDRA
AN1	ADDRB
AN2	ADDRC
AN3	ADDRD
AN4	ADDRE
AN5	ADDRF
AN6	ADDRG
AN7	ADDRH

				[Setting conditions] • When A/D conversion ends in single mode • When A/D conversion ends on all channels sp in scan mode [Clearing conditions] • When 0 is written after reading ADF = 1 • When DTC starts by an ADI interrupt and ADI read
6	ADIE	0	R/W	A/D Interrupt Enable Enables ADI interrupt by ADF when this bit is set
5	ADST	0	R/W	A/D Start Clearing this bit to 0 stops A/D conversion and en idle state. Setting this bit to 1 starts A/D conversio single mode, this bit is cleared to 0 automatically conversion on the specified channel ends. In scan conversion continues sequentially on the specifie channels until this bit is cleared to 0 by software, or a transition to the hardware standby mode.
4	—	0	R	Reserved This is a read-only bit and cannot be modified.

	001: AN1	001: AN0 and AN1	001: AN1 an
	010: AN2	010: AN0 to AN2	010: AN0 to
	011: AN3	011: AN0 to AN3	011: AN0 to
	100: AN4	100: AN4	100: AN0 to
	101: AN5	101: AN4 and AN5	101: AN0 to
	110: AN6	110: AN4 to AN6	110: AN0 to
	111: AN7	111: AN4 to AN7	111: AN0 to

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

[Legend] x: Don't care

				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 to 11: Scan mode (consecutive A/D conversion of channels 1 to
3	CKS1	0	R/W	Clock Select 1 and 0
2	CKS0	0	R/W	Set the A/D conversion time. Setting should be m while the conversion is stopped (ADST = 0). 00: Setting prohibited 01: Conversion time = 80 states (max) 10: Conversion time = 160 states (max) 11: Setting prohibited
1	ADSTCLR	0	R/W	A/D Start Clear Sets the automatic clearing of the ADST bit in sca 0: Disables the automatic clearing of the ADST bi mode. 1: Automatically clears the bit when A/D conversi of the selected channels are completed.

[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
   by software or an external trigger input.
2. When A/D conversion is completed, the result is transferred to the A/D data register
   corresponding to the channel.
3. On completion of A/D conversion, the ADF bit in ADCSR is set to 1. If the ADIE bi
   1 at this time, an ADI interrupt request is generated.
4. The ADST bit remains set to 1 during A/D conversion. When conversion ends, the A
   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 id

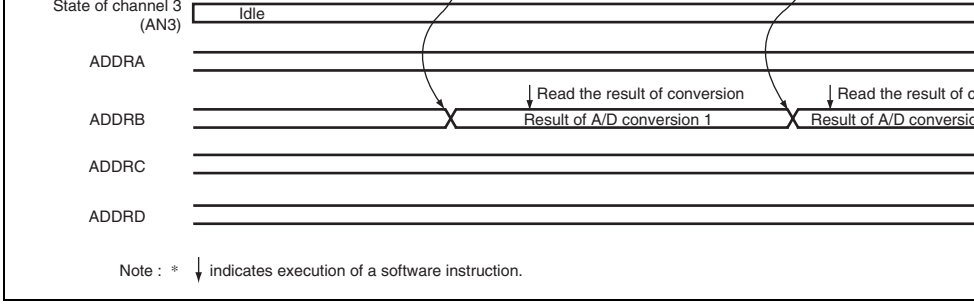
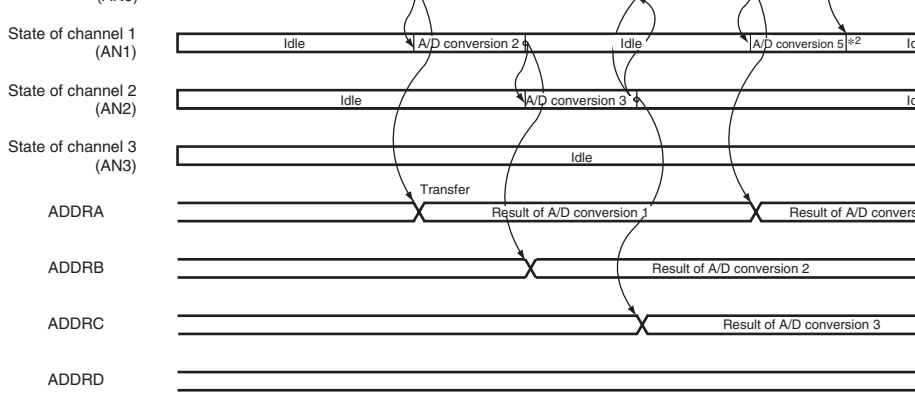


Figure 17.2 Example of A/D Converter Operation (When Channel 1 is Selected in Single Mode)

17.4.2 Scan Mode

In scan mode, A/D conversion is performed sequentially on the specified channels (four c
or eight channel maximum). Operations are as follows.

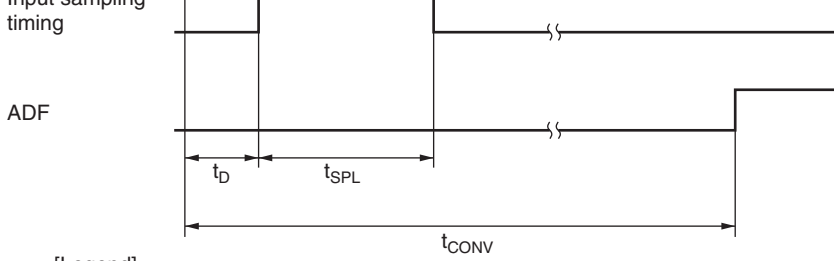
1. When the ADST bit in ADCSR is set to 1 by software or an external trigger input, A/
   conversion starts from the first channel of the selected channel. Consecutive A/D con
   either four channels maximum (SCANE and SCANS = B'10) or eight channels maxin
   (SCANE and SCANS = B'11) can be selected. In the case of consecutive A/D convers
   four channels, the operation starts from AN0 when CH2 = B'0, and starts from AN4 w
   CH2 = B'1. In the case of consecutive A/D conversion on eight channels, the operatio
   from AN0.
2. When A/D conversion for each channel is completed, the result is sequentially transfe
   the A/D data register corresponding to each channel.
3. When conversion of all the selected channels is completed, the ADF bit in ADCSR is
   If the ADIE bit is set to 1 at this time, an ADI interrupt is requested after A/D convers
   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 17.3 Example of A/D Converter Operation (When Channels AN0 to AN3 are Selected in Scan Mode)

In scan mode, the values given in table 17.3 apply to the first conversion time. In the secc
subsequent conversions, the conversion time is as shown in table 17.4. In either case, set t
and CKS0 bits in ADCR so that the conversion time falls within the range of A/D conver
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 17.4 A/D Conversion Timing

Table 17.4 A/D Conversion Time (Scan Mode)

CKS1	CKS0	Conversion Time (States)
0	0	Setting prohibited
	1	80 (fixed)
0	0	160 (fixed)
	1	Setting prohibited

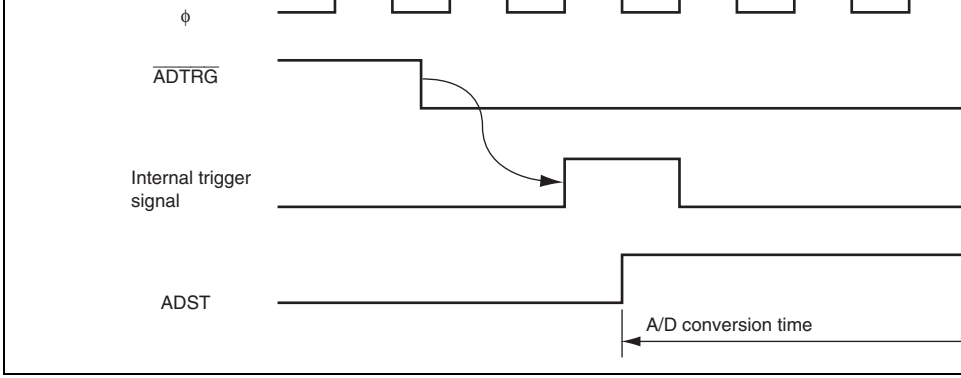


Figure 17.5 Timing of External Trigger Input

17.5 Interrupt Source

The A/D converter generates an A/D conversion end interrupt (ADI) at the end of A/D c
Setting the ADIE bit to 1 enables ADI interrupt requests while the ADF bit in ADCSR i
after A/D conversion ends. The ADI interrupt can be used to activate the DTC. Reading
converted data by the DTC activated by the ADI interrupt allows consecutive conversion
performed without software overhead.

Table 17.5 A/D Converter Interrupt Source

Name	Interrupt Source	Interrupt Flag	DTC Activati
ADI	A/D conversion end	ADF	Possible

when the digital output changes from the minimum voltage value B'00 0000 0000 (H'
B'00 0000 0001 (H'001) (see figure 17.7).

- Full-scale error

  The deviation of the analog input voltage value from the ideal A/D conversion charac
  when the digital output changes from B'11 1111 1110 (H'3FE) to B'11 1111 1111 (H'
  figure 17.7).

- Nonlinearity error

  The error with respect to the ideal A/D conversion characteristics between the zero vo
  the full-scale voltage. Does not include the offset error, full-scale error, or quantizatio
  (see figure 17.7).

- Absolute accuracy

  The deviation between the digital value and the analog input value. Includes the offse
  full-scale error, quantization error, and nonlinearity error.

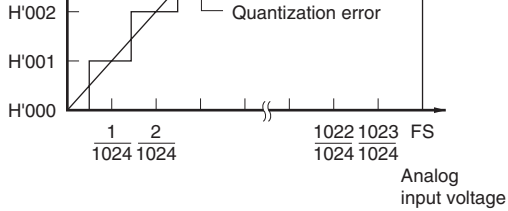


Figure 17.6 A/D Conversion Accuracy Definitions

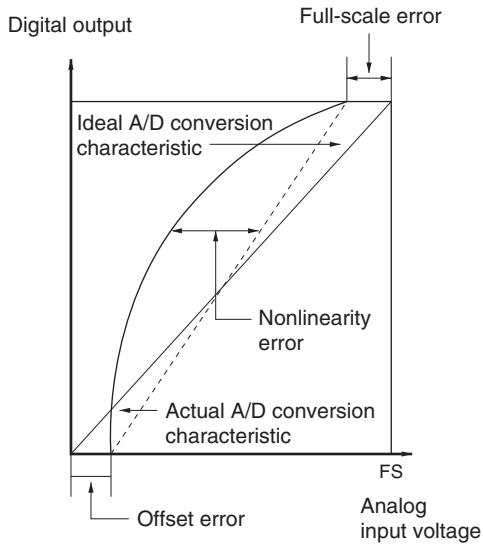


Figure 17.7 A/D Conversion Accuracy Definitions

This LSI's analog input is designed so that the conversion accuracy is guaranteed for an i
signal for which the signal source impedance is 5 kΩ or less. This specification is provide
enable the A/D converter's sample-and-hold circuit input capacitance to be charged withi
sampling time; if the sensor output impedance exceeds 5 kΩ, charging may be insufficien
may not be possible to guarantee the A/D conversion accuracy. However, if a large capac
provided externally in single mode, the input load will essentially comprise only the inter
resistance of 10 kΩ, and the signal source impedance is ignored. However, since a low-pa
effect is obtained in this case, it may not be possible to follow an analog signal with a larg
differential coefficient (voltage fluctuation ratio of 5 mV/μs or greater for example) (see f
17.8). When converting a high-speed analog signal or converting in scan mode, a low-imp
buffer should be inserted.

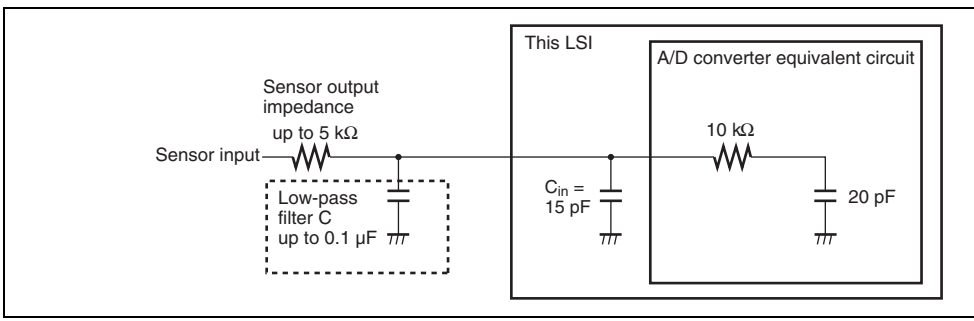


Figure 17.8 Example of Analog Input Circuit

If conditions shown below are not met, the reliability of this LSI may be adversely affec

- Analog input voltage range

  The voltage applied to analog input pin ANn during A/D conversion should be in the
  AVss ≤ V_{AN} ≤ AVref.
- Relation between AVcc, AVss and Vcc, Vss

  The relationship between AVcc, AVss and Vcc, Vss should be Avcc = Vcc ± 0.3V a
  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.

17.7.5 Notes on Board Design

In board design, digital circuitry and analog circuitry should be as mutually isolated as p
and layout in which digital circuit signal lines and analog circuit signal lines cross or are
proximity should be avoided as far as possible. Failure to do so may result in incorrect o
of the analog circuitry due to inductance, adversely affecting A/D conversion values. Al
circuitry must be isolated from the analog input signals (AN0 to AN7), the analog refere
voltage (AVref) and analog power supply (AVcc) by the analog ground (AVss). Also, th
ground (AVss) should be connected at one point to a stable digital ground (Vss) on the b

the analog input pin voltage. Therefore, careful consideration is required upon deciding th
constants.

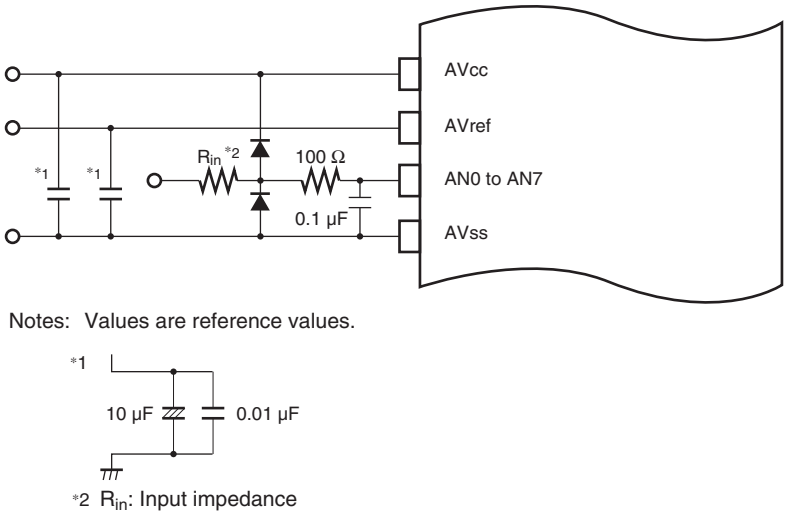


Figure 17.9 Example of Analog Input Protection Circuit

Note: Values are reference values.

Figure 17.10 Analog Input Pin Equivalent Circuit

17.7.7 Note on the Usage in Software Standby Mode

If this LSI enters software standby mode with the A/D conversion enabled, the content o
converter is retained and about the same amount of analog supply current may flow as th
when A/D conversion in progress. If the analog supply current must be reduced in softw
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 progra
  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 1000 times at the minimum
  value ranged from 1 to 1000 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

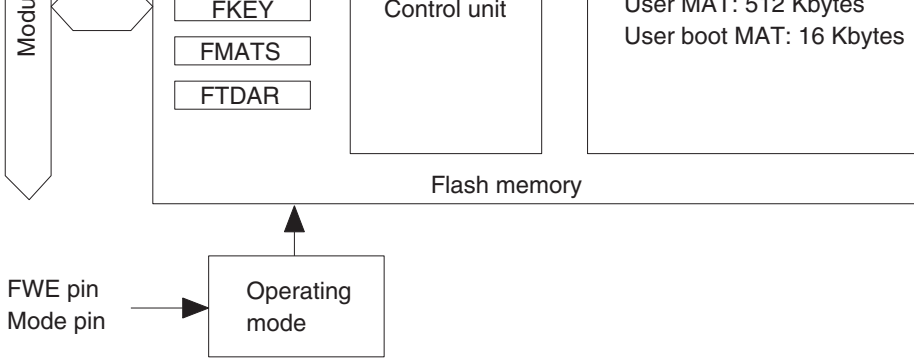
    The user boot program of the optional interface can be made and the user MAT c
    programmed.

- 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.



[Legend]
FCCS: Flash code control status register
FPCS: Flash program code select register
FECS: Flash erase code select register
FKEY: Flash key code register
FMATS: Flash MAT select register
FTDAR: Flash transfer destination address register

Note: To read from or write to the registers, the FLSHE bit in the serial timer co
register (STCR) must be set to 1.

Figure 19.1 Block Diagram of Flash Memory

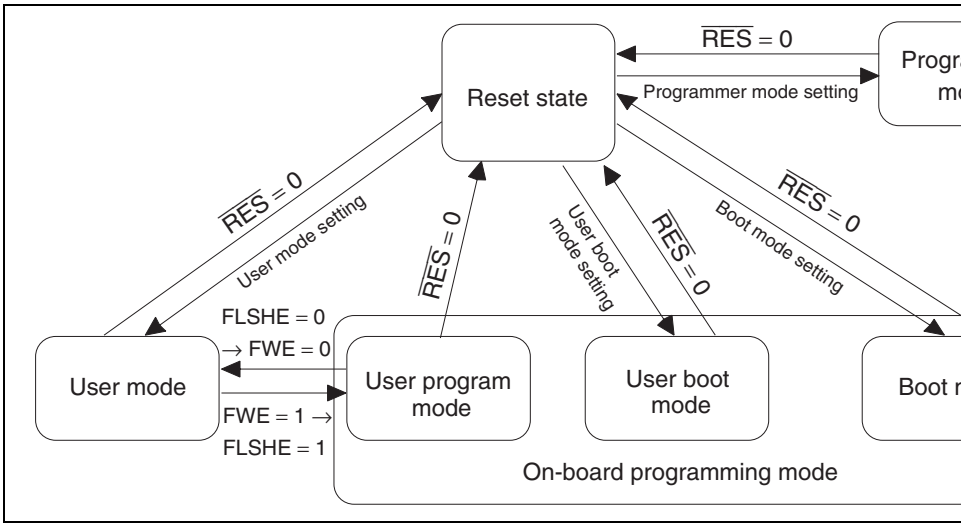


Figure 19.2 Mode Transition of Flash Memory

Programming/ erasing enable MAT	User MAT User boot MAT	User MAT	User MAT	User MAT User boo
All erasure	○ (Automatic)	○	○	○ (Autom
Block division erasure	○ *1	○	○	×
Program data transfer	From host via SCI	Via optional device	Via optional device	Via progr
Reset initiation MAT	Embedded program storage MAT	User MAT	User boot MAT*2	—
Transition to user mode	Changing mode setting and reset	Changing FLSHE bit and FWE pin	Changing mode setting and reset	—

Notes: 1. All-erasure is performed. After that, the specified block can be erased.
2. Firstly, the reset vector is fetched from the embedded program storage MAT. A
flash memory related registers are checked, the reset vector is fetched from th
boot MAT.

- The user boot MAT can be programmed or erased only in boot mode and programme
- The user MAT and user boot MAT are erased in boot mode. Then, the user MAT and
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.

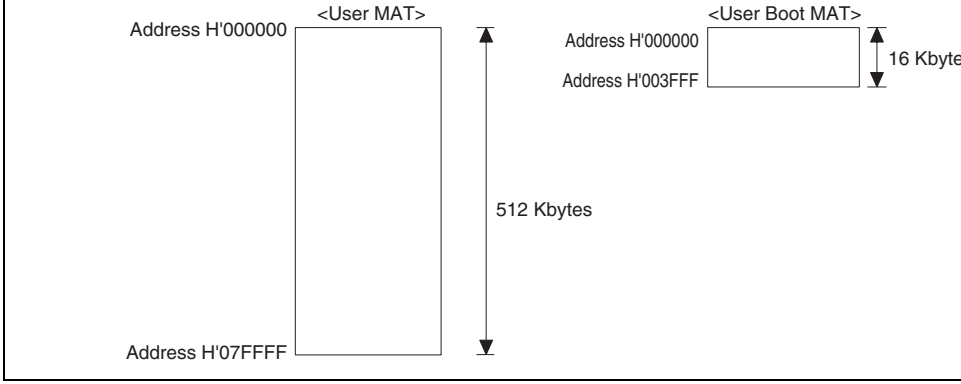


Figure 19.3 Flash Memory Configuration

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.

19.1.4 Block Division

The user MAT is divided into seven 64-Kbyte blocks, one 32-Kbyte block, and eight 4-K blocks as shown in figure 19.4. The user MAT can be erased in this divided-block units, erase-block number of EB0 to EB15 is specified when erasing. Programming is perform byte units starting at the addresses whose lowest-order byte is H'00 or H'80.

EB5 Erase unit: 4 kbytes	H'005000	H'005001	H'005002	←Programming unit: 128 bytes→	H'00507F
	H'005F80	H'005F81	H'005F82	– – – – – – – – – – – – – – –	H'005FFF
EB6 Erase unit: 4 kbytes	H'006000	H'006001	H'006002	←Programming unit: 128 bytes→	H'00607F
	H'006F80	H'006F81	H'006F82	– – – – – – – – – – – – – – –	H'006FFF
EB7 Erase unit: 4 kbytes	H'007000	H'007001	H'007002	←Programming unit: 128 bytes→	H'00707F
	H'007F80	H'007F81	H'007F82	– – – – – – – – – – – – – – –	H'007FFF
EB8 Erase unit: 32 kbytes	H'008000	H'008001	H'008002	←Programming unit: 128 bytes→	H'00807F
	H'00FF80	H'00FF81	H'00FF82	– – – – – – – – – – – – – – –	H'00FFFF
EB9 Erase unit: 64 kbytes	H'010000	H'010001	H'010002	←Programming unit: 128 bytes→	H'01007F
	H'01FF80	H'01FF81	H'01FF82	– – – – – – – – – – – – – – –	H'01FFFF
EB10 Erase unit: 64 kbytes	H'020000	H'020001	H'020002	←Programming unit: 128 bytes→	H'02007F
	H'02FF80	H'02FF81	H'02FF82	– – – – – – – – – – – – – – –	H'02FFFF
EB11 Erase unit: 64 kbytes	H'030000	H'030001	H'030002	←Programming unit: 128 bytes→	H'03007F
	H'03FF80	H'03FF81	H'03FF82	– – – – – – – – – – – – – – –	H'03FFFF
EB12 Erase unit: 64 kbytes	H'040000	H'04F001	H'04F002	←Programming unit: 128 bytes→	H'04F07F
	H'04FF80	H'04FF81	H'04FF82	– – – – – – – – – – – – – – –	H'04FFFF
EB13 Erase unit: 64 kbytes	H'050000	H'050001	H'050002	←Programming unit: 128 bytes→	H'05007F
	H'05FF80	H'05FF81	H'05FF82	– – – – – – – – – – – – – – –	H'05FFFF
EB14 Erase unit: 64 kbytes	H'060000	H'060001	H'060002	←Programming unit: 128 bytes→	H'06007F
	H'06FF80	H'06FF81	H'06FF82	– – – – – – – – – – – – – – –	H'06FFFF
EB15 Erase unit: 64 kbytes	H'070000	H'070001	H'070002	←Programming unit: 128 bytes→	H'07007F
	H'07FF80	H'07FF81	H'07FF82	– – – – – – – – – – – – – – –	H'07FFFF

Figure 19.4 Block Division of User MAT

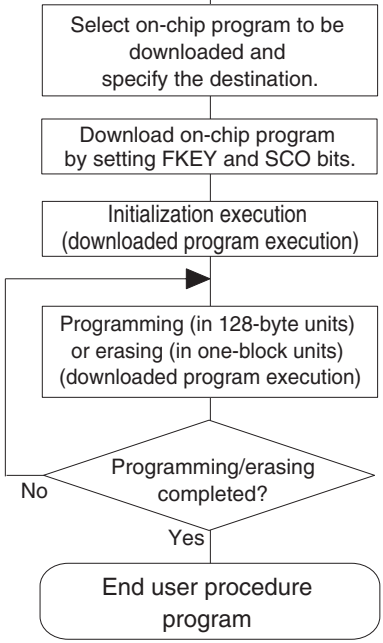


Figure 19.5 Overview of User Procedure Program

1. Selection of on-chip program to be downloaded

   For programming/erasing execution, the FLSHE bit in STCR must be set to 1 to tran
   user program mode.

   This LSI has programming/erasing programs which can be downloaded to the on-ch
   The on-chip program to be downloaded is selected by setting the corresponding bits
   programming/erasing interface register. The address of the programming destination
   specified by the flash transfer destination address register (FTDAR).

3. Initialization of programming/erasing

   The operating frequency is set before execution of programming/erasing. This setting
   performed by using the programming/erasing interface parameter.

4. Programming/erasing execution

   For programming/erasing execution, the FLSHE bit in STCR and the FWE pin must b
   to transition to user program mode.

   The program data/programming destination address is specified in 128-byte units whe
   programming.

   The block to be erased is specified in erase-block units when erasing.

   These specifications are set by using the programming/erasing interface parameter an
   chip program is initiated. The on-chip program is executed by using the JSR or BSR
   instruction and performing the subroutine call of the specified address in the on-chip
   The execution result is returned to the programming/erasing interface parameter.

   The area to be programmed must be erased in advance when programming flash mem

   All interrupts are prohibited during programming and erasing. Interrupts must be mas
   within the user system.

5. When programming/erasing is executed consecutively

   When the processing is not ended by the 128-byte programming or one-block erasure
   program address/data and erase-block number must be updated and consecutive
   programming/erasing is required.

   Since the downloaded on-chip program is left in the on-chip RAM after the processin
   download and initialization are not required when the same processing is executed
   consecutively.

TxD1	Output	Serial transmit data output (used in boot mode)
RxD1	Input	Serial receive data input (used in boot mode)

19.3 Register Descriptions

The registers/parameters which control flash memory are shown in the following. To rea
write to these registers/parameters, the FLSHE bit in the serial timer control register (ST
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/
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 ar 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 mem
(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 mem
(error protection) is valid.

[Setting conditions]

- When an interrupt, such as NMI, occurs duri
programming/erasing flash memory.
- When the flash memory is read during
programming/erasing flash memory (includin
vector read or an instruction fetch).
- When the SLEEP instruction is executed du
programming/erasing flash memory (includin
software-standby mode)
- When a bus master other than the CPU, suc
DTC, gets bus mastership during
programming/erasing flash memory.

				The interrupt exception handling on and after v number 32 should not be used because the co vector is not read, resulting in the CPU runawa 0: The space for the interrupt vector table is no When interrupt vector data is not read succe the operation for the interrupt exception han cannot be guaranteed. An occurrence of an should be masked. 1: The space for the interrupt vector table is m Even when interrupt vector data is not read successfully, the interrupt exception handlin 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/erasin program to the on-chip RAM is not executed.

[Clearing condition]
When download is completed

1: Request that the on-chip programming/erasin program is downloaded to the on-chip RAM i occurred.

[Setting conditions]
When all of the following conditions are satisfie 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 complete
				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 complete 1: On-chip erasing program is selected.

4	K4	0	R/W	cannot be set to the SCO bit. Therefore downlo the on-chip RAM cannot be executed.
3	K3	0	R/W	Only when H'5A is written, programming/erasing
2	K2	0	R/W	executed. Even if the on-chip programming/eras
1	K1	0	R/W	program is executed, the flash memory cannot b
0	K0	0	R/W	programmed or erased when the value other tha is written to FKEY.
				H'A5: Writing to the SCO bit is enabled. (The SC cannot be set by the value other than H'A
				H'5A: Programming/erasing is enabled. (The va than H'A5 is in software protection state.)
				H'00: Initial value

2	MS2	0	R/W	Switching between User MAT and User Boot M user boot MAT cannot be programmed in user mode if user boot MAT is selected by FMATS. boot MAT must be programmed in boot mode programmer mode.) H'AA: The user boot MAT is selected (in user- selection state when the value of these other than H'AA) Initial value when these bits are initiated boot mode. H'00: Initial value when these bits are initiated except for user boot mode (in user-MAT state) [Programmable condition] These bits are in the execution state in the on-
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 p downloaded by setting the SCO bit in FCCS to sure that this bit is cleared to 0 before setting th to 1 and the value specified by TDA6 to TDA0 is 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 ov range (H'04 to H'FF) and the download is stop
6	TDA6	0	R/W	Transfer Destination Address
5	TDA5	0	R/W	Specifies the start address to download an on-c program. H'00 to H'03 can be specified as the s 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 this sets the TDER bit to 1 and stops 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 19.4. T
meaning of the bits in FPFR varies in each processing program: initialization, programm
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 sp
FTDAR

be used to determine if downloading is executed or not. Since the confirmation whether bit is set to 1 is difficult, the certain determination must be performed by writing the sin the start address specified by FTDAR to the value other than the return value of downlo 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 c downloaded can be specified. When more tha types of the program are selected, the progran selected, or the program is selected without m error is occurred. 0: Download program can be selected normall 1: Download error is occurred (multi-selection program which is not mapped is selected)
1	FK	—	R/W	Flash Key Register Error Detect Returns the check result whether the value of 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 opera
frequency in this LSI is 20 to 25 MHz.

Bit	Bit Name	Initial Value	R/W	Description
31 to 16	—	—	—	Unused This bit should be cleared to 0.
15 to 0	F15 to F0	—	R/W	Frequency Set Set the operating frequency of the CPU. With t multiplication function, set the frequency multip setting value must be calculated as the followin methods. 1. The operating frequency which is shown in must be rounded in a number to three deci and be shown in a number of two decimal p 2. The value multiplied by 100 is converted to digit and is written to the FPEFEQ paramet (general register ER0). For example, when the operating frequency of is 25.000 MHz, the value is as follows. 1. The number to three decimal places of 25.0 rounded and the value is thus 25.00. 2. The formula that 25.00 × 100 = 2500 is con the binary digit and B'0000,1001,1101,010 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 nor 0: Initialization is ended normally (no error) 1: Initialization is ended abnormally (error occur

(3) Programming Execution

When flash memory is programmed, the programming destination address on the user MA be passed to the programming program in which the program data is downloaded.

1. The start address of the programming destination on the user MAT must be stored in a register ER1. This parameter is called as flash multipurpose address area parameter (F

   Since the program data is always in units of 128 bytes, the lower eight bits (A7 to A0) H'00 or H'80 as the boundary of the programming start address on the user MAT.
2. The program data for the user MAT must be prepared in the consecutive area. The pro data must be in the consecutive space which can be accessed by using the MOV.B ins of the CPU and in other than the flash memory space.

   When data to be programmed does not satisfy 128 bytes, the 128-byte program data n prepared by filling with the dummy code H'FF.

   The start address of the area in which the prepared program data is stored must be sto general register ER0. This parameter is called as flash multipurpose data destination a parameter (FMPDR).

For details on the program processing procedure, see section 19.4.2, User Program Mode

31 to 0	MOA31 to MOA0	—	R/W	Store the start address of the programming de on the user MAT. The consecutive 128-byte programming is executed starting from the spe address of the user MAT. Therefore, the speci programming start address becomes a 128-by boundary and MOA6 to MOA0 are always 0.

(b) Flash multipurpose data destination parameter (FMPDR: general register ER CPU):

This parameter stores the start address in the area which stores the data to be programme user MAT. When the storage destination of the program data is in flash memory, an erro The error occurrence is indicated by the WD bit in FPFR.

Bit	Bit Name	Initial Value	R/W	Description
31 to 0	MOD31 to MOD0	—	R/W	Store the start address of the area which store program data for the user MAT. The consecuti byte data is programmed to the user MAT star the specified start address.

(c) Flash pass/fail parameter (FPFR: general register R0L of CPU)

This parameter indicates the return value of the program processing result.

Bit	Bit Name	Initial Value	R/W	Description
7	—	—	—	Unused Return 0.

				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 erase bit is set to 1, there is a high possibility that the is partially rewritten. In this case, after removing factor, erase the user MAT. If FMATS is set to H'AA and the user boot MAT selected, an error occurs when programming is performed. In this case, both the user MAT and MAT are not rewritten. Programming of the user MAT should be performed in boot mode or prog mode. 0: Programming has ended normally 1: Programming has ended abnormally (progr result is not guaranteed)
4	FK	—	R/W	Flash Key Register Error Detect Returns the check result of the value of FKEY b start of the programming processing. 0: FKEY setting is normal (FKEY = H'5A) 1: FKEY setting is error (FKEY = value other tha
3	—	—	—	Unused Returns 0.

				• When the programming destination addre area other than flash memory is specified • When the specified address is not in a 128 boundary. (The lower eight bits of the addr other than H'00 and H'80.) 0: Setting of programming destination address 1: Setting of programming destination address abnormal
0	SF	—	R/W	Success/Fail Indicates whether the program processing is e normally or not. 0: Programming is ended normally (no error) 1: Programming is ended abnormally (error oc

Bit	Bit Name	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 corresponds to the EB0 block, and 15 correspo the EB15 block.
14	EB14	—	R/W	
13	EB13	—	R/W	
12	EB12	—	R/W	
11	EB11	—	R/W	
10	EB10	—	R/W	
9	EB9	—	R/W	
8	EB8	—	R/W	
7	EB7	—	R/W	
6	EB6	—	R/W	
5	EB5	—	R/W	
4	EB4	—	R/W	
3	EB3	—	R/W	
2	EB2	—	R/W	
1	EB1	—	R/W	
0	EB0	—	R/W	

				entered. When the low-level signal is input to th pin or the error protection state is entered, 1 is this bit. The state can be confirmed with the FW FLER bits in FCCS. For conditions to enter the protection state, see section 19.5.3, Error Prot 0: FWE and FLER settings are normal (FWE FLER = 0) 1: Programming cannot be performed (FWE FLER = 1)
5	EE	—	R/W	Erasure Execution Error Detect 1 is returned to this bit when the user MAT cou erased or when flash-memory related register are partially changed. If this bit is set to 1, ther possibility that the user MAT is partially erasec case, after removing the error factor, erase the MAT. If FMATS is set to H'AA and the user bo selected, an error occurs when erasure is perf this case, both the user MAT and user boot MA erased. Erasing of the user boot MAT should b performed in boot mode or programmer mode. 0: Erasure has ended normally 1: Erasure has ended abnormally (erasure re guaranteed)
4	FK	—	R/W	Flash Key Register Error Detect Returns the check result of FKEY value before the erasing processing. 0: FKEY setting is normal (FKEY = H'5A) 1: FKEY setting is error (FKEY = value other th

	Indicates whether the erasing processing is enc normally or not. 0: Erasure is ended normally (no error) 1: Erasure is ended abnormally (error occurs)

19.4 On-Board Programming Mode

When the pin is set in on-board programming mode and the reset start is executed, the on programming state that can program/erase the on-chip flash memory is entered. On-board programming mode has three operating modes: boot mode, user program mode, and user mode.

For details of the pin setting for entering each mode, see table 19.5. For details of the stat transition of each mode for flash memory, see figure 19.2.

Table 19.5 Setting On-Board Programming Mode

Mode Setting	FWE	$\overline{\text{MD2}}$	MD1	NMI
Boot mode	1	0	0	1
User program mode	1*	1	1	0/1
User boot mode	1	0	0	0

Note: * Before downloading the programming/erasing programs, the FLSHE bit must b 1 to transition to user program mode.

setting in boot mode, see table 19.5. The NMI and other interrupts are ignored in boot m
However, the NMI and other interrupts should be disabled in the user system.

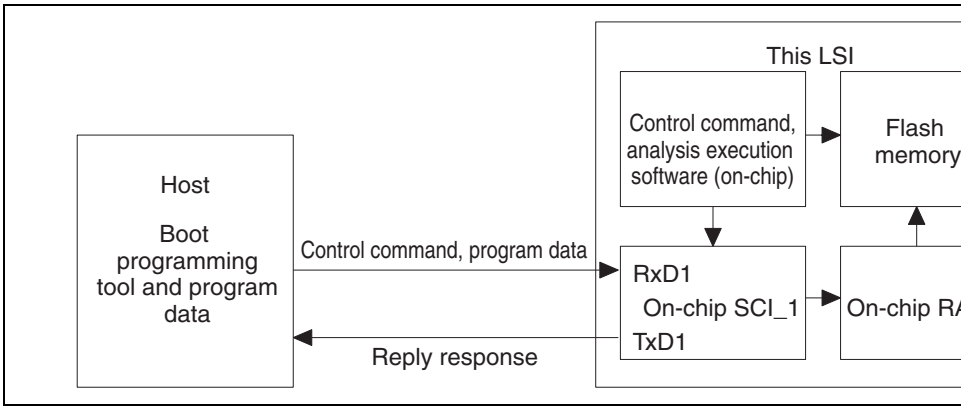


Figure 19.6 System Configuration in Boot Mode

set to 9,600 bps or 19,200 bps.

The system clock frequency, which can automatically adjust the transfer bit rate of the ho
the bit rate of this LSI, is shown in table 19.6. Boot mode must be initiated in the range o
system clock.

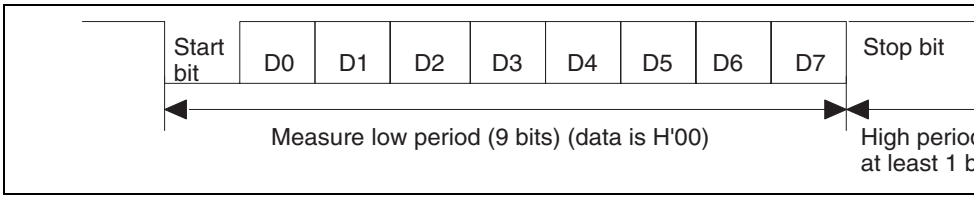


Figure 19.7 Automatic-Bit-Rate Adjustment Operation of SCI

Table 19.6 System Clock Frequency for Automatic-Bit-Rate Adjustment by This L

Bit Rate of Host	System Clock Frequency
9,600 bps	20 to 25 MHz
19,200 bps	

After inquiries have finished, all user MAT and user boot MAT are automatically er

4. Waiting for programming/erasing command
   - When the program preparation notice is received, the state for waiting program d entered. The programming start address and program data must be transmitted fo the programming command. When programming is finished, the programming st must be set to H'FFFFFFFF and transmitted. Then the state for waiting program returned to the state of programming/erasing command wait.
   - When the erasure preparation notice is received, the state for waiting erase-block 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 tra Then the state for waiting erase-block data is returned to the state for waiting programming/erasing command. The erasure must be used when the specified bl programmed without a reset start after programming is executed in boot mode. W programming can be executed by only one operation, all blocks are erased before for waiting programming/erasing/other command is entered. The erasing operatio required.
   - There are many commands other than programming/erasing. Examples are sum c blank check (erasure check), and memory read of the user MAT/user boot MAT acquisition of current status information.

Note that memory read of the user MAT/user boot MAT can only read the programmed all user MAT/user boot MAT has automatically been erased.

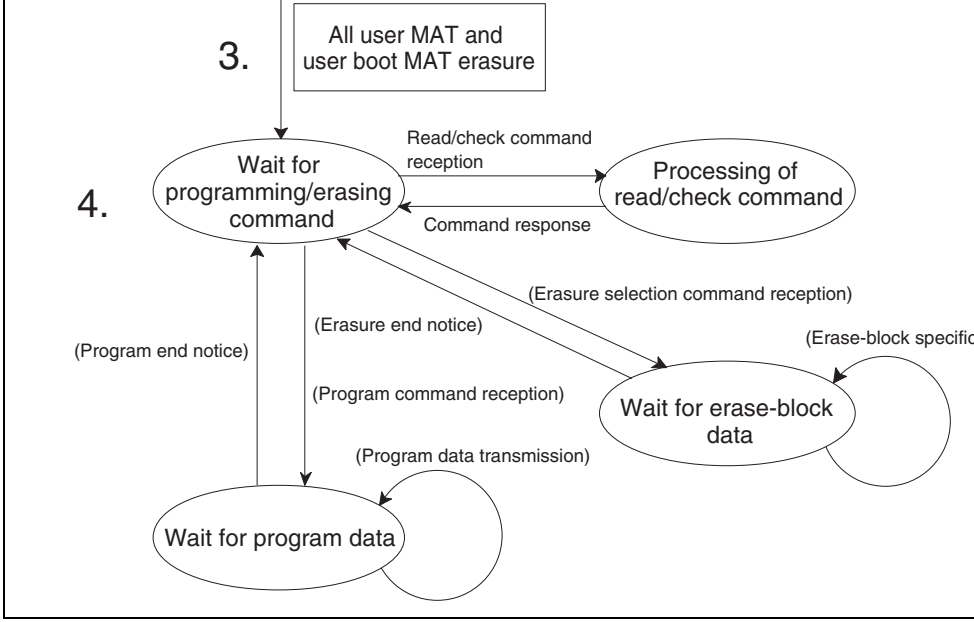


Figure 19.8 Overview of Boot Mode State Transition Diagram

period of 100 μs which is longer than normal.

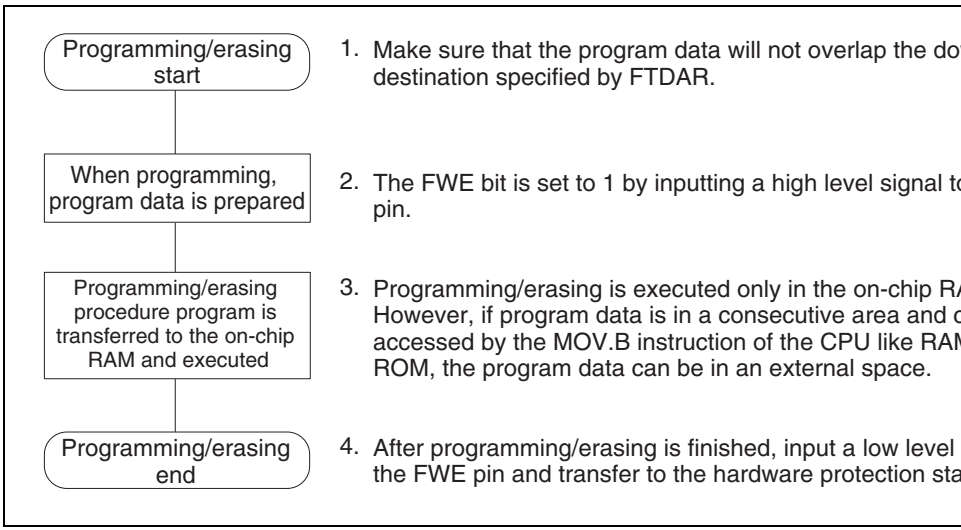


Figure 19.9 Programming/Erasing Overview Flow

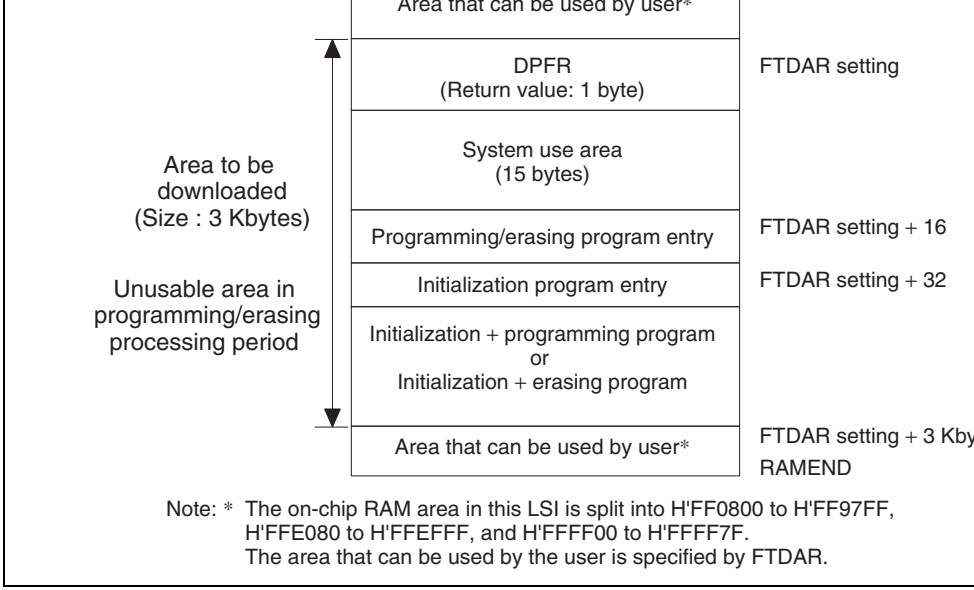


Figure 19.10 RAM Map When Programming/Erasing is Executed

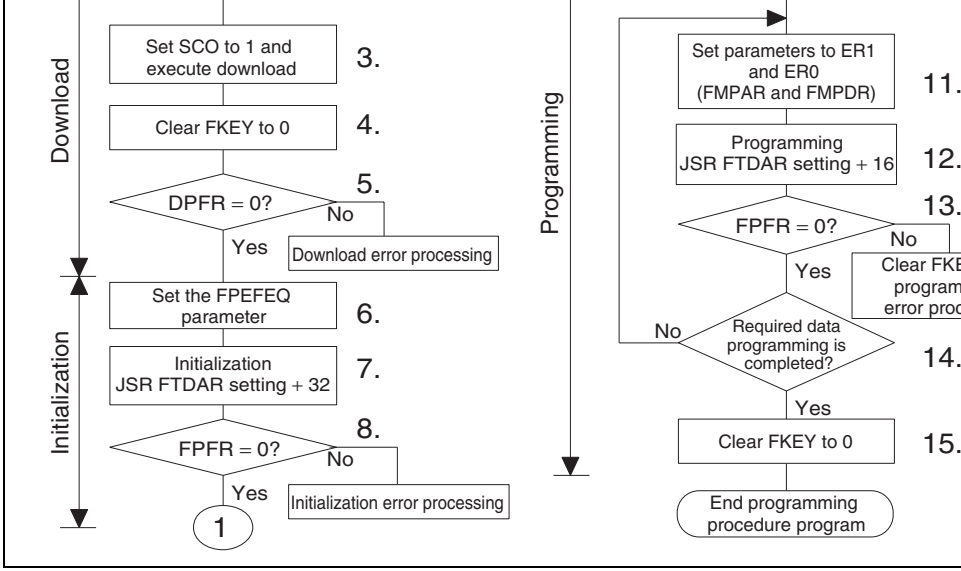


Figure 19.11 Programming Procedure

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 19.4.4, Procedure Program and Storable A Programming Data.

The following description assumes the area to be programmed on the user MAT is erase program data is prepared in the consecutive area. When erasing is not executed, erasing executed before writing.

download is not performed and a download error is returned to the SS bit in DPFR. Th
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 down
   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 is
   to the user procedure program, the SCO is cleared to 0. Therefore, the SCO bit cannot
   confirmed to be 1 in the user procedure program.

   The download result can be confirmed only by the return value of DPFR. Before the S
   is set to 1, incorrect determination must be prevented by setting the one byte of the sta
   address (to be used as DPFR) specified by FTDAR to a value other than the return val
   H'FF).

   When download is executed, particular interrupt processing, which is accompanied by
   switch as described below, is performed as an internal microcomputer processing. Fou
   instructions are executed immediately after the instructions that set the SCO bit to 1.

   — The user-MAT space is switched to the on-chip program storage area.

   — After the selection condition of the download program and the FTDAR setting are
     the transfer processing to the on-chip RAM specified by FTDAR is executed.

   — 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 allocate
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 b
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 downloa
destination specified by FTDAR). If the value is H'00, download has been perfor
normally. If the value is not H'00, the source that caused download to fail can be
investigated by the description below.

— If the value of the DPFR parameter is the same as before downloading (e.g. H'FF
address setting of the download destination in FTDAR may be abnormal. In this
confirm the setting of the TDER bit (bit 7) in FTDAR.

— If the value of the DPFR parameter is different from before downloading, check
(bit 2) and the FK bit (bit 1) in the DPFR parameter to ensure that the download
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 (g
register ER0).

The settable range of the FPEFEQ parameter is 20 to 25 MHz. When the frequen
out of this range, an error is returned to the FPFR parameter of the initialization p
and initialization is not performed. For details on the frequency setting, see the d
in 19.3.2 (2) (a), Flash programming/erasing frequency parameter (FPEFEQ).

— R0L is a return value of the FPFR parameter.

— Since the stack area is used in the initialization program, 128-byte stack area at the 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 de

8. The return value in the initialization program, FPFR (general register R0L) is determi

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 time voltage for more than the specified time will be applied and flash memory may be dar Therefore, interrupts and bus mastership to other than the CPU, such as to the DTC, a prohibited.

   To disable interrupts, bit 7 (I) in the condition code register (CCR) of the CPU should B'1 in interrupt control mode 0 or bits 7 and 6 (I and UI) should be set to B'11 in inter 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 err 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 pr
execution routine is executed, programming is not executed and an error is return
FPFR parameter. In this case, the program data must be transferred to the on-chi
and then programming must be executed.

12. Programming

There is an entry point of the programming program in the area from the start addres
by FTDAR + 16 bytes of the on-chip RAM. The subroutine is called and programmi
executed by using the following steps.

```
MOV.L   #DLTOP+16,ER2        ; Set entry address to ER2
JSR     @ER2                 ; Call programming routine
NOP
```

— The general registers other than R0L are held in the programming program.

— R0L is a return value of the FPFR parameter.

— Since the stack area is used in the programming program, a stack area of 128 byt
maximum must be allocated in RAM.

13. The return value in the programming program, FPFR (general register R0L) is deter

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
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 al
programmed is written to again, not only will a programming error occur, but also fl
memory will be damaged.

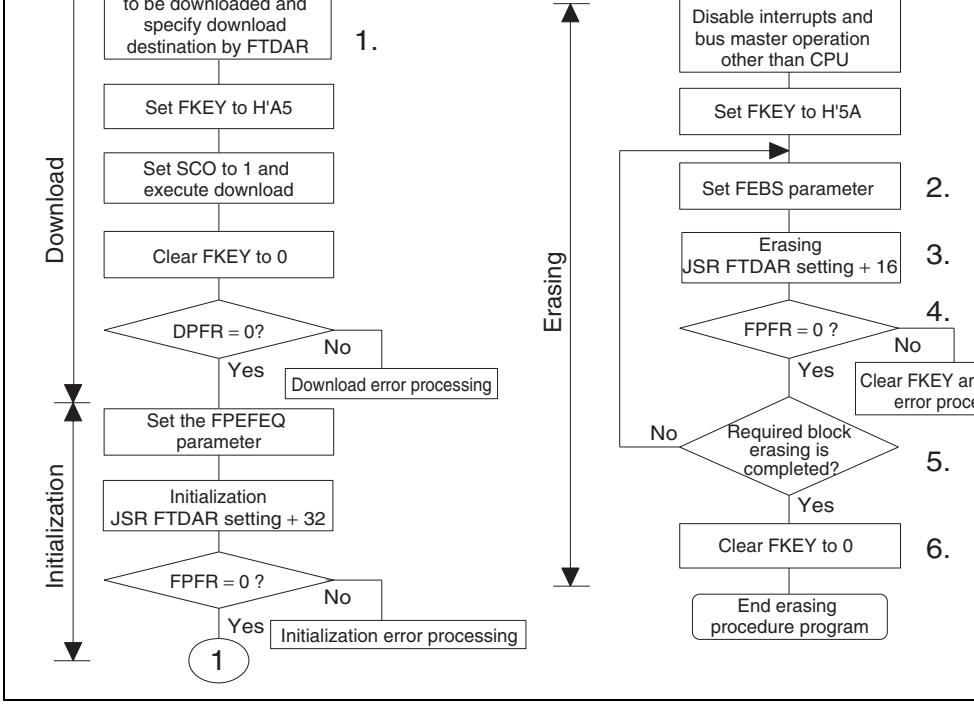


Figure 19.12 Erasing Procedure

The procedure program must be executed in an area other than the user MAT to be erased
Especially the part where the SCO bit in FCCS is set to 1 for downloading must be execu
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 19.4.4, Procedure Program and Storable Ar
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 initializatior same as those in the programming procedure. For details, refer to section 19.4.2 (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 paramete (general register ER0). If a value other than an erase block number of the user MAT 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 ar the start address of a download destination specified by FTDAR + 16 bytes of on-ch 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 th maximum must be allocated in RAM.

4. The return value in the erasing program, FPFR (general register R0L) is determined.

Figure 19.13 shows a repeating procedure of erasing and programming.

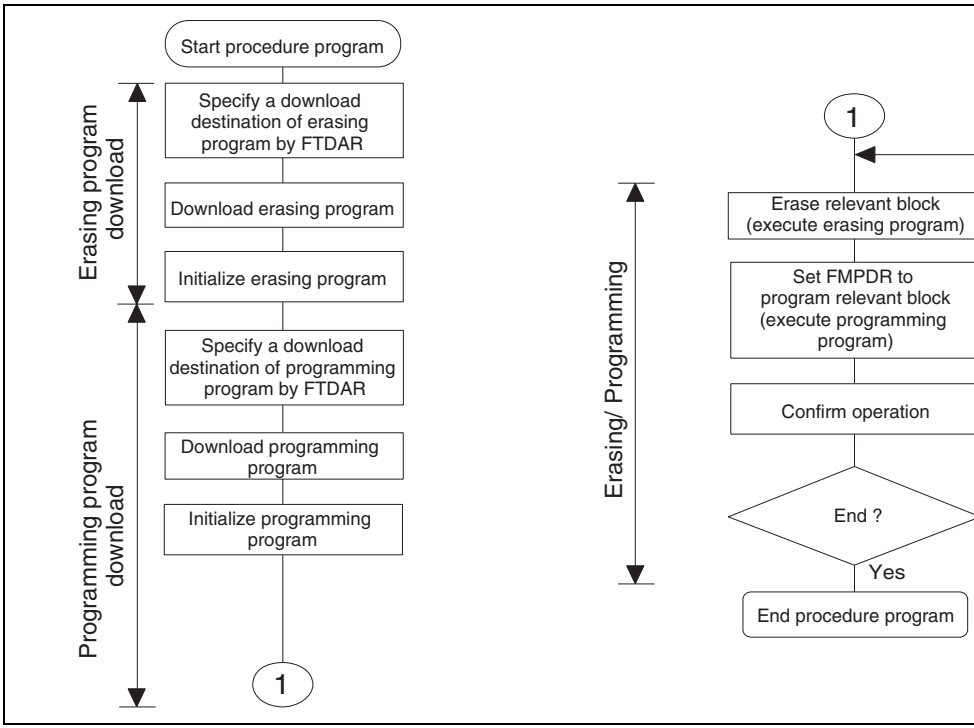


Figure 19.13 Repeating Procedure of Erasing and Programming

In the above procedure, download and initialization are performed only once at the begin

In this kind of operation, note the following:

For the mode pin settings to start up user boot mode, see table 19.5.

When the reset start is executed in user boot mode, the built-in check routine runs. The us
and user boot MAT states are checked by this check routine.

While the check routine is running, NMI and all other interrupts cannot be accepted.

Next, processing starts from the execution start address of the reset vector in the user boo
At this point, H'AA is set to FMATS because the execution MAT is the user boot MAT.

(2) User MAT Programming in User Boot Mode

For programming the user MAT in user boot mode, additional processing made by 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 19.14 shows the procedure for programming the user MAT in user boot mode.

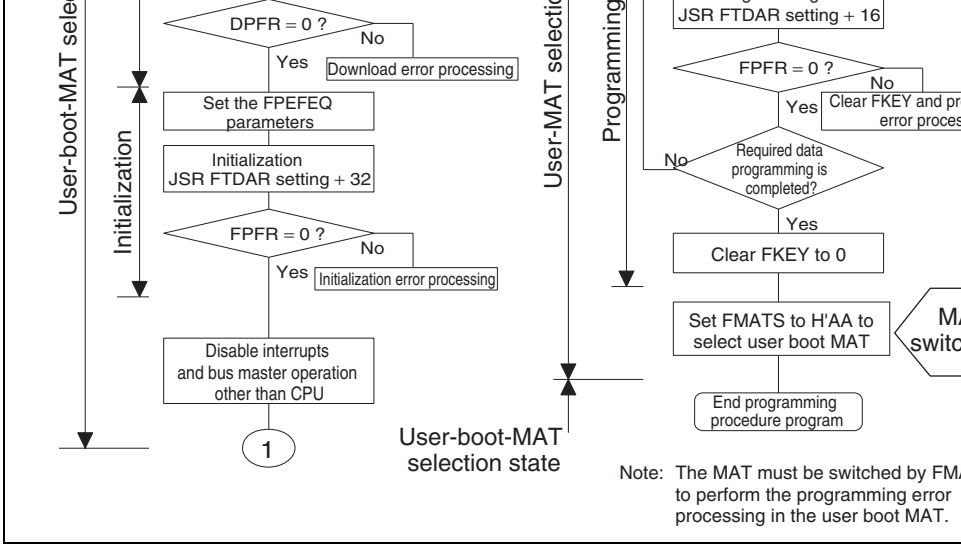


Figure 19.14 Procedure for Programming User MAT in User Boot Mode

The difference between the programming procedures in user program mode and user bo whether the MAT is switched or not as shown in figure 19.14.

In user boot mode, the user boot MAT can be seen in the flash memory space with the u hidden in the background. The user MAT and user boot MAT are switched only while th MAT is being programmed. Because the user boot MAT is hidden while the user MAT programmed, the procedure program must be located in an area other than flash memory programming completes, switch the MATs again to return to the first state.

MAT switching is enabled by writing a specific value to FMATS. However note that wh MATs are being switched, the LSI is in an unstable state, e.g. access to a MAT is not all MAT switching is completed, and if an interrupt occurs, from which MAT the interrupt

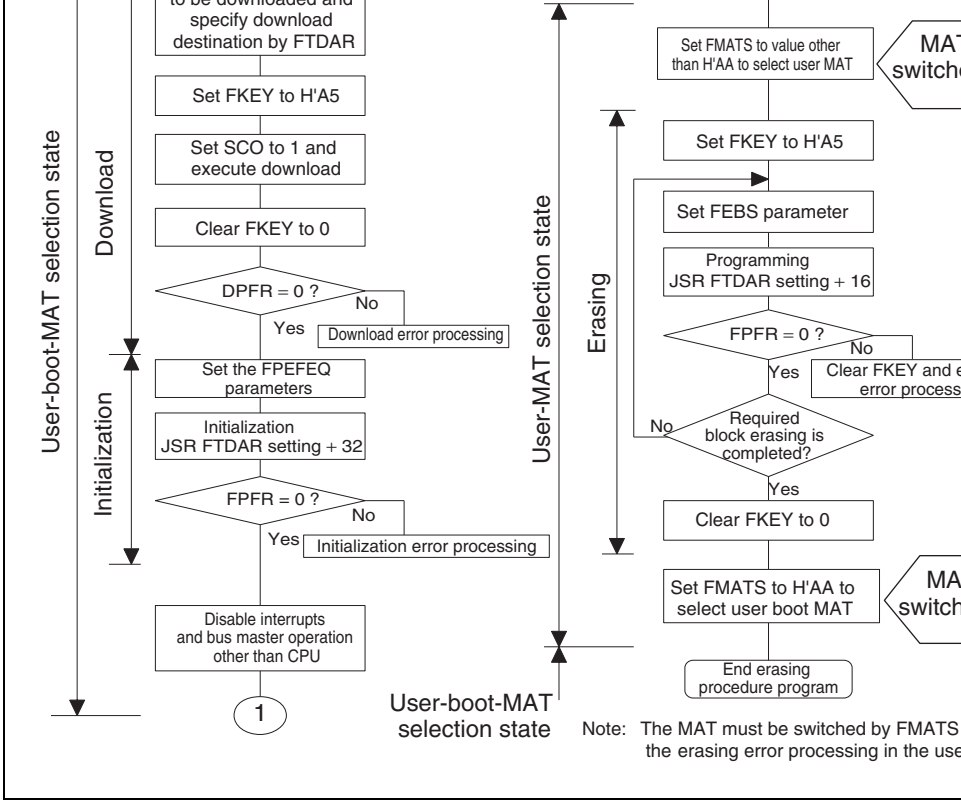


Figure 19.15 Procedure for Erasing User MAT in User Boot Mode

The difference between the erasing procedures in user program mode and user boot mod
on whether the MAT is switched or not as shown in figure 19.15.

2. The on-chip programming/erasing program will use 128 bytes at the maximum as a s
make sure that this area is secured.
3. Download by setting the SCO bit to 1 will lead to switching of the MAT. If, therefor
operation is used, it should be executed from the on-chip RAM.
4. The flash memory is accessible until the start of programming or erasing, that is, unt
result of downloading has been determined. When in a mode in which the external a
space is not accessible, such as single-chip mode, the required procedure programs, I
handling vector and NMI handler should be transferred to the on-chip RAM before
programming/erasing of the flash memory starts.
5. The flash memory is not accessible during programming/erasing operations, therefor
operation program is downloaded to the on-chip RAM to be executed. The NMI-han
vector and programs such as that which activate the operation program, and NMI ha
should thus be stored in on-chip memory other than flash memory or the external ad
space.
6. After programming/erasing, the flash memory should be inhibited until FKEY is clea
The reset state ($\overline{RES}$ = 0) must be in place for more than 100 μs when the LSI mode
to reset on completion of a programming/erasing operation.

   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 longe
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
boot MAT is operated in user-boot mode. The program which switches the MATs sh
executed from the on-chip RAM. See section 19.6, Switching between User MAT an
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
memory area, an error will occur even when the data stored is normal. Therefore, the

Programming	Table 19.8 (1)	Table 19.8 (3)
Erasing	Table 19.8 (2)	Table 19.8 (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	○	×	○

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 FKEY Clear	○	×	○	
Switching MATs by FMATS	○	×		○

Notes: 1. Transferring the data to the on-chip RAM enables this area to be used.
2. Switching FMATS by a program in the on-chip RAM enables this area to be us

Execution of Writing SCO = 1 to FCCS (Download)	○	×		
Operation for FKEY Clear	○	○		○
Determination of Download Result	○	○		○
Operation for Download Error	○	○		○
Operation for Settings of Initial Parameter	○	○		○
Execution of Initialization	○	×		○
Determination of Initialization Result	○	○		○
Operation for Initialization Error	○	○		○
NMI Handling Routine	○	×		○
Operation for Interrupt Inhibit	○	○		○
Switching MATs by FMATS	○	×		○
Operation for Writing H'5A to FKEY	○	×	○	

Operation for FKEY Clear	○	×	○
Switching MATs by FMATS	○	×	○

Note: * Switching FMATS by a program in the on-chip RAM enables this area to be us

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.	○	○

19.5.3 Error Protection

Error protection is a mechanism for aborting programming or erasure when an error occ
form of the microcomputer entering runaway during programming/erasing of the flash m
operations that are not according to the established procedures for programming/erasing
programming or erasure in such cases prevents damage to the flash memory due to exce
programming or erasing.

If the microcomputer malfunctions during programming/erasing of the flash memory, th
bit in the FCCS register is set to 1 and the error-protection state is entered, and this abor
programming or erasure.

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 reac
   instruction fetch).
3. When a SLEEP instruction (including software-standby mode) is executed during
   programming/erasing.

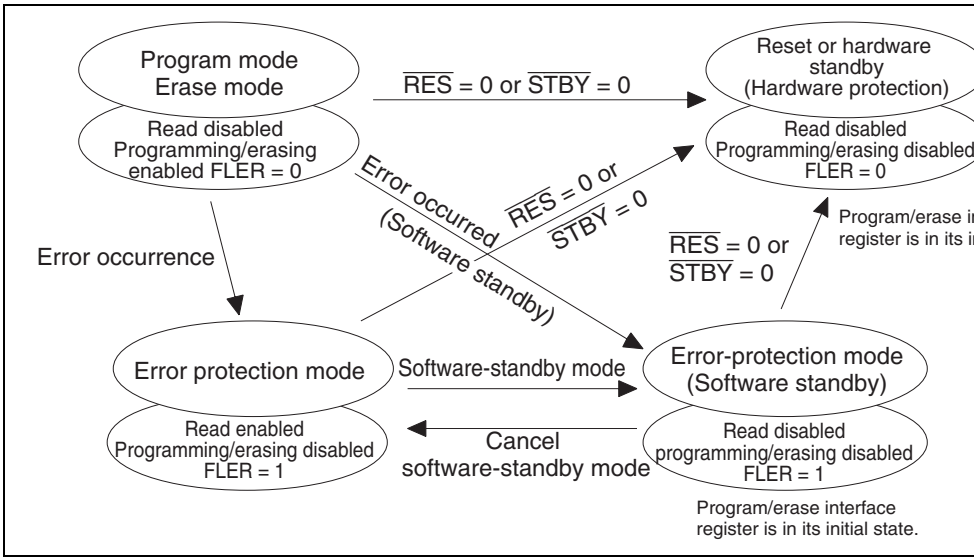


Figure 19.16 Transitions to Error-Protection State

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 MA
addition, configure the system so that NMI interrupts do not occur during MAT swit
4. After the MATs have been switched, take care because the interrupt vector table wil
been switched. If interrupt processing is to be the same before and after MAT switch
transfer the interrupt-processing routines to the on-chip RAM and set the WEINTE b
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 th
boot MAT, do not access addresses above the top of its 16-Kbyte memory space. If a
goes beyond the 16-Kbyte space, the values read are undefined.

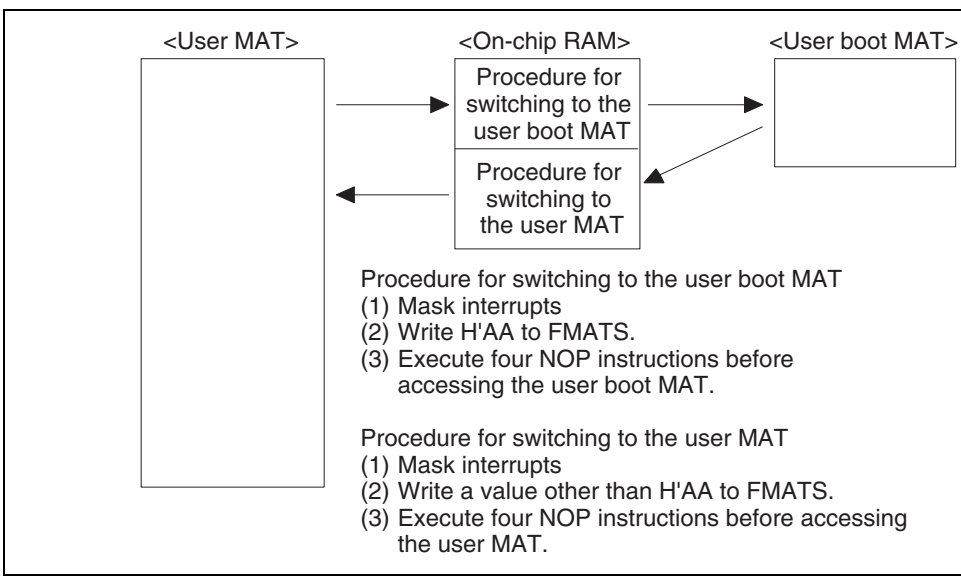


Figure 19.17 Switching between the User MAT and User Boot MAT

execution of automatic programming or automatic erasure. In programmer mode, provide
MHz input-clock signal.

Notes: 1. For the PROM programmer and the version of its program, see the instruction
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. program receives the command from the host to adjust the bit rate. After adjusting th the program enters the inquiry/selection state.

2. Inquiry/Selection State

   In this state, the boot program responds to inquiry commands from the host. The dev clock mode, and bit rate are selected. After selection of these settings, the program is enter the programming/erasing state by the command for a transition to the programming/erasing state. The program transfers the libraries required for erasure t 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 prog made to transfer the programming/erasing programs to the RAM by commands from Sum checks and blank checks are executed by sending these commands from the hos

These boot program states are shown in figure 19.18.

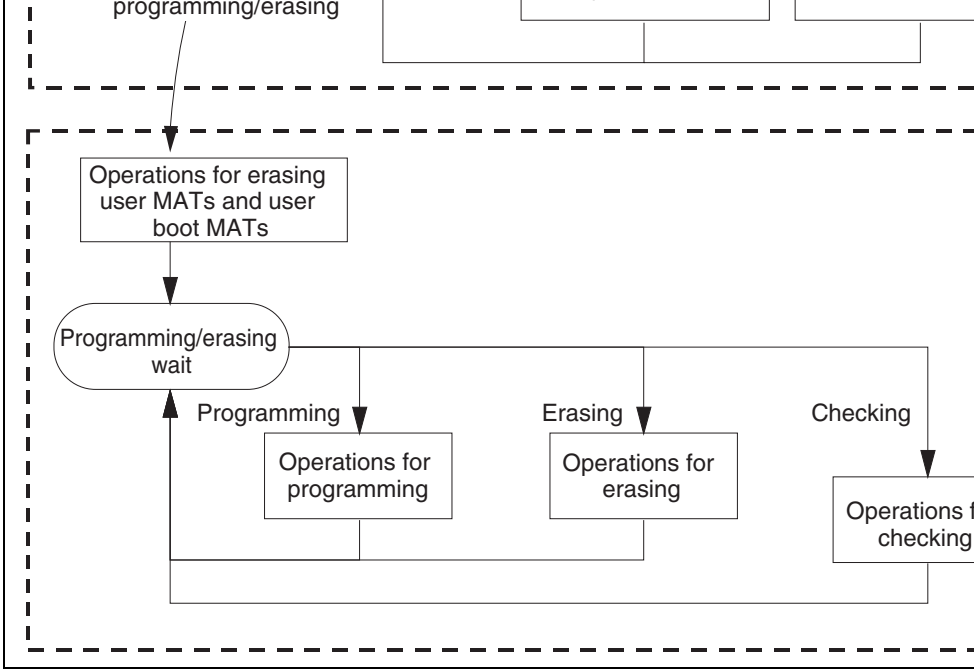


Figure 19.18 Boot Program States

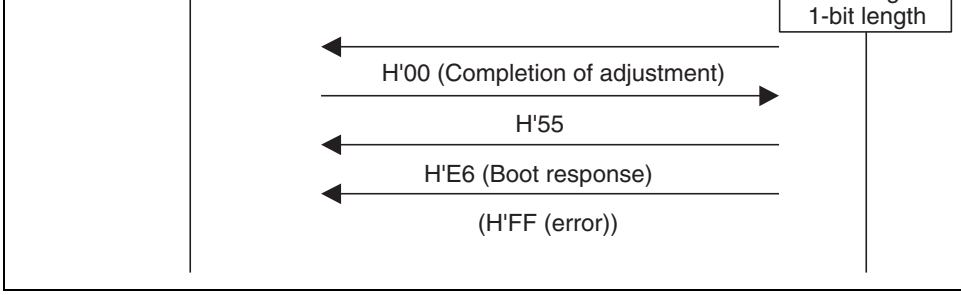


Figure 19.19 Bit-Rate-Adjustment Sequence

(3) Communications Protocol

After adjustment of the bit rate, the protocol for communications between the host and t
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 selection
   responses to inquiries.

   The amount of programming data is not included under this heading because it is det
   in another command.
3. Error response

   The error response is a response to inquiries. It consists of an error response and an e
   and comes two bytes.

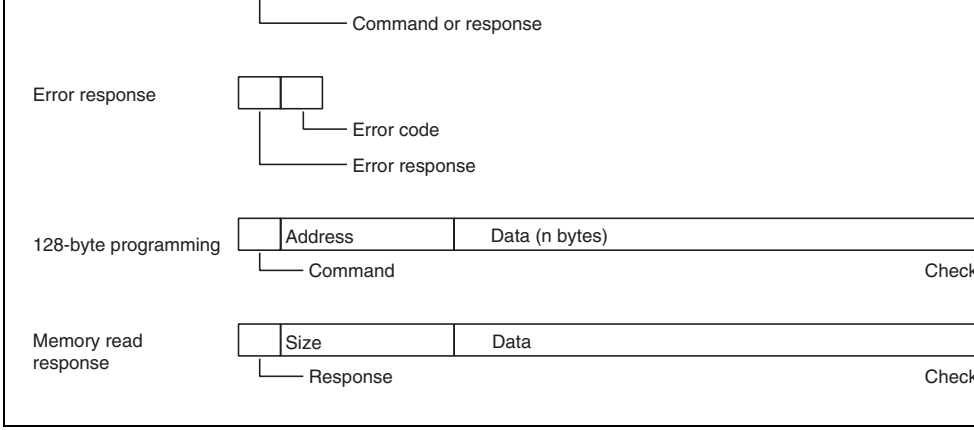


Figure 19.20 Communication Protocol Format

- 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 th command byte to the SUM byte becomes H'00.
- Data (n bytes): Detailed data of a command or response
- Error response (1 byte): Error response to a command
- Error code (1 byte): Type of the error
- Address (4 bytes): Address for programming
- Data (n bytes): Data to be programmed (the size is indicated in the response to the programming unit inquiry.)

Command	Command Name	Description
H'20	Supported Device Inquiry	Inquiry regarding device codes
H'10	Device Selection	Selection of device code
H'21	Clock Mode Inquiry	Inquiry regarding numbers of clock r values of each mode
H'11	Clock Mode Selection	Indication of the selected clock mod
H'22	Multiplication Ratio Inquiry	Inquiry regarding the number of freq multiplied clock types, the number o multiplication ratios, and the values multiple
H'23	Operating Clock Frequency Inquiry	Inquiry regarding the maximum and values of the main clock and periphe
H'24	User Boot MAT Information Inquiry	Inquiry regarding the number of use MATs and the start and last address each MAT
H'25	User MAT Information Inquiry	Inquiry regarding the a number of us and the start and last addresses of e
H'26	Block for Erasing Information Inquiry	Inquiry regarding the number of bloc the start and last addresses of each
H'27	Programming Unit Inquiry	Inquiry regarding the unit of program
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 entry to programming/erasing state
H'4F	Boot Program Status Inquiry	Inquiry into the operated status of th program

(a) Supported Device Inquiry

The boot program will return the device codes of supported devices and the product code
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, character codes and product names
- Number of devices (1 byte): The number of device types supported by the boot progr
- Number of characters (1 byte): The number of characters in the device codes and boo 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 b 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 i

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 comman
- 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 cl number of types will be H'02.)
- Number of multiplication ratios (1 byte): The number of multiplication ratios for ea (e.g. the number of multiplication ratios to which the main clock can be set and the p clock can be set.)
- Multiplication ratio (1 byte)

  Multiplication ratio: The value of the multiplication ratio (e.g. when the clock-freque multiplier is four, the value of multiplication ratio will be H'04.)

  Division ratio: The inverse of the division ratio, i.e. a negative number (e.g. when th divided by two, the value of division ratio will be H'FE. H'FE = D'-2)

  The number of multiplication ratios returned is the same as the number of multiplica and as many groups of data are returned as there are types.
- SUM (1 byte): Checksum

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 va 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 peri clocks, the number of types will be H'02.)
- Minimum value of operating clock frequency (2 bytes): The minimum value of the m or divided clock frequency.

  The minimum and maximum values represent the values in MHz, valid to the hundred of MHz, and multiplied by 100. (e.g. when the value is 20.00 MHz, it will be 2000, w H'07D0.)
- Maximum value (2 bytes): Maximum value among the multiplied or divided clock frequencies.

  There are as many pairs of minimum and maximum values as there are operating cloc 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 ad
  and area-last address
- Number of Areas (1 byte): The number of consecutive user boot MAT areas
  When user boot MAT areas are consecutive, the number of areas returned is H'01.
- Area-start address (4 byte): Start address of the area
- Area-last address (4 byte): Last address of the area
  There are as many groups of data representing the start and last addresses as there ar
- SUM (1 byte): Checksum

(h) User MAT Information Inquiry

The boot program will return the number of user MATs and their addresses.

Command

H'25

- Command, H'25, (1 byte): Inquiry regarding user MAT information

Response

H'35	Size	Number of areas	
Start address area			Last address area
...			
SUM			

- Response, H'35, (1 byte): Response to the user MAT information inquiry
- Size (1 byte): The number of bytes that represents the number of areas, area-start ad
  area-last address
- Number of areas (1 byte): The number of consecutive user MAT areas
  When the user MAT areas are consecutive, the number of areas is H'01.
- Area-start address (4 bytes): Start address of the area

Block start address	Block last address
...	
SUM	

- Response, H'36, (1 byte): Response to the number of erased blocks and addresses
- Size (three bytes): The number of bytes that represents the number of blocks, block-s
  addresses, and block-last addresses.
- Number of blocks (1 byte): The number of erased blocks
- Block start address (4 bytes): Start address of a block
- Block last Address (4 bytes): Last address of a block
  There are as many groups of data representing the start and last addresses as there are
- SUM (1 byte): Checksum

(j) Programming Unit Inquiry

The boot program will return the programming unit used to program data.

Command

H'27

- Command, H'27, (1 byte): Inquiry regarding programming unit

Response

H'37	Size	Programming unit	SUM

- Response, H'37, (1 byte): Response to programming unit inquiry
- Size (1 byte): The number of bytes that indicate the programming unit, which is fixe
- Programming unit (2 bytes): A unit for programming
  This is the unit for reception of programming.
- SUM (1 byte): Checksum

- Size(1 byte): The number of bytes that represents the bit rate, input frequency, num
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 i
H′00C0.)
- Input frequency (2 bytes): Frequency of the clock input to the boot program
This is valid to the hundredths place and represents the value in MHz multiplied by
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 wh
device can be set.
- Multiplication ratio 1 (1 byte) : The value of multiplication or division ratios for th
operating frequency
Multiplication ratio (1 byte): The value of the multiplication ratio (e.g. when the cloc
frequency is multiplied by four, the multiplication ratio will be H'04.)
Division ratio: The inverse of the division ratio, as a negative number (e.g. when the
frequency is divided by two, the value of division ratio will be H'FE. H'FE = D'-2)
- Multiplication ratio 2 (1 byte): The value of multiplication or division ratios for the
frequency
Multiplication ratio (1 byte): The value of the multiplication ratio (e.g. when the cloc
frequency is multiplied by four, the multiplication ratio will be H'04.)
(Division ratio: The inverse of the division ratio, as a negative number (E.g. when th
divided by two, the value of division ratio will be H'FE. H'FE = D'-2)
- SUM (1 byte): Checksum

Response | H'06 |

- Response, H'06, (1 byte): Response to selection of a new bit rate
When it is possible to set the bit rate, the response will be ACK.

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 o minimum to maximum frequencies which matches the clock modes of the specified devic the value is out of this range, an input-frequency error is generated.

2. Multiplication ratio

The received value of the multiplication ratio or division ratio is checked to ensure that it the clock modes of the specified device. When the value is out of this range, an multiplica ratio error is generated.

3. Operating frequency

Operating frequency is calculated from the received value of the input frequency and the multiplication or division ratio. The input frequency is input to the LSI and the LSI is ope the operating frequency. The expression is given below.

Operating frequency = Input frequency × Multiplication ratio, or
Operating frequency = Input frequency ÷ Division ratio

The calculated operating frequency should be checked to ensure that it is within the range minimum to maximum frequencies which are available with the clock modes of the speci device. When it is out of this range, an operating frequency error is generated.

response. The host will send an ACK with the new bit rate for confirmation and the boo
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 19.21.

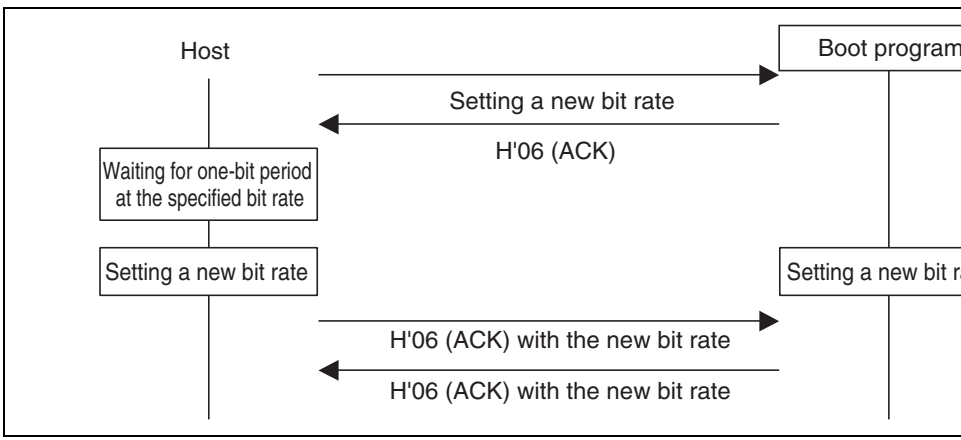


Figure 19.21 New Bit-Rate Selection Sequence

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 been
  by the transferred erasing program.

Error Response | H'C0 | H'51

- Error response, H'C0, (1 byte): Error response for user boot MAT blank check
- Error code, H'51, (1 byte): Erasing error
  An error occurred and erasure was not completed.

(7) Command Error

A command error will occur when a command is undefined, the order of commands is in
or a command is unacceptable. Issuing a clock-mode selection command before a device
or an inquiry command after the transition to programming/erasing state command, are e

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 inqui
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, ac
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
MAT should be made to inquire about the user boot MATs information inquiry (H'2
MATs information inquiry (H'25), erased block information inquiry (H'26), and prog
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 programmi 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 b
H'4B	User MAT sum check	Checks the checksum of the user M
H'4C	User boot MAT blank check	Checks whether the contents of the boot MAT are blank
H'4D	User MAT blank check	Checks whether the contents of the MAT are blank
H'4F	Boot program status inquiry	Inquires into the boot program's sta

programming command. The 128-byte programming command that follows the selec
command represents the data programmed according to the method specified by the
command. When more than 128-byte data is programmed, 128-byte commands shou
repeatedly be executed. Sending a 128-byte programming command with H'FFFFFF
address will stop the programming. On completion of programming, the boot progra
wait for selection of programming or erasing.

Where the sequence of programming operations that is executed includes programm
another method or of another MAT, the procedure must be repeated from the progra
selection command.

The sequence for programming-selection and 128-byte programming commands is s
figure 19.22.

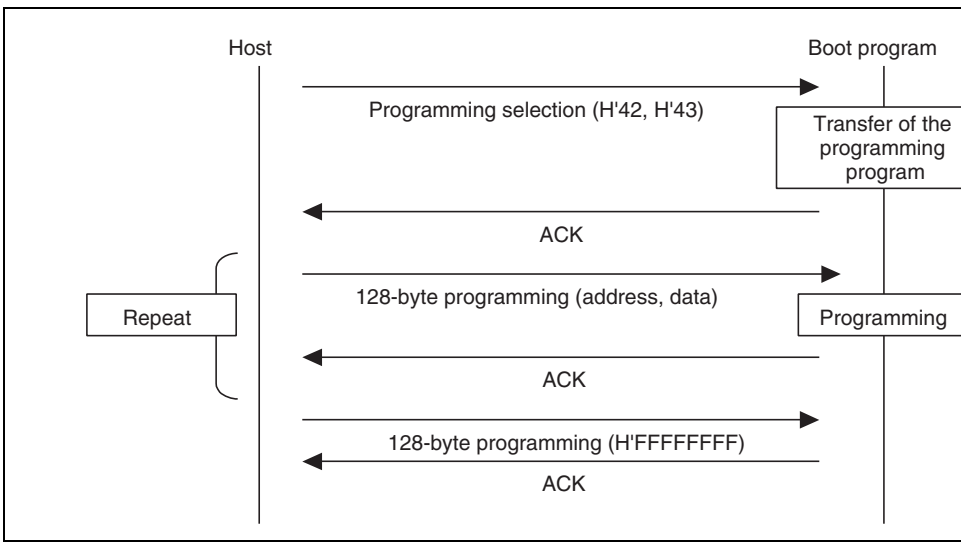


Figure 19.22 Programming Sequence

Error Response	H'C2	ERROR

- Error response : H'C2 (1 byte): Error response to user boot MAT programming selec
- ERROR : (1 byte): Error code

  H'54: Selection processing error (transfer error occurs and processing is not complete

- 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 complete

(b) 128-byte programming

The boot program will use the programming program transferred by the programming sel
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 contin

The specified address should match the unit for programming of data. For example, whe 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 res 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 programm 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 e
specified block. The command should be repeatedly executed if two or more blocks are to
erased. Sending a block-erasure command from the host with the block number H'FF will
erasure operating. On completion of erasing, the boot program will wait for selection of
programming or erasing.

The sequences of issuing the erasure selection command and block-erasure command are
figure 19.23.

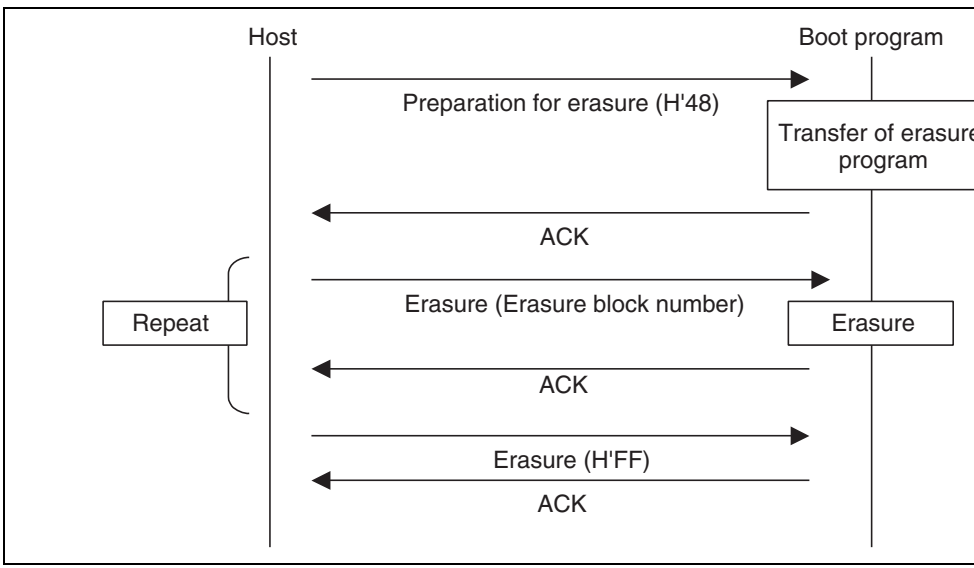


Figure 19.23 Erasure Sequence

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 compl

(b) Block Erasure

The boot program will erase the contents of the specified block.

Command

H'58	Size	Block number	SUM

- Command, H'58, (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 proce
  should be executed from the erasure selection command.

(11) Memory read

The boot program will return the data in the specified address.

Command	H'52	Size	Area	Read address	
	Read size				SUM

- Command: H'52 (1 byte): Memory read
- Size (1 byte): Amount of data that represents the area, read address, and read size (fi
- Area (1 byte)

  H'00: User boot MAT
  H'01: User MAT
  An address error occurs when the area setting is incorrect.
- Read address (4 bytes): Start address to be read from
- Read size (4 bytes): Size of data to be read
- SUM (1 byte): Checksum

Response	H'52	Read size						
	Data	···						
	SUM							

- Response: H'52 (1 byte): Response to memory read
- Read size (4 bytes): Size of data to be read
- Data (n bytes): Data for the read size from the read address
- SUM (1 byte): Checksum

The boot program will return the byte-by-byte total of the contents of the bytes of the us
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 us

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 resul

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 19.13 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
Do not set the programmer to HN28F101 or the programming voltage to 5.0 V. Use
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.
executed accidentally, reset must be released after the reset input period of 100 μs w
longer than normal.
6. The flash memory is not accessible until FKEY is cleared after programming/erasing
completes. If this LSI is restarted by a reset immediately after programming/erasing
finished, secure the reset period (period of $\overline{RES}$ = 0) of more than 100 μs. Though tr
the reset state or hardware standby state during programming/erasing is prohibited, i
executed accidentally, reset must be released after the reset input period of 100 μs w
longer than normal.
7. At powering on or off the Vcc power supply, fix the $\overline{RES}$ pin to low and set the flash
to hardware protection state. This power on/off timing must also be satisfied at a pov
power-on caused by a power failure and other factors.
8. Program the area with 128-byte programming-unit blocks in on-board programming
programmer mode only once. Perform programming in the state where the programm
block is fully erased.
9. When the chip is to be reprogrammed with the programmer after execution of progra
erasure in on-board programming mode, it is recommend that automatic programmin
performed after execution of automatic erasure.
10. To write data or programs to the flash memory, data or programs must be allocated t
addresses higher than that of the external interrupt vector table (H'000040) and H'FF
written to the areas that are reserved for the system in the exception handling vector

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 out
control. Do not use DTC to program flash related registers.
14. A programming/erasing program for flash memory used in the conventional H8S F-Z
microcomputer which does not support download of the on-chip program by a SCO tr
request cannot run in this LSI. Be sure to download the on-chip program to execute
programming/erasing of flash memory in this LSI.
15. Unlike the conventional H8S F-ZTAT microcomputer, no countermeasures are availa
runaway by WDT during programming/erasing. Prepare countermeasures (e.g. use of
periodic timer interrupts) for WDT with taking the programming/erasing time into
consideration as required.

20.1 Features

- Five test pins (ETCK, ETDI, ETDO, ETMS, and $\overline{\text{ETRST}}$)
- TAP controller
- Six instructions

  BYPASS mode

  EXTEST mode

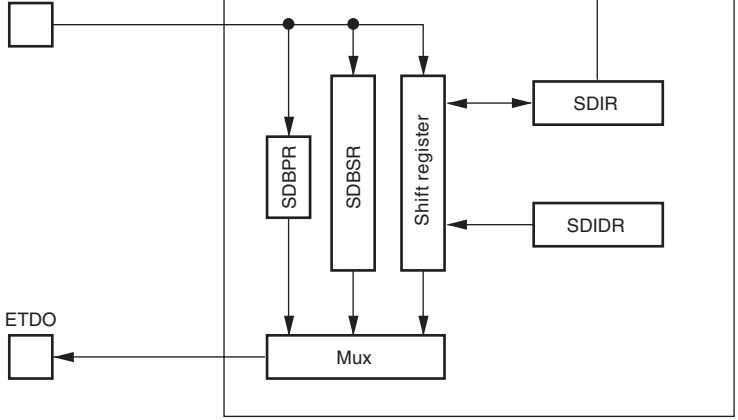
  SAMPLE/PRELOAD mode

  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 20.1 JTAG Block Diagram

			with a duty cycle close to 50% should be details, see section 24, Electrical Charac
Test mode select	ETMS	Input	Test Mode Select Input Sampled on the rise of the ETCK pin. Th pin controls the internal state of the TAP
Test data input	ETDI	Input	Serial Data Input Performs serial input of instructions and JTAG registers. ETDI is sampled on the ETCK pin.
Test data output	ETDO	Output	Serial Data Output Performs serial output of instructions and from JTAG registers. Transfer is perform synchronization with the ETCK pin. If the output, the ETDO pin goes to the high-im state.
Test reset	$\overline{\text{ETRST}}$	Input	Test Reset Input Signal Initializes the JTAG asynchronously.

input pin (ETDI). Data from SDIR can be output via the test data output pin (ETDO). The register (SDBPR) is a 1-bit register to which the ETDI and ETDO pins are connected in B CLAMP, or HIGHZ mode. The boundary scan register (SDBSR) is a 210-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 20.2 shows the kinds of serial transfer possible with each JTAG register.

Table 20.2 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
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
0	—	1	R	Reserved
				This bit is always read as 1 and cannot be modi

Table 20.3 shows the relationship between the pins of this LSI and the boundary scan reg

		Enable	210			—	
		Output	209			—	
B01	P46	Input	208	E02	NC	—	
		Enable	207			—	
		Output	206			—	
D04	P47	Input	205	F03	$\overline{\text{MD2}}$	Input	
		Enable	204			Reserved	
		Output	203			Reserved	
C02	P56	Input	202	F01	PC7	Input	
		Enable	201			Enable	
		Output	200			Output	
C01	P57	Input	199	F02	PC6	Input	
		Enable	198			Enable	
		Output	197			Output	
D03	VSS	—	—	F04	PC3	Input	
		—	—			Enable	
		—	—			Output	
D02	$\overline{\text{RES}}$	—	—	G01	PC2	Input	
		—	—			Enable	
		—	—			Output	
D01	MD1	Input	196	G02	PC1	Input	
		—	—			Enable	
		—	—			Output	

H02	PA5	Input	167	J04	P85	Input	1
		Enable	166			Enable	1
		Output	165			Output	1
J01	VCC	—	—	K04	P84	Input	1
		—	—			Enable	1
		—	—			Output	1
H03	NC	—	—	L04	P83	Input	1
		—	—			Enable	1
		—	—			Output	1
J02	PA4	Input	164	H05	P82	Input	1
		Enable	163			Enable	1
		Output	162			Output	1
K01	PA3	Input	161	J05	P81	Input	1
		Enable	160			Enable	1
		Output	159			Output	1
J03	NC	—	—	L05	P80	Input	1
		—	—			Enable	1
		—	—			Output	1
L01	PA2	Input	158	K05	NC	—	–
		Enable	157			—	–
		Output	156			—	–
K02	PA1	Input	155	J06	PE7	Input	1
		Enable	154			Enable	1
		Output	153			Output	1
L02	PA0	Input	152	L06	PE6	Input	1
		Enable	151			Enable	1
		Output	150			Output	1

K07	PE2	Input	110	K11	P76	Input	
		Enable	109			—	
		Output	108			—	
J07	PE1	Input	107	H08	P77	Input	
		Enable	106			—	
		Output	105			—	
L08	PE0	Input	104	J10	AVCC	—	
		Enable	103			—	
		Output	102			—	
H07	AVSS	—	—	J11	AVref	—	
		—	—			—	
		—	—			—	
K08	P70	Input	101	H09	P60	Input	
		—	—			Enable	
		—	—			Output	
L09	P71	Input	100	H10	P61	Input	
		—	—			Enable	
		—	—			Output	
J08	NC	—	—	H11	P62	Input	
		—	—			Enable	
		—	—			Output	
K09	P72	Input	99	G08	P63	Input	
		—	—			Enable	
		—	—			Output	
L10	P73	Input	98	G09	VCC	—	
		—	—			—	
		—	—			—	

F11	ETDI	—	—	B11	P14	Input	6
		—	—			Enable	5
		—	—			Output	5
F10	ETCK	—	—	C09	NC	—	–
		—	—			—	–
		—	—			—	–
F08	$\overline{\text{ETRST}}$	—	—	A11	P13	Input	5
		—	—			Enable	5
		—	—			Output	5
E11	VSS	—	—	B10	P12	Input	5
		—	—			Enable	5
		—	—			Output	5
E10	P23	Input	81	A10	P11	Input	5
		Enable	80			Enable	5
		Output	79			Output	4
E09	P22	Input	78	D08	VSS	—	–
		Enable	77			—	–
		Output	76			—	–
D11	P21	Input	75	B09	P10	Input	4
		Enable	74			Enable	4
		Output	73			Output	4
E08	P20	Input	72	A09	P30	Input	4
		Enable	71			Enable	4
		Output	70			Output	4
D10	P17	Input	69	C08	P31	Input	4
		Enable	68			Enable	4
		Output	67			Output	4

C07	P35	Input	30	D05	P44	Input	
		Enable	29			Enable	
		Output	28			Output	
A07	P36	Input	27	B04	VSS	—	
		Enable	26			—	
		Output	25			—	
B07	NC	—	—	A03	$\overline{\text{RESO}}$	—	
		—	—			—	
		—	—			—	
C06	P37	Input	24	C04	NC	—	
		Enable	23			—	
		Output	22			—	
A06	P40	Input	21	B03	XTAL	—	
		Enable	20			—	
		Output	19			—	
B06	P41	Input	18	A02	EXTAL	—	
		Enable	17			—	
		Output	16			—	
D06	P42	Input	15	C03	NC	—	
		Enable	14			—	
		Output	13			—	
A05	P43	Input	12			to ETDO	
		Enable	11				
		Output	10				

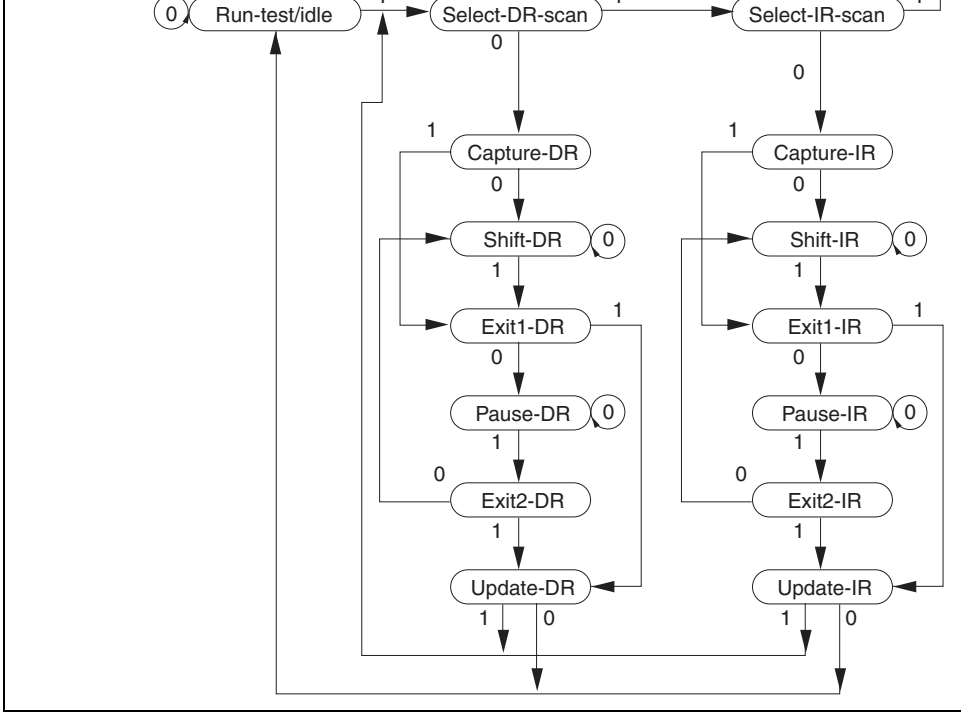


Figure 20.2 TAP Controller State Transitions

by setting a command in SDIR.

20.5.1 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 outpu
This LSI system circuits are not affected by execution of this instruction.

In a SAMPLE operation, a snapshot of a value to be transferred from an input pin to the i
circuitry, or a value to be transferred from the internal circuitry to an output pin, is latche
boundary scan register and read from the scan path. Snapshot latching does not affect nor
operation of this LSI.

printed circuit board, and input pins are used to latch test results into the boundary scan
from the printed circuit board. If testing is carried out by using the EXTEST instruction
the Nth test data is scanned in when test data (N-1) is scanned out.

Data loaded into the output pin boundary scan register in the Capture-DR state is not use
external circuit testing (it is replaced by a shift operation).

(4) CLAMP (Instruction code: B'0010)

When the CLAMP instruction is enabled, the output pin outputs the value of the bounda
register that has been previously set by the SAMPLE/PRELOAD instruction. While the
instruction is enabled, the state of the boundary scan register maintains the previous stat
regardless of the state of the TAP controller.

A bypass register is connected between the ETDI and ETDO pins. The related circuit op
the same way when the BYPASS instruction is enabled.

(5) HIGHZ (Instruction code: B'0011)

When the HIGHZ instruction is enabled, all output pins enter a high-impedance state. W
HIGHZ instruction is enabled, the state of the boundary scan register maintains the prev
regardless of the state of the TAP controller.

A bypass register is connected between the ETDI and ETDO pins. The related circuit op
the same way when the BYPASS instruction is enabled.

2. Boundary scan mode does not cover clock-related pins (EXTAL, XTAL, and
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 20.3 shows a design example of the reset signal circuit wherein no reset signa
interference occurs.

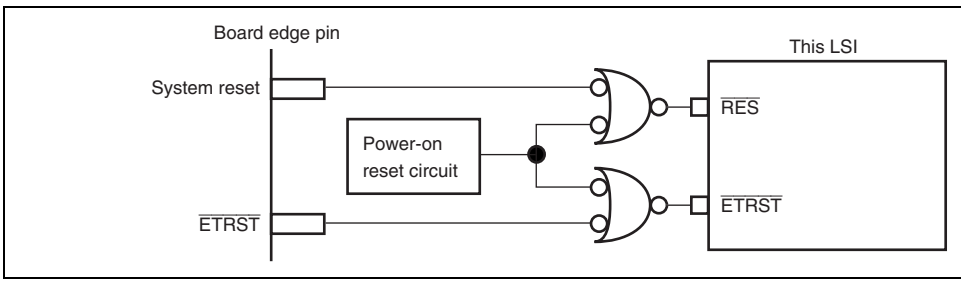


Figure 20.3 Reset Signal Circuit Without Reset Signal Interference

3. The registers are not initialized in standby mode. If the $\overline{\text{ETRST}}$ pin is set to 0 in stan
IDCODE mode will be entered.
4. The frequency of the ETCK pin must be lower than that of the system clock. For det
section 24, Electrical Characteristics.
5. Data input/output in serial data transfer starts from the LSB. Figures 20.4 and 20.5 sl
examples of serial data input/output.
6. When data that exceeds the number of bits of the register connected between the ET
ETDO pins is serially transferred, the serial data that exceeds the number of register
output from the ETDO pin is the same as that input from the ETDI pin.
7. If the JTAG serial transfer sequence is disrupted, the $\overline{\text{ETRST}}$ pin must be reset. Tran
should then be retried, regardless of the transfer operation.

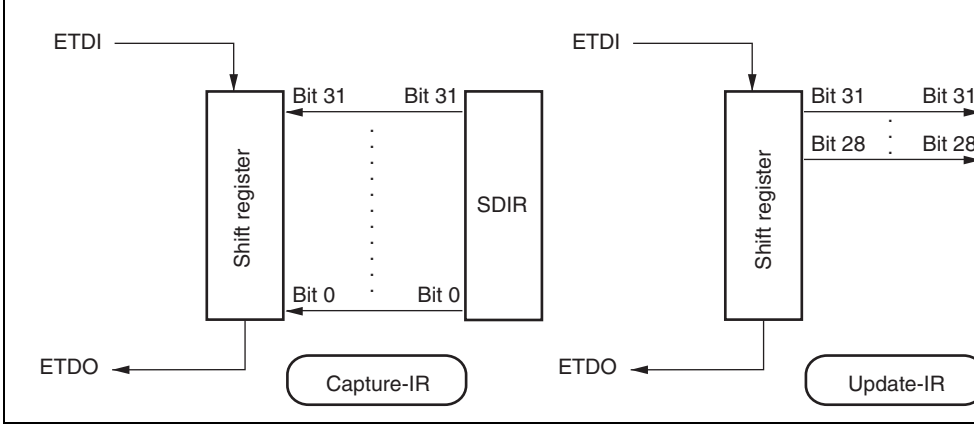


Figure 20.4 Serial Data Input/Output (1)

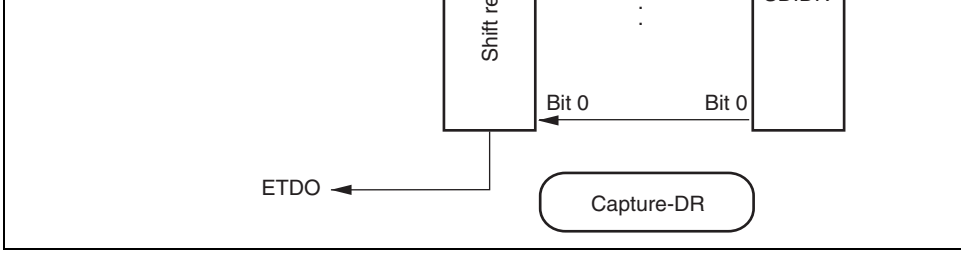


Figure 20.5 Serial Data Input/Output (2)

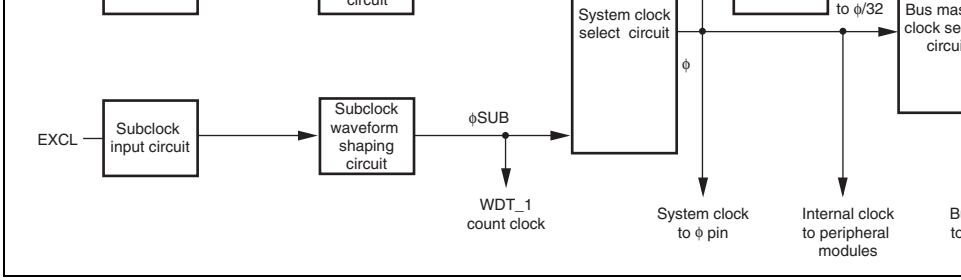


Figure 21.1 Block Diagram of Clock Pulse Generator

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 22.1.1, 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 22.1.2, Low-P
Control Register (LPWRCR).

Figure 21.3 shows the equivalent circuit of a crystal resonator. A crystal resonator having characteristics given in table 21.2 should be used.

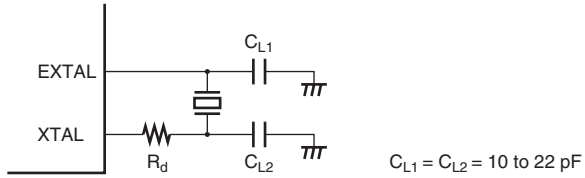


Figure 21.2 Typical Connection to Crystal Resonator

Table 21.1 Damping Resistance Values

Frequency (MHz)	5	6.25
R_d (Ω)	300	240

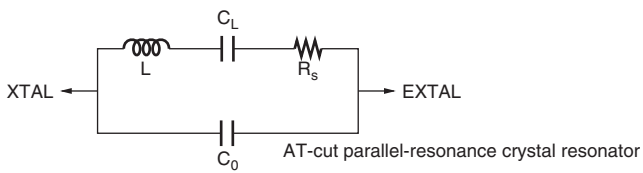


Figure 21.3 Equivalent Circuit of Crystal Resonator

To input an inverted clock to the XTAL pin, the external clock should be tied to high in
mode.

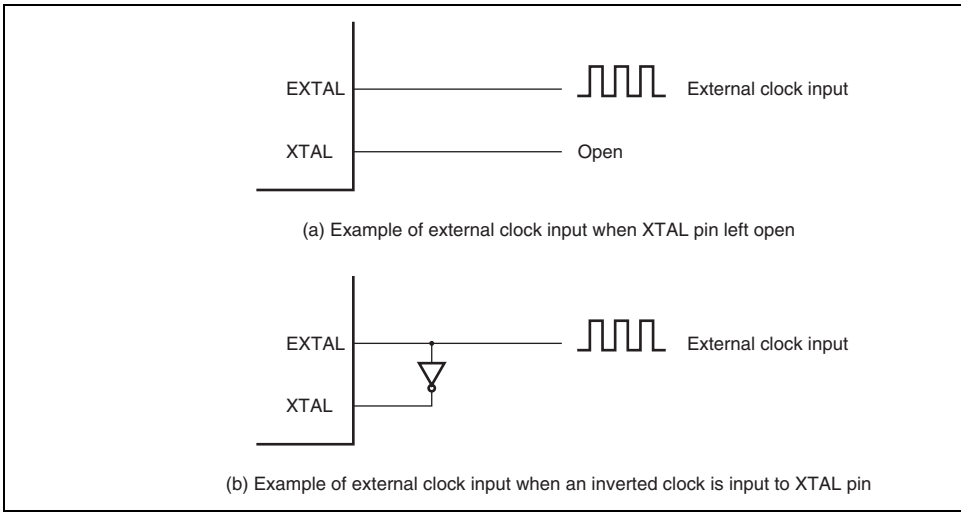


Figure 21.4 Example of External Clock Input

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 24.5 ar
24.8 in section 24, Electrical Characteristics.

21.3 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.

21.4 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.

21.5 Subclock Input Circuit

The subclock input circuit controls subclock input from the EXCL pin. To use the subclo
32.768-kHz external clock should be input from the EXCL pin. At this time, the P56DDR
P5DDR should be cleared to 0, and the EXCLE bit in LPWRCR should be set to 1.

When the subclock is not used, subclock input should not be enabled.

21.6 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.

21.8.1 Note on Resonator

Since all kinds of characteristics of the resonator are closely related to the board design
user, use the example of resonator connection in this document for only reference; be su
an resonator that has been sufficiently evaluated by the user. Consult with the resonator
manufacturer about the resonator circuit ratings which vary depending on the stray capa
the resonator and installation circuit. Make sure the voltage applied to the oscillation pin
exceed the maximum rating.

21.8.2 Notes on Board Design

When using a crystal resonator, the crystal resonator and its load capacitors should be pl
close as possible to the EXTAL and XTAL pins. Other signal lines should be routed awa
the oscillation circuit to prevent inductive interference with the correct oscillation as sho
figure 21.5.

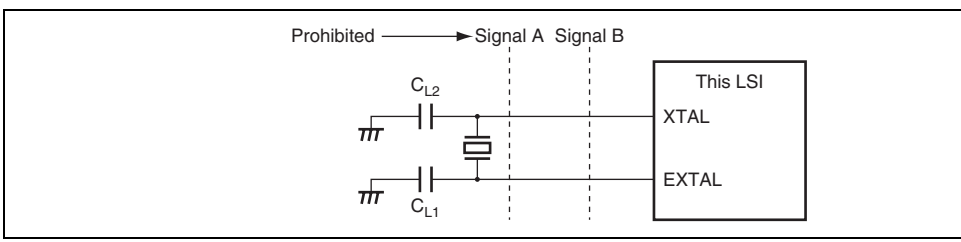


Figure 21.5 Note on Board Design of Oscillation Circuit Section

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)

22.1.1 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 master other than the CPU in medium-speed mode. 0: All bus masters operate based on the mediu clock. 1: The DTC operates based on the system cloc The operating clock is changed when a DTC tra requested even if the CPU operates based on th medium-speed clock.
2	SCK2	0	R/W	System Clock Select 2 to 0
1	SCK1	0	R/W	Select a clock for the bus master in high-speed medium-speed mode.
0	SCK0	0	R/W	000: High-speed mode (Initial value) 001: Medium-speed clock: ϕ/2 010: Medium-speed clock: ϕ/4 011: Medium-speed clock: ϕ/8 100: Medium-speed clock: ϕ/16 101: Medium-speed clock: ϕ/32 11x: Must not be set.

[Legend]

x: Don't care

Recommended specification

Note: * Setting prohibited.

[Legend] x: Don't care

				from the EXCL pin is sampled using the clock (φ generated by the system clock pulse generator. 0: Sampling using φ/32 clock 1: Sampling using φ/4 clock
4	EXCLE	0	R/W	Subclock Input Enable Enables/disables subclock input from the EXCL 0: Disables subclock input from the EXCL pin 1: Enables subclock input from the EXCL pin
3 to 0	—	All 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	Reserved The initial value should not be changed.

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	

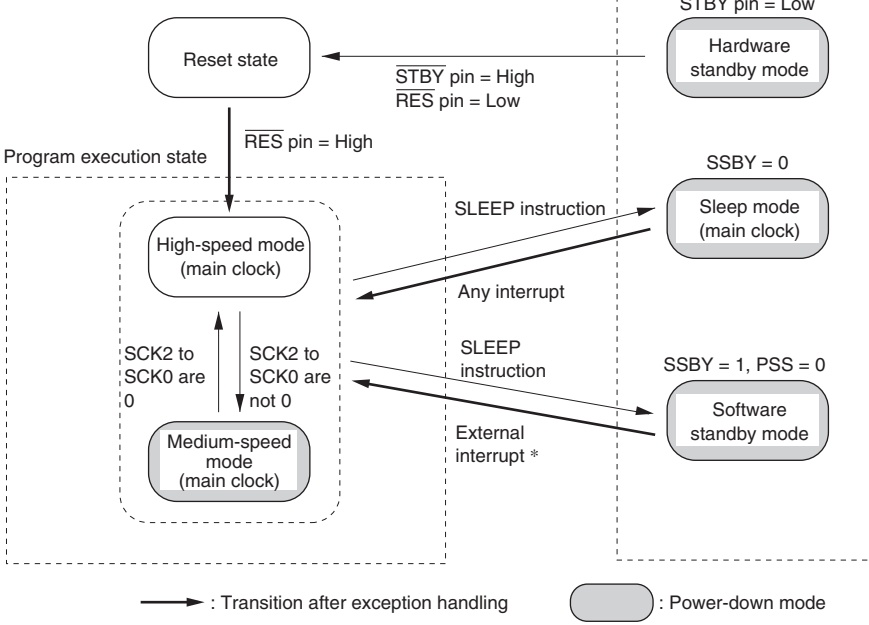
MSTPCRH (bit 3) MSTP11	MSTPCRA (bit 1) MSTPA1	Function
0	0	14-bit PWM timer (PWMX_0) operates.
0	1	14-bit PWM timer (PWMX_0) stops.
1	x	

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 2	SMSTPB7 to SMSTPB2	All 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 value 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 IRQ5

Figure 22.1 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_3					
	CRC					
	D/A converter					
	SCI_1, SCI_3			Functioning/ Halted (retained/reset)	Halted (retained/reset)	Ha
	PWMX_0, PWMX_1			Functioning/ Halted (reset)	Halted (reset)	
	A/D converter					
	RAM		Functioning (DTC)	Functioning	Retained	Re
	I/O		Functioning			Hi im

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 22.2 shows an example of medium-speed mode timing.

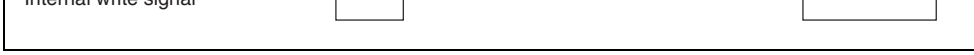


Figure 22.2 Medium-Speed Mode Timing

22.4 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 22.3 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.

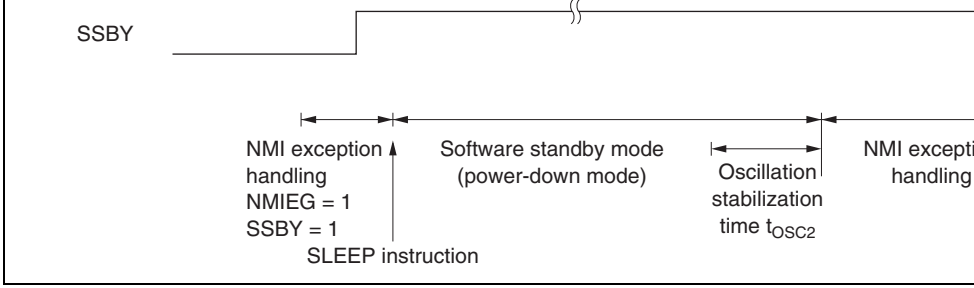


Figure 22.3 Software Standby Mode Application Example

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 22.4 shows an example of hardware standby mode timing.

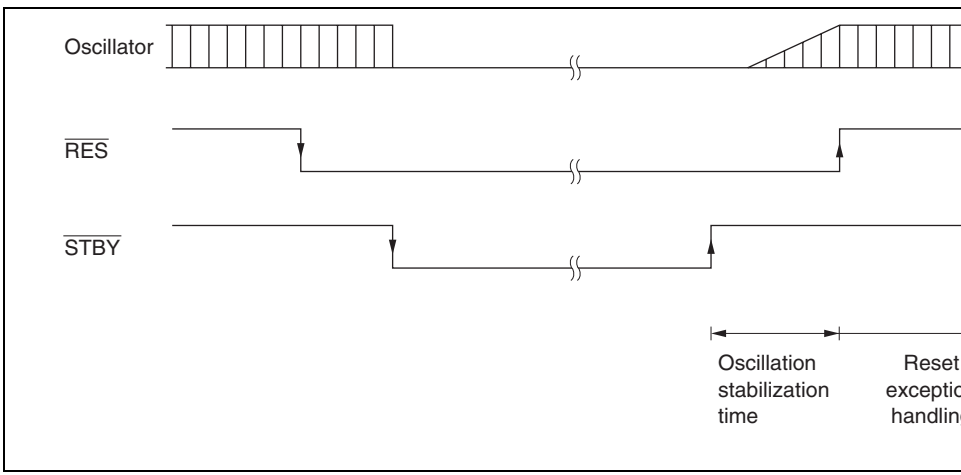


Figure 22.4 Hardware Standby Mode Timing

While an on-chip peripheral module is in module stop mode, read/write access to its regis
disabled.

The current consumption increases during oscillation settling.

22.8.3 DTC Module Stop Mode

If the DTC module stop mode specification and DTC bus request occur simultaneously,
released to the DTC and the MSTP bit cannot be set to 1. After completing the DTC bu
the MSTP bit to 1 again.

22.8.4 Notes on Subclock Usage

When using the subclock, make a transition to power-down mode after setting the EXCI
LPWRCR to 1 and loading the subclock two or more cycles. When not using the subclo
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

23.1 Register Addresses (Address Order)

The data bus width indicates the numbers of bits by which the register is accessed.

The number of access states indicates the number of states based on the specified referen

Note: Access to undefined or reserved addresses is prohibited. Since operation or cont
operation is not guaranteed when these registers are accessed, do not attempt su

SMIC flag register	SMICFLG	8	H'FD08	LPC	16
SMIC control/status register	SMICCSR	8	H'FD0A	LPC	16
SMIC data register	SMICDTR	8	H'FD0B	LPC	16
SMIC interrupt register 0	SMICIR0	8	H'FD0C	LPC	16
SMIC interrupt register 1	SMICIR1	8	H'FD0E	LPC	16
Bidirectional data register 0MW	TWR0MW	8	H'FD10	LPC	16
Bidirectional data register 0SW	TWR0SW	8	H'FD10	LPC	16
Bidirectional data register 1	TWR1	8	H'FD11	LPC	16
Bidirectional data register 2	TWR2	8	H'FD12	LPC	16
Bidirectional data register 3	TWR3	8	H'FD13	LPC	16
Bidirectional data register 4	TWR4	8	H'FD14	LPC	16
Bidirectional data register 5	TWR5	8	H'FD15	LPC	16
Bidirectional data register 6	TWR6	8	H'FD16	LPC	16
Bidirectional data register 7	TWR7	8	H'FD17	LPC	16
Bidirectional data register 8	TWR8	8	H'FD18	LPC	16
Bidirectional data register 9	TWR9	8	H'FD19	LPC	16
Bidirectional data register 10	TWR10	8	H'FD1A	LPC	16
Bidirectional data register 11	TWR11	8	H'FD1B	LPC	16
Bidirectional data register 12	TWR12	8	H'FD1C	LPC	16
Bidirectional data register 13	TWR13	8	H'FD1D	LPC	16
Bidirectional data register 14	TWR14	8	H'FD1E	LPC	16
Bidirectional data register 15	TWR15	8	H'FD1F	LPC	16
Input data register 3	IDR3	8	H'FD20	LPC	16
Output data register 3	ODR3	8	H'FD21	LPC	16

Output data register 1	ODR1	8	H'FD29	LPC	16
Status register 1	STR1	8	H'FD2A	LPC	16
SERIRQ control register 5	SIRQCR5	8	H'FD2B	LPC	16
Input data register 2	IDR2	8	H'FD2C	LPC	16
Output data register 2	ODR2	8	H'FD2D	LPC	16
Status register 2	STR2	8	H'FD2E	LPC	16
Host interface select register	HISEL	8	H'FD2F	LPC	16
Host interface control register 0	HICR0	8	H'FD30	LPC	16
Host interface control register 1	HICR1	8	H'FD31	LPC	16
Host interface control register 2	HICR2	8	H'FD32	LPC	16
Host interface control register 3	HICR3	8	H'FD33	LPC	16
SERIRQ control register2	SIRQCR2	8	H'FD34	LPC	16
BT data buffer	BTDTR	8	H'FD35	LPC	16
BT FIFO valid size register 0	BTFVSR0	8	H'FD36	LPC	16
BT FIFO valid size register 1	BTFVSR1	8	H'FD37	LPC	16
LPC channel 1, 2 address register H	LADR12H	8	H'FD38	LPC	16
LPC channel 1, 2 address register L	LADR12L	8	H'FD39	LPC	16
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	P3NCE	8	H'FE44	PORT	8

Port C data direction register	PCDDR	8	H'FE4E	PORT	8
Flash code control status register	FCCS	8	H'FE88	FLASH	8
Flash program code select register	FPCS	8	H'FE89	FLASH	8
Flash erase code select register	FECS	8	H'FE8A	FLASH	8
Flash key code register	FKEY	8	H'FE8C	FLASH	8
Flash MAT select register	FMATS	8	H'FE8D	FLASH	8
Flash transfer destination address register	FTDAR	8	H'FE8E	FLASH	8
Serial mode register_1	SMR_1	8	H'FE98	SCI_1	8
Bit rate register_1	BRR_1	8	H'FE99	SCI_1	8
Serial control register_1	SCR_1	8	H'FE9A	SCI_1	8
Transmit data register_1	TDR_1	8	H'FE9B	SCI_1	8
Serial status register_1	SSR_1	8	H'FE9C	SCI_1	8
Receive data register_1	RDR_1	8	H'FE9D	SCI_1	8
Smart card mode register_1	SCMR_1	8	H'FE9E	SCI_1	8
A/D data register A	ADDRA	16	H'FEA0	ADC	16
A/D data register B	ADDRB	16	H'FEA2	ADC	16
A/D data register C	ADDRC	16	H'FEA4	ADC	16
A/D data register D	ADDRD	16	H'FEA6	ADC	16
A/D data register E	ADDRE	16	H'FEA8	ADC	16
A/D data register F	ADDRF	16	H'FEAA	ADC	16
A/D data register G	ADDRG	16	H'FEAC	ADC	16
A/D data register H	ADDRH	16	H'FEAE	ADC	16
A/D control/status register	ADCSR	8	H'FEB0	ADC	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
PWMX (D/A) counter_1	DACNT_1	16	H'FECE	PWMX_1	8
CRC control register	CRCCR	8	H'FED4	CRC	16
CRC data input register	CRCDIR	8	H'FED5	CRC	16
CRC data output register	CRCDOR	16	H'FED6	CRC	16
I^2C bus control extended register_0	ICXR_0	8	H'FED8	IIC_0	8
I^2C bus control extended register_1	ICXR_1	8	H'FED9	IIC_1	8
I^2C SMBus control register	ICSMBCR	8	H'FEDB	IIC	8
I^2C bus control extended register_2	ICXR_2	8	H'FEDC	IIC_2	8
I^2C bus control extended register_3	ICXR_3	8	H'FEDD	IIC_3	8
I^2C bus transfer select register	IICX3	8	H'FEDF	IIC	8
Keyboard comparator control register	KBCOMP	8	H'FEE4	EVC	8

DTC enable register A	DTCERA	8	H'FEEE	DTD	8
DTC enable register B	DTCERB	8	H'FEEF	DTC	8
DTC enable register C	DTCERC	8	H'FEF0	DTC	8
DTC enable register D	DTCERD	8	H'FEF1	DTC	8
DTC enable register E	DTCERE	8	H'FEF2	DTC	8
DTC vector register	DTVECR	8	H'FEF3	DTC	8
Address break control register	ABRKCR	8	H'FEF4	INT	8
Break address register A	BARA	8	H'FEF5	INT	8
Break address register B	BARB	8	H'FEF6	INT	8
Break address register C	BARC	8	H'FEF7	INT	8
IRQ enable register 16	IER16	8	H'FEF8	INT	8
IRQ status register 16	ISR16	8	H'FEF9	INT	8
IRQ sense control register 16H	ISCR16H	8	H'FEEA	INT	8
IRQ sense control register 16L	ISCR16L	8	H'FEFB	INT	8
IRQ sense port select register 16	ISSR16	8	H'FEFC	PORT	8
IRQ sense port select register	ISSR	8	H'FEFD	PORT	8
Peripheral clock select register	PCSR	8	H'FF82	PWM	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

Output Compare register B	OCRB	16	H'FF94	FRT	16
Timer control register	TCR	8	H'FF96	FRT	16
Timer output compare control register	TOCR	8	H'FF97	FRT	16
Output Compare register AR	OCRAR	16	H'FF98	FRT	16
Output Compare register AF	OCRAF	16	H'FF9A	FRT	16
PWMX (D/A) data register A_0	DADRA_0	16	H'FFA0	PWMX_0	8
PWMX (D/A) control register_0	DACR_0	8	H'FFA0	PWMX_0	8
PWMX (D/A) data register B_0	DADRB_0	16	H'FFA6	PWMX_0	8
PWMX (D/A) counter_0	DACNT_0	16	H'FFA6	PWMX_0	8
Timer control/status register_0	TCSR_0	8	H'FFA8 (read)	WDT_0	16
Timer control/status register_0	TCSR_0	16	H'FFA8 (write)	WDT_0	16
Timer counter_0	TCNT_0	8	H'FFA9 (read)	WDT_0	16
Timer counter_0	TCNT_0	16	H'FFA8 (write)	WDT_0	16
Port A output data register	PAODR	8	H'FFAA	PORT	8
Port A input data register	PAPIN	8	H'FFAB (read)	PORT	8
Port A data direction register	PADDR	8	H'FFAB (write)	PORT	8
Port 1 pull-up MOS control register	P1PCR	8	H'FFAC	PORT	8
Port 2 pull-up MOS control register	P2PCR	8	H'FFAD	PORT	8
Port 3 pull-up MOS control register	P3PCR	8	H'FFAE	PORT	8
Port 1 data direction register	P1DDR	8	H'FFB0	PORT	8

Port 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 8 data direction register	P8DDR	8	H'FFBD	PORT	8
Port 7 input data register	P7PIN	8	H'FFBE (Read)	PORT	8
Port 8 data register	P8DR	8	H'FFBF	PORT	8
Interrupt enable register	IER	8	H'FFC2	INT	8
Serial timer control register	STCR	8	H'FFC3	SYSTEM	8
System control register	SYSCR	8	H'FFC4	SYSTEM	8
Mode control register	MDCR	8	H'FFC5	SYSTEM	8
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

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	TCSR_1	8	H'FFEA (read)	WDT_1	16
Timer control/status register_1	TCSR_1	16	H'FFEA (write)	WDT_1	16
Timer counter_1	TCNT_1	8	H'FFEB (read)	WDT_1	16
Timer counter_1	TCNT_1	16	H'FFEA (write)	WDT_1	16
Timer control register_X	TCR_X	8	H'FFF0	TMR_X	8
Timer control/status register_X	TCSR_X	8	H'FFF1	TMR_X	8
Timer counter_X	TCNT_X	8	H'FFF4	TMR_X	8
Time constant register A_X	TCORA_X	8	H'FFF6	TMR_X	8
Time constant register B_X	TCORB_X	8	H'FFF7	TMR_Y	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

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
SMICCSR	bit 7	bit 6	bit 5	bit 4	bit 3	bit 2	bit 1	bit 0
SMICDTR	bit 7	bit 6	bit 5	bit 4	bit 3	bit 2	bit 1	bit 0
SMICIR0	—	—	—	HDTWI	HDTRI	STARI	CTLWI	BUSYI
SMICIR1	—	—	—	HDTWIE	HDTRIE	STARIE	CTLWIE	BUSYIE
TWR0MW	bit 7	bit 6	bit 5	bit 4	bit 3	bit 2	bit 1	bit 0
TWR0SW	bit 7	bit 6	bit 5	bit 4	bit 3	bit 2	bit 1	bit 0
TWR1	bit 7	bit 6	bit 5	bit 4	bit 3	bit 2	bit 1	bit 0
TWR2	bit 7	bit 6	bit 5	bit 4	bit 3	bit 2	bit 1	bit 0
TWR3	bit 7	bit 6	bit 5	bit 4	bit 3	bit 2	bit 1	bit 0
TWR4	bit 7	bit 6	bit 5	bit 4	bit 3	bit 2	bit 1	bit 0
TWR5	bit 7	bit 6	bit 5	bit 4	bit 3	bit 2	bit 1	bit 0
TWR6	bit 7	bit 6	bit 5	bit 4	bit 3	bit 2	bit 1	bit 0
TWR7	bit 7	bit 6	bit 5	bit 4	bit 3	bit 2	bit 1	bit 0
TWR8	bit 7	bit 6	bit 5	bit 4	bit 3	bit 2	bit 1	bit 0
TWR9	bit 7	bit 6	bit 5	bit 4	bit 3	bit 2	bit 1	bit 0
TWR10	bit 7	bit 6	bit 5	bit 4	bit 3	bit 2	bit 1	bit 0

STR3*1	IBF3B	OBF3B	MWMF	SWMF	C/D3	DBU32	IBF3A	OBF3A
STR3*2	DBU37	DBU36	DBU35	DBU34	C/$\overline{D}$3	DBU32	IBF3A	OBF3A
SIRQCR4	IRQ15E	IRQ14E	IRQ13E	IRQ8E	IRQ7E	IRQ5E	IRQ4E	IRQ3E
LADR3H	bit 15	bit 14	bit 13	bit 12	bit 11	bit 10	bit 9	bit 8
LADR3L	bit 7	bit 6	bit 5	bit 4	bit 3	—	bit 1	TWRE
SIRQCR0	Q/$\overline{C}$	SELREQ	IEDIR2	SMIE3B	SMIE3A	SMIE2	IRQ12E0	IRQ1E0
SIRQCR1	IRQ11E3	IRQ10E3	IRQ9E3	IRQ6E3	IRQ11E2	IRQ10E2	IRQ9E2	IRQ6E2
IDR1	bit7	bit6	bit5	bit4	bit3	bit2	bit1	bit0
ODR1	bit7	bit6	bit5	bit4	bit3	bit2	bit1	bit0
STR1	DBU17	DBU16	DBU15	DBU14	C/$\overline{D}$0	DBU12	IBF1	OBF1
SIRQCR5	SELIRQ15	SELIRQ14	SELIRQ13	SELIRQ8	SELIRQ7	SELIRQ5	SELIRQ4	SELIRQ3
IDR2	bit7	bit6	bit5	bit4	bit3	bit2	bit1	bit0
ODR2	bit7	bit6	bit5	bit4	bit3	bit2	bit1	bit0
STR2	DBU27	DBU26	DBU25	DBU24	C/$\overline{D}$2	DBU22	IBF2	OBF2
HISEL	SELSTR3	SELIRQ11	SELIRQ10	SELIRQ9	SELIRQ6	SELSMI	SELIRQ12	SELIRQ1
HICR0	—	LPC2E	LPC1E	—	SDWNE	—	—	—
HICR1	LPCBSY	CLKREQ	IRQBSY	LRSTB	SDWNB	—	—	—
HICR2	—	LRST	—	ABRT	—	IBFIE2	IBFIE1	ERRIE
HICR3	LFRAME	—	SERIRQ	LRESET	—	—	—	—
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	bit15	bit14	bit13	bit12	bit11	bit10	bit9	bit8
LADR12L	bit7	bit6	bit5	bit4	bit3	—	bit1	bit0
SUBMSTPBH	SMSTPB15	SMSTPB14	SMSTPB13	SMSTPB12	SMSTPB11	SMSTPB10	SMSTPB9	SMSTPB8
SUBMSTPBL	SMSTPB7	SMSTPB6	SMSTPB5	SMSTPB4	SMSTPB3	SMSTPB2	SMSTPB1	SMSTPB0

PEPIN	PE7PIN	PE6PIN	PE5PIN	PE4PIN	PE3PIN	PE2PIN	PE1PIN	PE0PIN
PEDDR	PE7DDR	PE6DDR	PE5DDR	PE4DDR	PE3DDR	PE2DDR	PE1DDR	PE0DDR
PCODR	PC7ODR	PC6ODR	PC5ODR	PC4ODR	PC3ODR	PC2ODR	PC1ODR	PC0ODR
PCPIN	PC7PIN	PC6PIN	PC5PIN	PC4PIN	PC3PIN	PC2PIN	PC1PIN	PC0PIN
PCDDR	PC7DDR	PC6DDR	PC5DDR	PC4DDR	PC3DDR	PC2DDR	PC1DDR	PC0DDR
FCCS	FWE	—	—	FLER	WEINTE	—	—	SCO
FPCS	—	—	—	—	—	—	—	PPVS
FECS	—	—	—	—	—	—	—	EPVB
FKEY	K7	K6	K5	K4	K3	K2	K1	K0
FMATS	MS7	MS6	MS5	MS4	MS3	MS2	MS1	MS0
FTDAR	TDER	TDA6	TDA5	TDA4	TDA3	TDA2	TDA1	TDA0
SMR_1*	C/$\overline{A}$ (GM)	CHR (BLK)	PE (PE)	O/$\overline{E}$ (O/$\overline{E}$)	STOP (BCP1)	MP (BCP0)	CKS1 (CKS1)	CKS0 (CKS0)
BRR_1	bit7	bit6	bit5	bit4	bit3	bit2	bit1	bit0
SCR_1	TIE	RIE	TE	RE	MPIE	TEIE	CKE1	CKE0
TDR_1	bit7	bit6	bit5	bit4	bit3	bit2	bit1	bit0
SSR_1*	TDRE (TDRE)	RDRF (RDRF)	ORER (ORER)	FER (ERS)	PER (PER)	TEND (TEND)	MPB (MPB)	MPBT (MPBT)
RDR_1	bit7	bit6	bit5	bit4	bit3	bit2	bit1	bit0
SCMR_1	—	—	—	—	SDIR	SINV	—	SMIF
ADDRA	AD9	AD8	AD7	AD6	AD5	AD4	AD3	AD2
	AD1	AD0	—	—	—	—	—	—
ADDRB	AD9	AD8	AD7	AD6	AD5	AD4	AD3	AD2
	AD1	AD0	—	—	—	—	—	—
ADDRC	AD9	AD8	AD7	AD6	AD5	AD4	AD3	AD2
	AD1	AD0	—	—	—	—	—	—
ADDRD	AD9	AD8	AD7	AD6	AD5	AD4	AD3	AD2
	AD1	AD0	—	—	—	—	—	—

ADCR	ADF	ADIE	ADST	—	—	CH2	CH1	CH0
ADCR	TRGS1	TRGS0	SCANE	SCANS	CKS1	CKS0	ADSTCLR	EXTRGS
P4NCE	P47NCE	P46NCE	P45NCE	P44NCE	—	—	—	—
P4NCMC	P47NCMC	P46NCMC	P45NCMC	P44NCMC	—	—	—	—
P6PCR	—	—	—	—	P63PCR	P62PCR	P61PCR	P60PCR
P4PCR	P47PCR	P46PCR	P45PCR	P44PCR	—	—	—	—
ICCR_3	ICE	IEIC	MST	TRS	ACKE	BBSY	IRIC	SCP
ICSR_3	ESTP	STOP	IRTR	AASX	AL	AAS	ADZ	ACKB
ICDR_3	bit7	bit6	bit5	bit4	bit3	bit2	bit1	bit0
SARX_3	SVAX6	SVAX5	SVAX4	SVAX3	SVAX2	SVAX1	SVAX0	FSX
ICMR_3	MLS	WAIT	CKS2	CKS1	CKS0	BC2	BC1	BC0
SAR_3	SVA6	SVA5	SVA4	SVA3	SVA2	SVA1	SVA0	FS
ICCR_2	ICE	IEIC	MST	TRS	ACKE	BBSY	IRIC	SCP
ICSR_2	ESTP	STOP	IRTR	AASX	AL	AAS	ADZ	ACKB
ICDR_2	bit7	bit6	bit5	bit4	bit3	bit2	bit1	bit0
SARX_2	SVAX6	SVAX5	SVAX4	SVAX3	SVAX2	SVAX1	SVAX0	FSX
ICMR_2	MLS	WAIT	CKS2	CKS1	CKS0	BC2	BC1	BC0
SAR_2	SVA6	SVA5	SVA4	SVA3	SVA2	SVA1	SVA0	FS
DADRA_1	DA13	DA12	DA11	DA10	DA9	DA8	DA7	DA6
	DA5	DA4	DA3	DA2	DA1	DA0	CFS	—
DACR_1	—	PWME	—	—	OEB	OEA	OS	CKS
DADRB_1	DA13	DA12	DA11	DA10	DA9	DA8	DA7	DA6
	DA5	DA4	DA3	DA2	DA1	DA0	CFS	REGS
DACNT_1	UC7	UC6	UC5	UC4	UC3	UC2	UC1	UC0
	UC8	UC9	UC10	UC11	UC12	UC13	—	REGS
CRCCR	DORCLR	—	—	—	—	LMS	G0	G0
CRCDIR	bit7	bit6	bit5	bit4	bit3	bit2	bit1	bit0

RDCOMF	EVENTE	—	—	—	—	—	—	—
ICRD	ICRD7	ICRD6	—	—	—	—	—	—
ICRA	ICRA7	ICRA6	ICRA5	ICRA4	ICRA3	ICRA2	ICRA1	ICRA0
ICRB	ICRB7	ICRB6	—	ICRB4	ICRB3	ICRB2	ICRB1	—
ICRC	ICRC7	ICRC6	—	ICRC4	ICRC3	ICRC2	ICRC1	—
ISR	IRQ7F	IRQ6F	IRQ5F	IRQ4F	IRQ3F	IRQ2F	IRQ1F	IRQ0F
ISCRH	IRQ7SCB	IRQ7SCA	IRQ6SCB	IRQ6SCA	IRQ5SCB	IRQ5SCA	IRQ4SCB	IRQ4SCA
ISCRL	IRQ3SCB	IRQ3SCA	IRQ2SCB	IRQ2SCA	IRQ1SCB	IRQ1SCA	IRQ0SCB	IRQ0SCA
DTCERA	DTCEA7	DTCEA6	DTCEA5	DTCEA4	DTCEA3	—	—	—
DTCERB	—	DTCEB6	DTCEB5	—	—	—	—	—
DTCERC	—	—	—	DTCEC4	—	DTCEC2	DTCEC1	DTCEC0
DTCERD	DTCED7	—	—	DTCED4	DTCED3	—	—	—
DTCERE	—	—	—	—	DTCEE3	DTCEE2	DTCEE1	—
DTVECR	SWDTE	DTVEC6	DTVEC5	DTVEC4	DTVEC3	DTVEC2	DTVEC1	DTVEC0
ABRKCR	CMF	—	—	—	—	—	—	BIE
BARA	A23	A22	A21	A20	A19	A18	A17	A16
BARB	A15	A14	A13	A12	A11	A10	A9	A8
BARC	A7	A6	A5	A4	A3	A2	A1	—
IER16	IRQ15E	IRQ14E	IRQ13E	IRQ12E	IRQ11E	IRQ10E	IRQ9E	IRQ8E
ISR16	IRQ15F	IRQ14F	IRQ13F	IRQ12F	IRQ11F	IRQ10F	IRQ9F	IRQ8F
ISCR16H	IRQ15SCB	IRQ15SCA	IRQ14SCB	IRQ14SCA	IRQ13SCB	IRQ13SCA	IRQ12SCB	IRQ12SCA
ISCR16L	IRQ11SCB	IRQ11SCA	IRQ10SCB	IRQ10SCA	IRQ9SCB	IRQ9SCA	IRQ8SCB	IRQ8SCA
ISSR16	ISS15	ISS14	ISS13	ISS12	ISS11	ISS10	ISS9	ISS8
ISSR	ISS7	ISS6	ISS5	ISS4	ISS3	ISS2	ISSR1	ISS0
PCSR	PWCKX1B	PWCKX1A	PWCKX0B	PWCKX0A	PWCKX1C	PWCKB	PWCKA	PWCKX0C
SBYCR	SSBY	STS2	STS1	STS0	DTSPEED	SCK2	SCK1	SCK0
LPWRCR	—	—	NESEL	EXCLE	—	—	—	—

TIER	—	—	—	—	OCIAE	OCIBE	OVIE	—
TCSR	—	—	—	—	OCFA	OCFB	OVF	CCLRA
FRC	bit15	bit14	bit13	bit12	bit11	bit10	bit9	bit8
	bit7	bit6	bit5	bit4	bit3	bit2	bit1	bit0
OCRA	bit15	bit14	bit13	bit12	bit11	bit10	bit9	bit8
	bit7	bit6	bit5	bit4	bit3	bit2	bit1	bit0
OCRB	bit15	bit14	bit13	bit12	bit11	bit10	bit9	bit8
	bit7	bit6	bit5	bit4	bit3	bit2	bit1	bit0
TCR	—	—	—	—	—	—	CKS1	CKS0
TOCR	—	OCRAMS	ICRS	OCRS	—	—	—	—
OCRAR	bit15	bit14	bit13	bit12	bit11	bit10	bit9	bit8
	bit7	bit6	bit5	bit4	bit3	bit2	bit1	bit0
OCRAF	bit15	bit14	bit13	bit12	bit11	bit10	bit9	bit8
	bit7	bit6	bit5	bit4	bit3	bit2	bit1	bit0
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	bit7	bit6	bit5	bit4	bit3	bit2	bit1	bit0
PAODR	PA7ODR	PA6ODR	PA5ODR	PA4ODR	PA3ODR	PA2ODR	PA1ODR	PA0ODR
PAPIN	PA7PIN	PA6PIN	PA5PIN	PA4PIN	PA3PIN	PA2PIN	PA1PIN	PA0PIN
PADDR	PA7DDR	PA6DDR	PA5DDR	PA4DDR	PA3DDR	PA2DDR	PA1DDR	PA0DDR
P1PCR	P17PCR	P16PCR	P15PCR	P14PCR	P13PCR	P12PCR	P11PCR	P10PCR
P2PCR	—	—	—	—	P23PCR	P22PCR	P21PCR	P20PCR

P4DR	P47DR	P46DR	P45DR	P44DR	P43DR	P42DR	P41DR	P40DR
P5DDR	P57DDR	P56DDR	—	—	P53DDR	P52DDR	—	—
P6DDR	—	—	—	—	P63DDR	P62DDR	P61DDR	P60DDR
P5DR	P57DR	P56DR	—	—	P53DR	P52DR	—	—
P6DR	—	—	—	—	P63DR	P62DR	P61DR	P60DR
P8DDR	P87DDR	P86DDR	P85DDR	P84DDR	P83DDR	P82DDR	P81DDR	P80DDR
P7PIN	P77PIN	P76PIN	P75PIN	P74PIN	P73PIN	P72PIN	P71PIN	P70PIN
P8DR	P87DR	P86DR	P85DR	P84DR	P83DR	P82DR	P81DR	P80DR
IER	IRQ7E	IRQ6E	IRQ5E	IRQ4E	IRQ3E	IRQ2E	IRQ1E	IRQ0E
STCR	IICX2	IICX1	IICX0	—	FLSHE	—	ICKS1	ICKS0
SYSCR	—	—	INTM1	INTM0	XRST	NMIEG	—	RAME
MDCR	—	—	—	—	—	MDS2	MDS1	—
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	bit7	bit6	bit5	bit4	bit3	bit2	bit1	bit0
TCORA_1	bit7	bit6	bit5	bit4	bit3	bit2	bit1	bit0
TCORB_0	bit7	bit6	bit5	bit4	bit3	bit2	bit1	bit0
TCORB_1	bit7	bit6	bit5	bit4	bit3	bit2	bit1	bit0
TCNT_0	Bit7	bit6	bit5	bit4	bit3	bit2	bit1	bit0
TCNT_1	bit7	bit6	bit5	bit4	bit3	bit2	bit1	bit0
ICCR_0	ICE	IEIC	MST	TRS	ACKE	BBSY	IRIC	SCP
ICSR_0	ESTP	STOP	IRTR	AASX	AL	AAS	ADZ	ACKB
ICDR_0	bit7	bit6	bit5	bit4	bit3	bit2	bit1	bit0
SARX_0	SVAX6	SVAX5	SVAX4	SVAX3	SVAX2	SVAX1	SVAX0	FSX
ICMR_0	MLS	WAIT	CKS2	CKS1	CKS0	BC2	BC1	BC0
SAR_0	SVA6	SVA5	SVA4	SVA3	SVA2	SVA1	SVA0	FS

TCSR_1	OVF	WT/$\overline{\text{IT}}$	TME	PSS	RST/$\overline{\text{NMI}}$	CKS2	CKS1	CKS0
TCNT_1	bit7	bit6	bit5	bit4	bit3	bit2	bit1	bit0
TCR_X	CMIEB	CMIEA	OVIE	CCLR1	CCLR0	CKS2	CKS1	CKS0
TCSR_X	CMFB	CMFA	OVF	—	—	—	—	—
TCNT_X	bit7	bit6	bit5	bit4	bit3	bit2	bit1	bit0
TCORA_X	bit7	bit6	bit5	bit4	bit3	bit2	bit1	bit0
TCORB_X	bit7	bit6	bit5	bit4	bit3	bit2	bit1	bit0
TCR_Y	CMIEB	CMIEA	OVIE	CCLR1	CCLR0	CKS2	CKS1	CKS0
TCSR_Y	CMFB	CMFA	OVF	—	—	—	—	—
TCORA_Y	bit7	bit6	bit5	bit4	bit3	bit2	bit1	bit0
TCORB_Y	bit7	bit6	bit5	bit4	bit3	bit2	bit1	bit0
TCNT_Y	bit7	bit6	bit5	bit4	bit3	bit2	bit1	bit0
TCONRS	TMRX/Y	—	—	—	—	—	—	—

Note: Some bits have different names in normal mode and smart card interface mode. T
name in smart card interface mode is enclosed in parentheses.

BTCR	Initialized	Initialized	—	—	—	—	Initialized
BTIMSR	Initialized	Initialized	—	—	—	—	Initialized
SMICFLG	Initialized	Initialized	—	—	—	—	Initialized
SMICCSR	—	—	—	—	—	—	—
SMICDTR	—	—	—	—	—	—	—
SMICIR0	Initialized	Initialized	—	—	—	—	Initialized
SMICIR1	Initialized	Initialized	—	—	—	—	Initialized
TWR0MW	—	—	—	—	—	—	—
TWR0SW	—	—	—	—	—	—	—
TWR1	—	—	—	—	—	—	—
TWR2	—	—	—	—	—	—	—
TWR3	—	—	—	—	—	—	—
TWR4	—	—	—	—	—	—	—
TWR5	—	—	—	—	—	—	—
TWR6	—	—	—	—	—	—	—
TWR7	—	—	—	—	—	—	—
TWR8	—	—	—	—	—	—	—
TWR9	—	—	—	—	—	—	—
TWR10	—	—	—	—	—	—	—
TWR11	—	—	—	—	—	—	—
TWR12	—	—	—	—	—	—	—
TWR13	—	—	—	—	—	—	—
TWR14	—	—	—	—	—	—	—

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	
HICR0	Initialized	Initialized	—	—	—	—	Initialized	
HICR1	Initialized	Initialized	—	—	—	—	Initialized	
HICR2	Initialized	Initialized	—	—	—	—	Initialized	
HICR3	—	—	—	—	—	—	—	
SIRQCR2	Initialized	Initialized	—	—	—	—	Initialized	
BTDTR	—	—	—	—	—	—	—	
BTFVSR0	Initialized	Initialized	—	—	—	—	Initialized	
BTFVSR1	Initialized	Initialized	—	—	—	—	Initialized	
LADR12H	Initialized	Initialized	—	—	—	—	Initialized	
LADR12L	Initialized	Initialized	—	—	—	—	Initialized	
SUBMSTPBH	Initialized	Initialized	—	—	—	—	Initialized	SY
SUBMSTPBL	Initialized	Initialized	—	—	—	—	Initialized	
ECS	Initialized	Initialized	—	—	—	—	Initialized	EV
ECCR	Initialized	Initialized	—	—	—	—	Initialized	
MSTPCRA	Initialized	Initialized	—	—	—	—	Initialized	SY
P3NCE	Initialized	Initialized	—	—	—	—	Initialized	PO
P3NCMC	Initialized	Initialized	—	—	—	—	Initialized	

FPCS	Initialized	Initialized	—	—	—	—	Initialized	
FECS	Initialized	Initialized	—	—	—	—	Initialized	
FKEY	Initialized	Initialized	—	—	—	—	Initialized	
FMATS	Initialized	Initialized	—	—	—	—	Initialized	
FTDAR	Initialized	Initialized	—	—	—	—	Initialized	
SMR_1	Initialized	Initialized	—	—	—	—	Initialized	SCI
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	ADC
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	
ADDRH	Initialized	Initialized	—	—	Initialized	Initialized	Initialized	
ADCSR	Initialized	Initialized	—	—	Initialized	Initialized	Initialized	
ADCR	Initialized	Initialized	—	—	Initialized	Initialized	Initialized	
P4NCE	Initialized	Initialized	—	—	—	—	Initialized	POR
P4NCMC	Initialized	Initialized	—	—	—	—	Initialized	
P6PCR	Initialized	Initialized	—	—	Initialized	Initialized	Initialized	
P4PCR	Initialized	Initialized	—	—	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	PV
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	CR
CRCDIR	Initialized	Initialized	—	—	—	—	Initialized	
CRCDOR	Initialized	Initialized	—	—	—	—	Initialized	
ICXR_0	Initialized	Initialized	—	—	—	—	Initialized	IIC
ICXR_1	Initialized	Initialized	—	—	—	—	Initialized	IIC
ICSMBCR	Initialized	Initialized	—	—	—	—	Initialized	IIC
ICXR_2	Initialized	Initialized	—	—	—	—	Initialized	IIC
ICXR_3	Initialized	Initialized	—	—	—	—	Initialized	IIC
IICX3	Initialized	Initialized	—	—	—	—	Initialized	IIC
KBCOMP	Initialized	Initialized	—	—	—	—	Initialized	EV
ICRD	Initialized	Initialized	—	—	—	—	Initialized	IN
ICRA	Initialized	Initialized	—	—	—	—	Initialized	
ICRB	Initialized	Initialized	—	—	—	—	Initialized	
ICRC	Initialized	Initialized	—	—	—	—	Initialized	
ISR	Initialized	Initialized	—	—	—	—	Initialized	
ISCRH	Initialized	Initialized	—	—	—	—	Initialized	
ISCRL	Initialized	Initialized	—	—	—	—	Initialized	
DTCERA	Initialized	Initialized	—	—	—	—	Initialized	DT
DTCERB	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	
PCSR	Initialized	Initialized	—	—	—	—	Initialized	SYS
SBYCR	Initialized	Initialized	—	—	—	—	Initialized	
LPWRCR	Initialized	Initialized	—	—	—	—	Initialized	
MSTPCRH	Initialized	Initialized	—	—	—	—	Initialized	
MSTPCRL	Initialized	Initialized	—	—	—	—	Initialized	
ICCR_1	Initialized	Initialized	—	—	—	—	Initialized	IIC_
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	FRT
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	

PADDR	Initialized	Initialized	—	—	—	—	Initialized	
P1PCR	Initialized	Initialized	—	—	—	—	Initialized	
P2PCR	Initialized	Initialized	—	—	—	—	Initialized	
P3PCR	Initialized	Initialized	—	—	—	—	Initialized	
P1DDR	Initialized	Initialized	—	—	—	—	Initialized	
P2DDR	Initialized	Initialized	—	—	—	—	Initialized	
P1DR	Initialized	Initialized	—	—	—	—	Initialized	
P2DR	Initialized	Initialized	—	—	—	—	Initialized	
P3DDR	Initialized	Initialized	—	—	—	—	Initialized	
P4DDR	Initialized	—	—	—	—	—	Initialized	
P3DR	Initialized	Initialized	—	—	—	—	Initialized	
P4DR	Initialized	—	—	—	—	—	Initialized	
P5DDR	Initialized	Initialized	—	—	—	—	Initialized	
P6DDR	Initialized	Initialized	—	—	—	—	Initialized	
P5DR	Initialized	Initialized	—	—	—	—	Initialized	
P6DR	Initialized	Initialized	—	—	—	—	Initialized	
P8DDR	Initialized	Initialized	—	—	—	—	Initialized	
P7PIN	—	—	—	—	—	—	—	
P8DR	Initialized	Initialized	—	—	—	—	Initialized	
IER	Initialized	Initialized	—	—	—	—	Initialized	IN
STCR	Initialized	Initialized	—	—	—	—	Initialized	SY
SYSCR	Initialized	Initialized	—	—	—	—	Initialized	
MDCR	Initialized	Initialized	—	—	—	—	Initialized	
TCR_0	Initialized	Initialized	—	—	—	—	Initialized	TM
TCR_1	Initialized	Initialized	—	—	—	—	Initialized	TM
TCSR_0	Initialized	Initialized	—	—	—	—	Initialized	TM
TCSR_1	Initialized	Initialized	—	—	—	—	Initialized	TM

ICDR_0	—	—	—	—	—	—	—	IIC_
SARX_0	Initialized	Initialized	—	—	—	—	Initialized	
ICMR_0	Initialized	Initialized	—	—	—	—	Initialized	
SAR_0	Initialized	Initialized	—	—	—	—	Initialized	
SMR_3	Initialized	Initialized	—	—	—	—	Initialized	SCI_
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	WDT
TCNT_1	Initialized	Initialized	—	—	—	—	Initialized	
TCR_X	Initialized	Initialized	—	—	—	—	Initialized	TMR
TCSR_X	Initialized	Initialized	—	—	—	—	Initialized	
TCNT_X	Initialized	Initialized	—	—	—	—	Initialized	
TCORA_X	Initialized	Initialized	—	—	—	—	Initialized	
TCORB_X	Initialized	Initialized	—	—	—	—	Initialized	TMR
TCR_Y	Initialized	Initialized	—	—	—	—	Initialized	
TCSR_Y	Initialized	Initialized	—	—	—	—	Initialized	
TCORA_Y	Initialized	Initialized	—	—	—	—	Initialized	
TCORB_Y	Initialized	Initialized	—	—	—	—	Initialized	
TCNT_Y	Initialized	Initialized	—	—	—	—	Initialized	
TCONRS	Initialized	Initialized	—	—	—	—	Initialized	TMR

multiplexed with analog input)			
Input voltage (pins multiplexed with IIC functions)	(1)	V_{in}	–0.3 to AVCC +0.3
Input voltage	(2)	V_{in}	–0.3 to +6.5
Input voltage (except (1), (2))		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 (AN0 to AN7)		V_{AN}	–0.3 to AVCC +0.3
Operating temperature		T_{opr}	–40 to +85
Operating temperature (when flash memory is programmed or erased)		T_{opr}	0 to +75
Storage temperature		T_{stg}	–55 to +125

Caution: Permanent damage to this LSI may result if absolute maximum ratings are exce

Note: * Voltage applied to the VCC pin.
Make sure power is not applied to the VCL pin.

Item			Symbol	Min.	Typ.	Max.	Unit	C
Schmitt trigger input voltage	DB7 to DB4, ExDB7 to ExDB0, EVENT7 to EVENT0, $\overline{\text{(Ex)IRQ15}}$, $\overline{\text{(Ex)IRQ14}}$, $\overline{\text{ExIRQ13}}$, $\overline{\text{ExIRQ12}}$, $\overline{\text{(Ex)IRQ11}}$, $\overline{\text{(Ex)IRQ10}}$, $\overline{\text{ExIRQ9}}$, $\overline{\text{ExIRQ8}}$, $\overline{\text{(Ex)IRQ7}}$ to $\overline{\text{(Ex)IRQ0}}$, $\overline{\text{ETRST}}$, XTAL, EXCL, $\overline{\text{ADTRG}}$	(1)	V_T^-	VCC × 0.2	—	—	V	
			V_T^+	—	—	VCC × 0.7		
			$V_T^+ - V_T^-$	VCC × 0.05	—	—		
	SCL3 to SCL0, SDA3 to SDA0		V_T^-	VCC × 0.3	—	—		
			V_T^+	—	—	VCC × 0.7		
			$V_T^+ - V_T^-$	VCC × 0.05	—	—		
Input high voltage	$\overline{\text{RES}}$, $\overline{\text{STBY}}$, NMI, FWE, $\overline{\text{MD2}}$, MD1	(2)	V_{IH}	VCC × 0.9	—	VCC + 0.3	V	
	EXTAL			VCC × 0.7	—	VCC + 0.3		
	Port 7			2.2	—	AVCC + 0.3		
	SCL3 to SCL0, SDA3 to SDA0, Port 80 to 83, C0 to C3			—	—	5.5		
	SERIRQ, LAD3 to LAD0, LCLK, $\overline{\text{LRESET}}$, $\overline{\text{LFRAME}}$			VCC × 0.5	—	VCC + 0.3		
				2.0	—	VCC + 0.3		
	Input pins other than (1) and (2) above			2.2	—	VCC + 0.3		

high voltage	SDA0*[2]							
	Port 80 to 83, C0 to C3*[3]			0.5	—	—		I_{O}
	SERIRQ, LAD3 to LAD0			VCC × 0.9	—	—		I_{O}
	Output pins other than (4) above			VCC – 0.5	—	—		I_{O}
				VCC – 1.0	—	—		I_{O}
Output low voltage	SCL3 to SCL0, SDA3 to SDA0	(5)	V_{OL}	—	—	0.5	V	I_{O}
				—	—	0.4		I_{O}
	SERIRQ, LAD3 to LAD0			—	—	VCC × 0.1		I_{O}
	Output pins other than (5) above			—	—	0.4		I_{O}
	HC7 to HC0			—	—	1.0		I_{O}

leakage current (off state)	Ports 8 to E						
Input pull-up MOS current	Ports 1 to 4, 6, A	$-I_P$	20	—	300		V_{IN} = 0 V
Supply current*[4]	Normal operation	I_{CC}	—	45	60	mA	f = 25 MHz, high-spe All modules operatin
	Sleep mode		—	35	45		f = 25 MHz
	Standby mode*[5]		—	40	100	μA	Ta ≤ 50 °C
			—	—	250		50 °C < Ta
Analog power supply current	During A/D conversion	AI_{cc}	—	1.0	2.0	mA	
	A/D conversion standby		—	2.5	5.0	μA	
Reference power supply current	During A/D conversion	AI_{ref}	—	0.1	1.0	mA	
	A/D conversion standby		—	0.5	5.0	μA	
Input capacitance	All input pin	C_{in}	—	—	10	pF	V_{in} = 0 V, f = 1 MHz, T_a = 25 °C
RAM standby voltage		V_{RAM}	3.0	—	—	V	
VCC start voltage		VCC_{START}	—	0	0.8	V	
VCC rising edge		SVCC	—	—	20	ms/V	

Notes: 1. Do not leave the AVCC, AVref, and AVSS pins open even if the A/D converter or D/A co not used.

Even if the A/D converter or D/A converter is not used, apply a value in the range from 3 V to the AVCC and AVref pins by connecting them to the power supply (VCC). The relat between these two pins should be AVref ≤ AVCC.

2. An external pull-up resistor is necessary to provide high-level output from SCL3 to SCL0 SDA3 to SDA0 (ICE bit in ICCR is 1).

Item		Symbol	Min.	Typ.	Max.	Un
Permissible output low current (per pin)	SCL3 to SCL0, SDA3 to SDA0	I_{OL}	—	—	10	mA
	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 24.3.

2. When driving a Darlington transistor or LED, always insert a current-limiting resistor in t line, as show in figures 24.1 and 24.2.

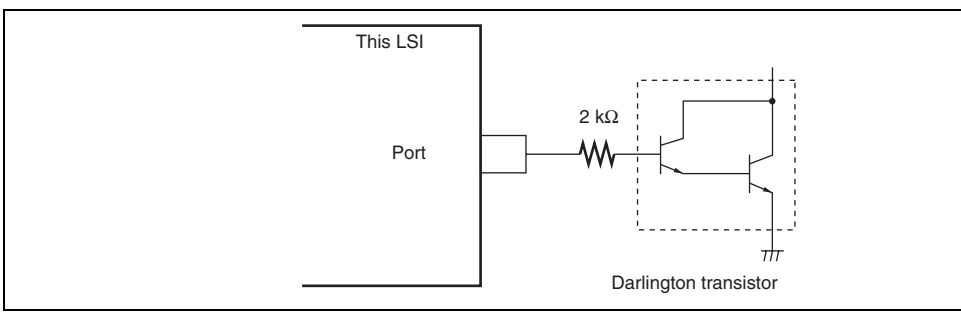


Figure 24.1 Darlington Transistor Drive Circuit (Example)

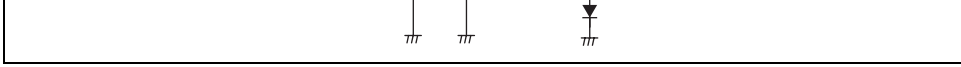


Figure 24.3 Output Load Circuit

24.3.1 Clock Timing

Table 24.4 shows the clock timing. The clock timing specified here covers clock output
clock pulse generator (crystal) and external clock input (EXTAL pin) oscillation stabiliz
times. For details of external clock input (EXTAL pin and EXCL pin) timing, see table
24.6.

Table 24.4 Clock Timing

Condition: VCC = 3.0 V to 3.6 V, VSS = 0 V, ϕ = 20 MHz to 25 MHz

Item	Symbol	Min.	Max.	Unit	Re
Clock cycle time	t_{cyc}	40	50	ns	Fig
Clock high level pulse width	t_{CH}	10	—		
Clock low level pulse width	t_{CL}	10	—		
Clock rise time	t_{Cr}	—	5		
Clock fall time	t_{Cf}	—	5		
Reset oscillation stabilization (crystal)	t_{OSC1}	10	—	ms	Fig
Software standby oscillation stabilization time (crystal)	t_{OSC2}	8	—		Fig

External clock input falling time	t_{EXf}	—	5	ns	
Clock low level pulse width	t_{CL}	0.4	0.6	t_{cyc}	Figu
Clock high level pulse width	t_{CH}	0.4	0.6	t_{cyc}	
External clock output stabilization delay time	t_{DEXT}*	500	—	μs	Figu

Note: * t_{DEXT} includes a $\overline{RES}$ pulse width (t_{RESW}).

Table 24.6 Subclock Input Conditions

Condition: VCC = 3.0 V to 3.6 V, VSS = 0 V, ϕ = 20 MHz to 25 MHz

Item	Symbol	Min.	Typ.	Max.	Unit	Tes Con
Subclock input low level pulse width	t_{EXCLL}	—	15.26	—	μs	Figu
Subclock input high level pulse width	t_{EXCLH}	—	15.26	—	μs	
Subclock input rising time	t_{EXCLr}	—	—	10	ns	
Subclock input falling time	t_{EXCLf}	—	—	10	ns	
Clock low level pulse width	t_{CL}	0.4	—	0.6	t_{cyc}	Figu
Clock high level pulse width	t_{CH}	0.4	—	0.6	t_{cyc}	

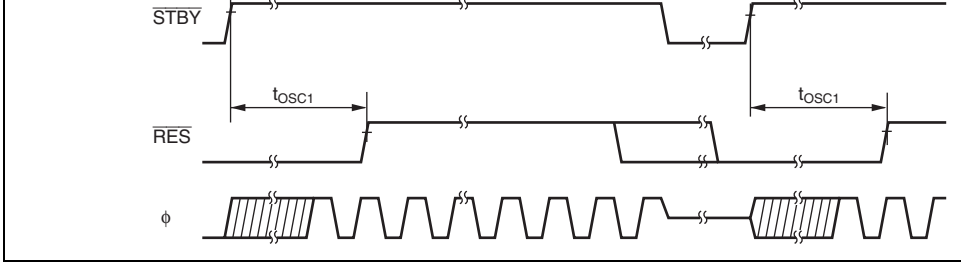


Figure 24.5 Oscillation Stabilization Timing

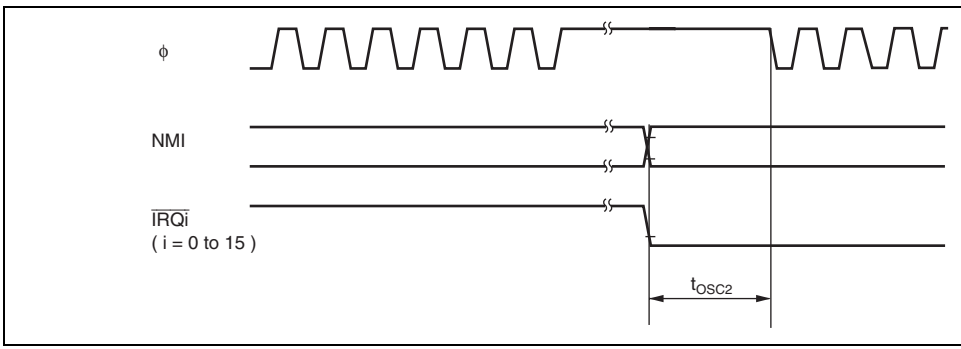
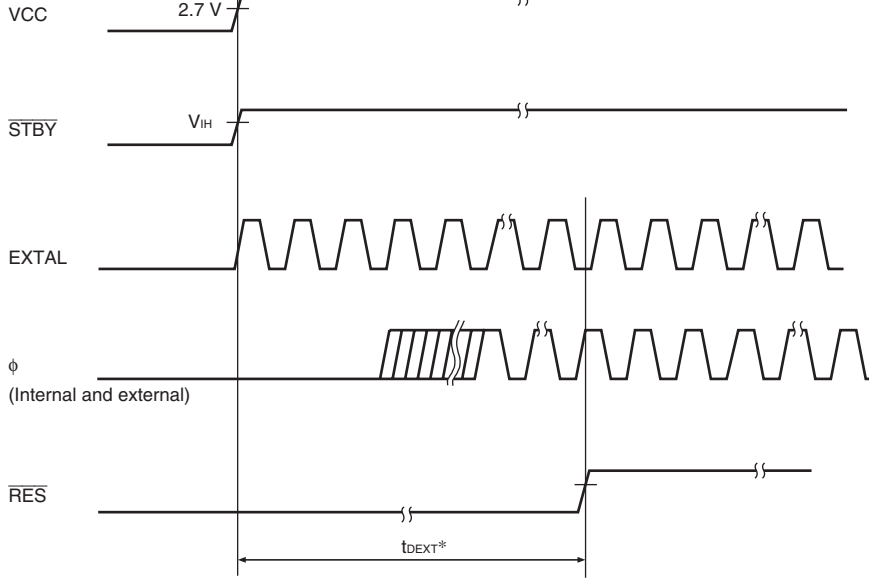


Figure 24.6 Oscillation Stabilization Timing (Exiting Software Standby Mo



Note: The external clock output stabilization delay time (t_{DEXT}) includes a $\overline{RES}$ pulse width (t_{RESW}).

Figure 24.8 Timing of External Clock Output Stabilization Delay Time

Table 24.7 shows the control signal timing. Only external interrupts NMI and IRQ0 to I
be operated based on the subclock (ϕSUB = 32.768 kHz).

Table 24.7 Control Signal Timing

Condition: VCC = 3.0 V to 3.6 V, VSS = 0 V, ϕ = 20 MHz to 25 MHz

Item	Symbol	Min.	Max.	Unit	Test Conditions
$\overline{\text{RES}}$ setup time	t_{RESS}	200	—	ns	Figure 24.10
$\overline{\text{RES}}$ pulse width	t_{RESW}	20	—	t_{cyc}	
NMI setup time	t_{NMIS}	150	—	ns	Figure 24.11
NMI hold time	t_{NMIH}	10	—		
NMI pulse width (exiting software standby mode)	t_{NMIW}	200	—		
IRQ setup time ($\overline{\text{IRQ15}}$ to $\overline{\text{IRQ0}}$)	t_{IRQS}	150	—		
IRQ hold time ($\overline{\text{IRQ15}}$ to $\overline{\text{IRQ0}}$)	t_{IRQH}	10	—		
IRQ pulse width ($\overline{\text{IRQ15}}$ to $\overline{\text{IRQ0}}$) (exiting software standby mode)	t_{IRQW}	200	—		

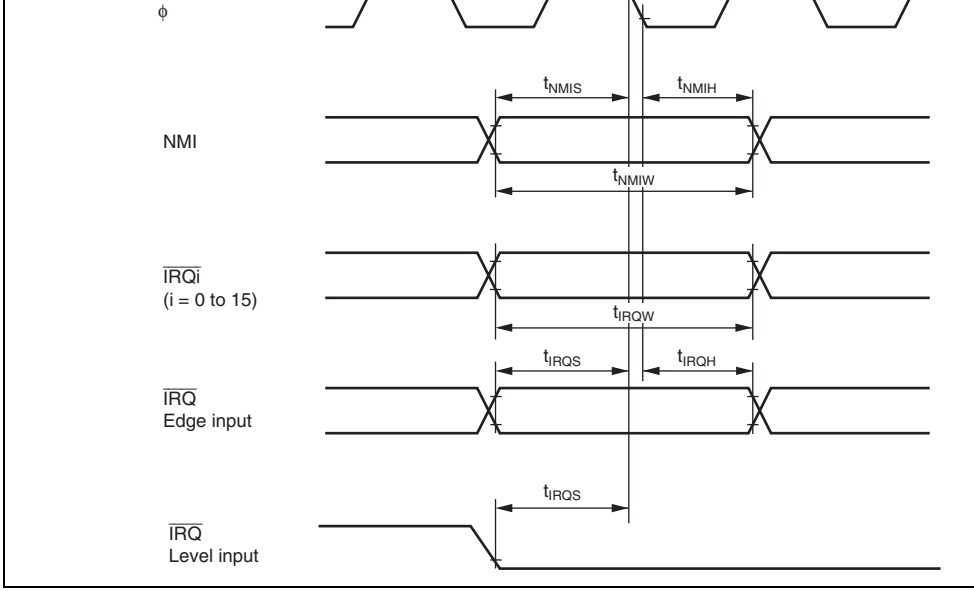


Figure 24.11 Interrupt Input Timing

I/O ports	Output data delay time		t_{PWD}	—	30	ns	Figure 24.1
	Input data setup time		t_{PRS}	20	—		
	Input data hold time		t_{PRH}	20	—		
PWMX	Timer output delay time		t_{PWOD}	—	30	ns	Figure 24.1
SCI	Input clock cycle	Asynchronous	t_{Scyc}	4	—	t_{cyc}	Figure 24.1
		Synchronous		6	—		
	Input clock pulse width		t_{SCKW}	0.4	0.6	t_{Scyc}	
	Input clock rise time		t_{SCKr}	—	1.5	t_{cyc}	
	Input clock fall time		t_{SCKf}	—	1.5		
	Transmit data delay time (synchronous)		t_{TXD}	—	30	ns	Figure 24.1
	Receive data setup time (synchronous)		t_{RXS}	20	—		
	Receive data hold time (synchronous)		t_{RXH}	20	—		
A/D converter	Trigger input setup time		t_{TRGS}	20	—	ns	Figure 24.1
WDT	$\overline{\text{RESO}}$ output delay time		t_{RESD}	—	50	ns	Figure 24.1
	$\overline{\text{RESO}}$ output pulse width		t_{RESOW}	132	—	t_{cyc}	

Figure 24.12 I/O Port Input/Output Timing

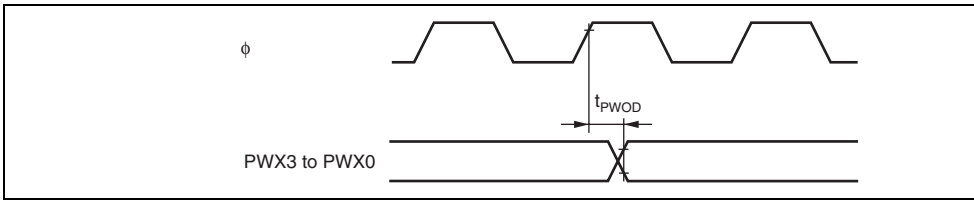


Figure 24.13 PWMX Output Timing

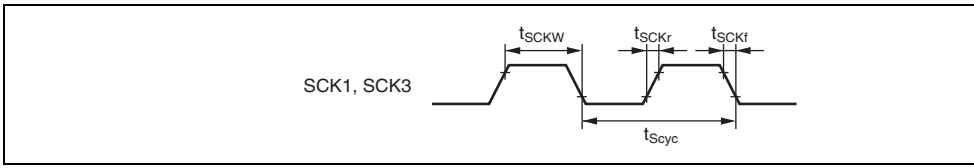


Figure 24.14 SCK Clock Input Timing

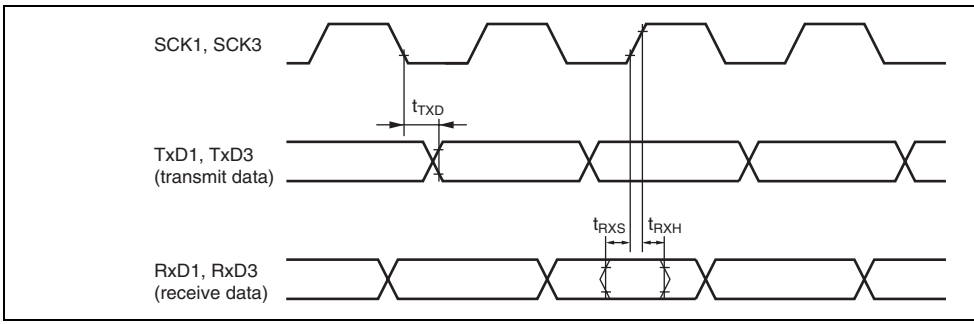


Figure 24.15 SCI Input/Output Timing (Clock Synchronous Mode)

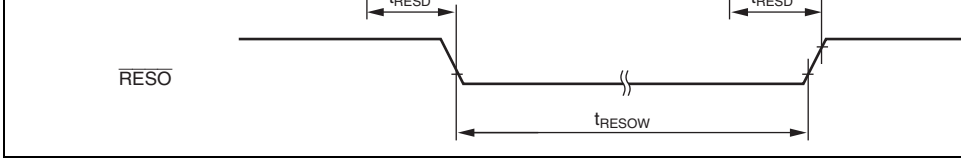
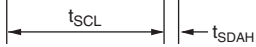


Figure 24.17 WDT Output Timing ($\overline{\text{RESO}}$)

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	—	—	
Retransmission start condition input setup time	t_{STAS}	3	—	—	
Stop condition input setup time	t_{STOS}	3	—	—	
Data input setup time	t_{SDAS}	0.5	—	—	
Data input hold time	t_{SDAH}	0	—	—	ns
SCL, SDA capacitive load	C_b	—	—	400	pF

Note: * 17.5 t_{cyc} or 37.5 t_{cyc} can be set according to the clock selected for use by the IIC



Note: * S, P, and Sr indicate the following conditions:
S: Start condition
P: Stop condition
Sr: Retransmission start condition

Figure 24.18 I²C Bus Interface Input/Output Timing

Table 24.10 LPC Module Timing

Conditions: VCC = 3.0 V to 3.6V, VSS = 0 V, ϕ = 20 MHz to 25 MHz

Item	Symbol	Min.	Typ.	Max.	Unit	Test Conditio
Input clock cycle	t_{Lcyc}	30	—	—	ns	Figure 24.19
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	—	—		

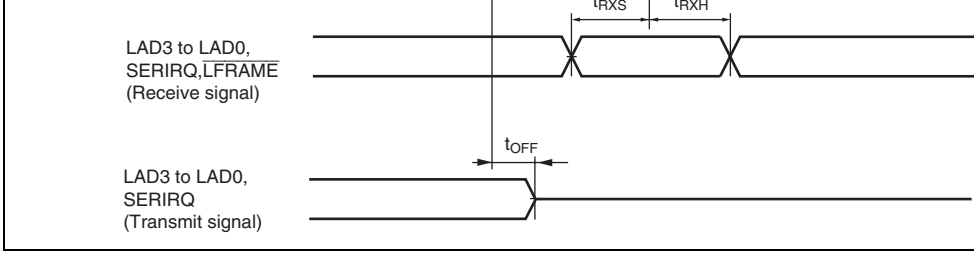


Figure 24.19 LPC Interface (LPC) Timing

ETRST pulse width	t_{TRSTW}	20	—	t_{cyc}	Figure 24.2
Reset hold transition pulse width	t_{RSTHW}	3	—		
ETMS setup time	t_{TMSS}	20	—	ns	Figure 24.2
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 $t_{cyc} \leq t_{TCKcyc}$

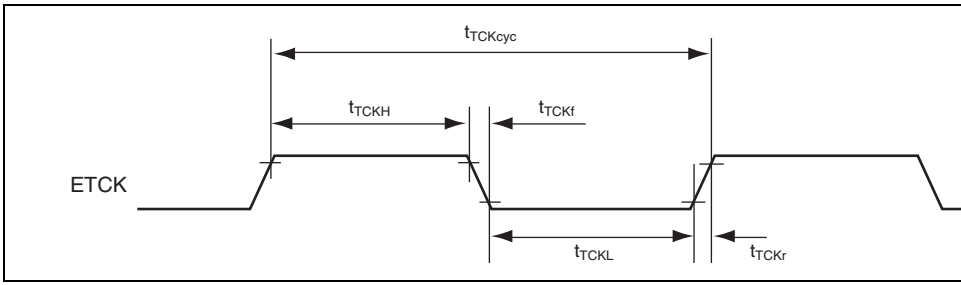


Figure 24.20 JTAG ETCK Timing

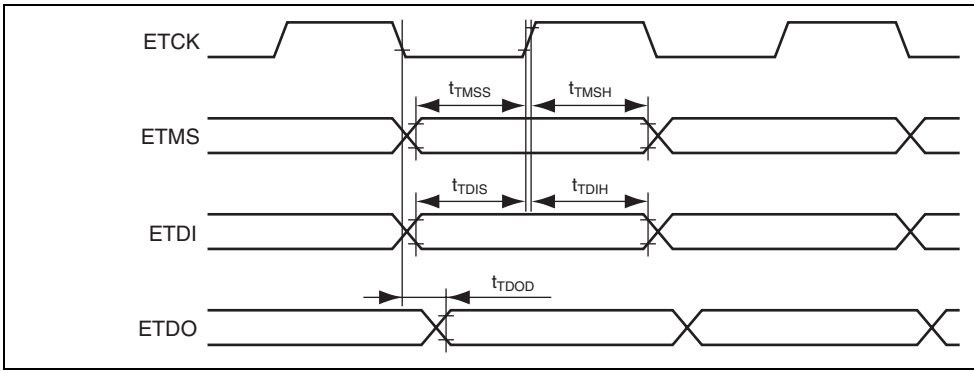


Figure 24.22 JTAG Input/Output Timing

VSS = AVSS = 0 V, φ = 20 MHz to 25 MHz

Item	Condition A Min.	Condition A Typ.	Condition A Max.	Condition B Min.	Condition B Typ.	Condition B 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	

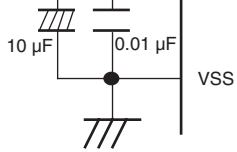
Notes: 1. Value when using the maximum operating frequency in single mode of 80 sta
2. Value when using the maximum operating frequency in single mode of 160 st

Item	Symbol	Min.	Typ.	Max.	Unit	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	—	4.5	12	s/512 Kbytes	Ta =
Erase time (total)*1*2*4	Σt_E	—	4.5	12	s/512 Kbytes	Ta =
Programming and Erase time (total)*1*2*4	Σt_{PE}	—	9.0	24	s/512 Kbytes	Ta =
Reprogramming count*5	N_{WEC}	100*3	1000	—	Times	
Data retention time*4	t_{DRP}	10	—	—	Years	

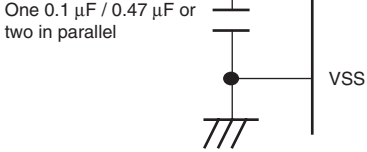
Notes: 1. Programming and erase time depends on the data.
2. Programming and erase time do not include data transfer time.
3. This value indicates the minimum number of which the flash memory are reprogrammed with all characteristics guaranteed. (The guaranteed value rang 1 to the minimum number.)
4. 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.

		—	300	1500	ms/32-Kbyte block	
		—	600	3000	ms/64-Kbyte block	
Programming time (total)*1*2*4	Σt_P	—	4.5	24	s/512 Kbytes	Ta =
Erase time (total)*1*2*4	Σt_E	—	4.5	24	s/512 Kbytes	Ta =
Programming and Erase time (total)*1*2*4	Σt_{PE}	—	9.0	48	s/512 Kbytes	Ta =
Reprogramming count*5	N_{WEC}	1000*3	—	—	Times	
Data retention time*4	t_{DRP}	10	—	—	Years	

Notes: 1. Programming and erase time depends on the data.
2. Programming and erase time do not include data transfer time.
3. This value indicates the minimum number of which the flash memory are reprogrammed with all characteristics guaranteed. (The guaranteed value ran 1 to the minimum number.)
4. This value indicates the characteristics while the flash memory is reprogramm the specified range (including the minimum number).
5. Reprogramming count in each erase block.



It is recommended that a bypass capacitor be connected to the VCC pin. (The values are reference values.)
When connecting, place a bypass capacitor near the pin.



Do not connect Vcc power supply to the 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 24.23 Connection of VCL Capacitor

Port 3	T	T	kept	kept	I/O port
Port 4	T	T	kept	kept	I/O port
Port 57	T	T	kept	kept	I/O port
Port 56 ϕ, EXCL	T	T	[DDR = 1] : H [DDR = 0] : T	[DDR = 1] : Clock output [DDR = 0] : T	Clock o EXCL in Input po
Port 53, 52	T	T	kept	kept	I/O port
Port 6	T	T	kept	kept	I/O port
Port 7	T	T	T	T	Input po
Port 8	T	T	kept	kept	I/O port
Port A	T	T	kept	kept	I/O port
Port C	T	T	kept	kept	I/O port
Port E	T	T	kept	kept	I/O port

[Legend]

H: High level

L: Low level

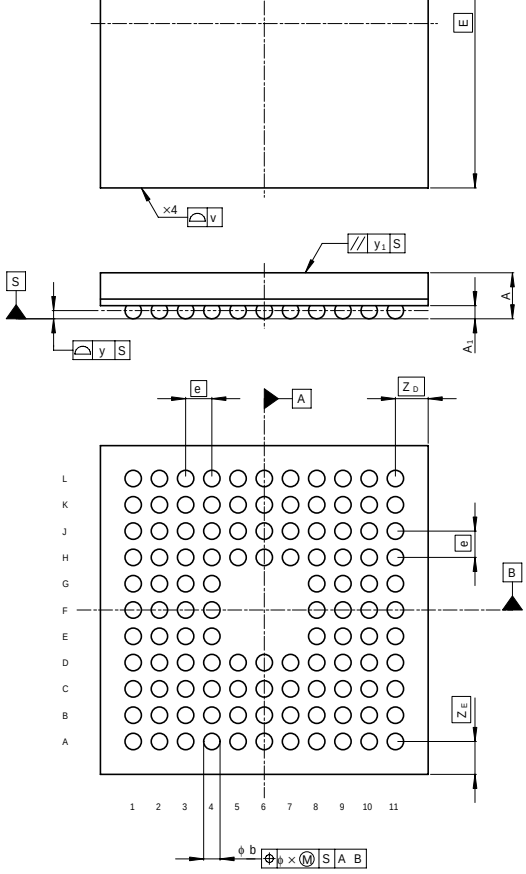
T: High impedance

kept: Input port pins are in the high-impedance state (when DDR = 0 and PCR = 1, the up MOS remains on).

Output port pins retain their states.

Functions of some pins will be changed to the I/O port function, which is determin DDR and DR, because the on-chip peripheral module associated with that pin fu initialized.

DDR: Data direction register



Reference Symbol	Dimension in Millim		
	Min	Nom	
D	—	10.00	
E	—	10.00	
v	—	—	
w	—	—	
A	—	—	
A_1	0.35	0.40	
e	—	0.80	
b	0.45	0.50	
x	—	—	
y	—	—	
y_1	—	—	
S_D	—	—	
S_E	—	—	
Z_D	—	1.00	
Z_E	—	1.00	

Figure C.1 Package Dimensions (PLBG0112GA-A)

4	0	0	—	Flash programming/
6	0	1	Emulation	On-chip emulation m

5.5 Interrupt Exception Handling Vector Table Table 5.3 Interrupt Sources, Vector Addresses, and Interrupt Priorities	77	Note added Note: Vector numbers not listed above are reserve system.
Section 7 Data Transfer Controller (DTC)	97 to 124	Description amended normal mode → normal transfer mode repeat mode → repeat transfer mode

		CRAL is decremented by 1 every time data is transfe the contents of CRAH are transferred when the coun reaches H'00. The number of times data is transferre when CRAH = CRAL = H'01, 255 when CRAH = CRA and 256 when CRAH = CRAL = H'00. In block transfer mode CRA is divided in two, with the eight bits designated as CRAH and the lowest eight b CRAL. CRAH holds the value for the block size, and functions as an 8-bit block size counter (1 to 256). CR decremented by 1 every time data is transferred, and contents of CRAH are transferred when the counter v reaches H'00. The block size is one byte (or one wor CRAH = CRAL = H'01, 255 bytes (or 255 words) whe = 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	111	Newly added
10.1 Features • Special functions provided by automatic addition function	193	Description deleted

(Data 2)

[6] ACKB set to 1

[11] BBS
SCP
(Stop co

16.3.2 Host Interface Control Registers 2 and 3 (HICR2 and HICR3) • HICR3	417	Table amended

Bit	Bit Name	Initial Value	R/W Slave	R/W Host	Description
7	LFRAME	Undefined	R	—	0: $\overline{\text{LFRAME}}$ Pin s 1: $\overline{\text{LFRAME}}$ Pin s
6	—	Undefined	R	—	Reserved
5	SERIRQ	Undefined	R	—	0: SERIRQ Pin s 1: SERIRQ Pin s
4	LRESET	Undefined	R	—	0: $\overline{\text{LRESET}}$ Pin s 1: $\overline{\text{LRESET}}$ Pin s
3	—	Undefined	R	—	Reserved
2	—	Undefined	R	—	Reserved
1	—	Undefined	R	—	Reserved
0	—	Undefined	R	—	Reserved

	MD1					
Input high voltage	EXTAL		VCC × 0.7	—	VCC	
	Port 7		2.2	—	AVC	
	SCL3 to SCL0, SDA3 to SDA0, Port 80 to 83, C0 to C3		—	—	5.5	
	SERIRQ, LAD3 to LAD0, LCLK, LRESET, LFRAME		VCC × 0.5	—	VCC	
			2.0	—	VCC	
	Input pins other than (1) and (2) above		2.2	—	VCC	

24.5 Flash Memory Characteristics	694	Title and Table amended

Table 24.13 Flash Memory Characteristics (100 Programming/ Erasing Cycles Specification)

Item	Symbol	Min.	Typ.	Max.	Unit
Programming time (total)*1*2*4	Σt_P	—	4.5	12	s/512 Kbytes
Erase time (total)*1*2*4	Σt_E	—	4.5	12	s/512 Kbytes
Programming and Erase time (total)*1*2*4	Σt_{PE}	—	9.0	24	s/512 Kbytes
Reprogramming count*5	N_{WEC}	100*3	1000	—	Times
Data retention time*4	T_{DRP}	10	—	—	Years

Table 24.14 Flash Memory Characteristics (1,000 Programming/ Erasing Cycles Specification)	695	Newly added
B. Product Lineup	698	Table amended

Product Type		Type Code	Mark Code	Pa
R4F2153	F-ZTAT version	R4F2153	F2153VBR25KDV	PL

B

D

E

G

H

I

L

M

N

P

R

S

T

U

V

Renesas 16-Bit Single-Chip Microcomputer
Hardware Manual
H8S/2153 Group

Publication Date: Rev.1.00, March 17, 2008
Rev.3.00, 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 Technology Corp. Sales Strategic Planning Div. Nippon Bldg., 2-6-2, Ohte-machi, Chiyoda-ku, Tokyo 100-0004, J

RENESAS

RENESAS SALES OFFICES

http://www.ren

Refer to "**http://www.renesas.com/en/network**" for the latest and detailed information.

Renesas Technology America, Inc.
450 Holger Way, San Jose, CA 95134-1368, U.S.A
Tel: <1> (408) 382-7500, Fax: <1> (408) 382-7501

Renesas Technology Europe Limited
Dukes Meadow, Millboard Road, Bourne End, Buckinghamshire, SL8 5FH, U.K.
Tel: <44> (1628) 585-100, Fax: <44> (1628) 585-900

Renesas Technology (Shanghai) Co., Ltd.
Unit 204, 205, AZIACenter, No.1233 Lujiazui Ring Rd, Pudong District, Shanghai, China 200120
Tel: <86> (21) 5877-1818, Fax: <86> (21) 6887-7858/7898

Renesas Technology Hong Kong Ltd.
7th Floor, North Tower, World Finance Centre, Harbour City, Canton Road, Tsimshatsui, Kowloon, Hong Kong
Tel: <852> 2265-6688, Fax: <852> 2377-3473

Renesas Technology Taiwan Co., Ltd.
10th Floor, No.99, Fushing North Road, Taipei, Taiwan
Tel: <886> (2) 2715-2888, Fax: <886> (2) 3518-3399

Renesas Technology Singapore Pte. Ltd.
1 Harbour Front Avenue, #06-10, Keppel Bay Tower, Singapore 098632
Tel: <65> 6213-0200, Fax: <65> 6278-8001

Renesas Technology Korea Co., Ltd.
Kukje Center Bldg. 18th Fl., 191, 2-ka, Hangang-ro, Yongsan-ku, Seoul 140-702, Korea
Tel: <82> (2) 796-3115, Fax: <82> (2) 796-2145

Renesas Technology Malaysia Sdn. Bhd
Unit 906, Block B, Menara Amcorp, Amcorp Trade Centre, No.18, Jln Persiaran Barat, 46050 Petaling Jaya, Selangor Darul Ehsan, Ma
Tel: <603> 7955-9390, Fax: <603> 7955-9510

H8S/2153 Group
Hardware Manual

Renesas Electronics Corporation
1753, Shimonumabe, Nakahara-ku, Kawasaki-shi, Kanagawa 211-8668 Japan

REJ09B0384-0300